

Transport Area Working Group  
Internet-Draft  
Updates: 3819 (if approved)  
Intended status: Best Current Practice  
Expires: September 4, 2014

B. Briscoe  
BT  
J. Kaippallimalil  
Huawei  
P. Thaler  
Broadcom Corporation  
March 03, 2014

Guidelines for Adding Congestion Notification to Protocols that  
Encapsulate IP  
draft-briscoe-tsvwg-ecn-encap-guidelines-04

Abstract

The purpose of this document is to guide the design of congestion notification in any lower layer or tunnelling protocol that encapsulates IP. The aim is for explicit congestion signals to propagate consistently from lower layer protocols into IP. Then the IP internetwork layer can act as a portability layer to carry congestion notification from non-IP-aware congested nodes up to the transport layer (L4). Following these guidelines should assure interworking between new lower layer congestion notification mechanisms, whether specified by the IETF or other standards bodies.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 4, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Scope . . . . .	5
2. Terminology . . . . .	6
3. Modes of Operation . . . . .	7
3.1. Feed-Forward-and-Up Mode . . . . .	8
3.2. Feed-Up-and-Forward Mode . . . . .	9
3.3. Feed-Backward Mode . . . . .	10
3.4. Null Mode . . . . .	12
4. Feed-Forward-and-Up Mode: Guidelines for Adding Congestion Notification . . . . .	12
4.1. IP-in-IP Tunnels with Tightly Coupled Shim Headers . . . . .	13
4.2. Wire Protocol Design: Indication of ECN Support . . . . .	13
4.3. Encapsulation Guidelines . . . . .	15
4.4. Decapsulation Guidelines . . . . .	17
4.5. Sequences of Similar Tunnels or Subnets . . . . .	18
4.6. Reframing and Congestion Markings . . . . .	19
5. Feed-Up-and-Forward Mode: Guidelines for Adding Congestion Notification . . . . .	19
6. Feed-Backward Mode: Guidelines for Adding Congestion Notification . . . . .	21
7. IANA Considerations (to be removed by RFC Editor) . . . . .	22
8. Security Considerations . . . . .	22
9. Conclusions . . . . .	22
10. Acknowledgements . . . . .	23
11. Comments Solicited . . . . .	23
12. References . . . . .	23
12.1. Normative References . . . . .	23
12.2. Informative References . . . . .	24
Appendix A. Outstanding Document Issues . . . . .	27
Appendix B. Changes in This Version (to be removed by RFC Editor) . . . . .	27

## 1. Introduction

The benefits of Explicit Congestion Notification (ECN) described below can only be fully realised if support for ECN is added to the relevant subnetwork technology, as well as to IP. When a lower layer buffer drops a packet obviously it does not just drop at that layer; the packet disappears from all layers. In contrast, when a lower layer marks a packet with ECN, the marking needs to be explicitly propagated up the layers. The same is true if a buffer marks the outer header of a packet that encapsulates inner tunnelled headers. Forwarding ECN is not as straightforward as other headers because it has to be assumed ECN may be only partially deployed. If an egress at any layer is not ECN-aware, or if the ultimate receiver or sender is not ECN-aware, congestion needs to be indicated by dropping a packet, not marking it.

The purpose of this document is to guide the addition of congestion notification to any subnet technology or tunnelling protocol, so that lower layer equipment can signal congestion explicitly and it will propagate consistently into encapsulated (higher layer) headers, otherwise the signals will not reach their ultimate destination.

ECN is defined in the IP header (v4 & v6) [RFC3168] to allow a resource to notify the onset of queue build-up without having to drop packets, by explicitly marking a proportion of packets with the congestion experienced (CE) codepoint.

Given a suitable marking scheme, ECN removes nearly all congestion loss and it cuts delays for two main reasons:

- o It avoids the delay when recovering from congestion losses, which particularly benefits small flows or real-time flows, making their delivery time predictably short [RFC2884];
- o As ECN is used more widely by end-systems, it will gradually remove the need to configure a degree of delay into buffers before they start to notify congestion (the cause of bufferbloat). This is because drop involves a trade-off between sending a timely signal and trying to avoid impairment, whereas ECN is solely a signal not an impairment, so there is no harm triggering it earlier.

Some lower layer technologies (e.g. MPLS, Ethernet) are used to form subnetworks with IP-aware nodes only at the edges. These networks are often sized so that it is rare for interior queues to overflow. However, this has often been more due to the inability of the original TCP protocol to saturate the links. For many years, fixes such as window scaling [RFC1323] proved hard to deploy. But now that modern

operating systems are finally capable of saturating interior links, even the buffers of well-provisioned interior switches will need to signal episodes of queuing.

Propagation of ECN is defined for MPLS [RFC5129], and is being defined for TRILL [trill-rbridge-options], but it remains to be defined for a number of other subnetwork technologies.

Similarly, ECN propagation is yet to be defined for many tunnelling protocols. [RFC6040] defines how ECN should be propagated for IP-in-IP [RFC2003] and IPsec [RFC4301] tunnels. However, as Section 9.3 of RFC3168 pointed out, ECN support will need to be defined for other tunnelling protocols, e.g. L2TP [RFC2661], GRE [RFC1701], [RFC2784], PPTP [RFC2637] and GTP [GTPv1], [GTPv1-U], [GTPv2-C].

Incremental deployment is the most tricky aspect when adding support for ECN. The original ECN protocol in IP [RFC3168] was carefully designed so that a congested buffer would not mark a packet (rather than drop it) unless both source and destination hosts were ECN-capable. Otherwise its congestion markings would never be detected and congestion would just deteriorate further. However, to support congestion marking below the IP layer, it is not sufficient to only check that the two end-points support ECN; correct operation also depends on the decapsulator at each subnet egress faithfully propagating congestion notifications to the higher layer. Otherwise, a legacy decapsulator might silently fail to propagate any ECN signals from the outer to the forwarded header. Then the lost signals would never be detected and again congestion would deteriorate further. The guidelines given later require protocol designers to carefully consider incremental deployment, and suggest various safe approaches for different circumstances.

Of course, the IETF does not have standards authority over every link layer protocol. So this document gives guidelines for designing propagation of congestion notification across the interface between IP and protocols that may encapsulate IP (i.e. that can be layered beneath IP). Each lower layer technology will exhibit different issues and compromises, so the IETF or the relevant standards body must be free to define the specifics of each lower layer congestion notification scheme. Nonetheless, if the guidelines are followed, congestion notification should interwork between different technologies, using IP in its role as a 'portability layer'.

Therefore, the capitalised term 'SHOULD' or 'SHOULD NOT' are often used in preference to 'MUST' or 'MUST NOT', because it is difficult to know the compromises that will be necessary in each protocol design. If a particular protocol design chooses to contradict a

'SHOULD (NOT)' given in the advice below, it MUST include a sound justification.

It has not been possible to give common guidelines for all lower layer technologies, because they do not all fit a common pattern. Instead they have been divided into a few distinct modes of operation: feed-forward-and-upward; feed-upward-and-forward; feed-backward; and null mode. These modes are described in Section 3, then in the following sections separate guidelines are given for each mode.

This document updates the advice to subnetwork designers about ECN in Section 13 of [RFC3819].

### 1.1. Scope

This document only concerns wire protocol processing of explicit notification of congestion and makes no changes or recommendations concerning algorithms for congestion marking or for congestion response (algorithm issues should be independent of the layer the algorithm operates in).

The question of congestion notification signals with different semantics to those of ECN in IP is touched on in a couple of specific cases (e.g. QCN [IEEE802.1Qau]) and with schemes with multiple severity levels such as PCN [RFC6660]). However, no attempt is made to give guidelines about schemes with different semantics that are yet to be invented.

The semantics of congestion signals can be relative to the traffic class. Therefore correct propagation of congestion signals could depend on correct propagation of any traffic class field between the layers. In this document, correct propagation of traffic class information is assumed, while what 'correct' means and how it is achieved is covered elsewhere (e.g. [RFC2983]) and is outside the scope of the present document.

Note that these guidelines do not require the subnet wire protocol to be changed to accommodate congestion notification. Another way to add congestion notification without consuming header space in the subnet protocol might be to use a parallel control plane protocol.

This document focuses on the congestion notification interface between IP and lower layer protocols that can encapsulate IP, where the term 'IP' includes v4 or v6, unicast, multicast or anycast. However, it is likely that the guidelines will also be useful when a lower layer protocol or tunnel encapsulates itself (e.g. Ethernet MAC in MAC [IEEE802.1Qah]) or when it encapsulates other protocols. In

the feed-backward mode, propagation of congestion signals for multicast and anycast packets is out-of-scope (because it would be so complicated that it is hoped no-one would attempt such an abomination).

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Further terminology used within this document:

**Protocol data unit (PDU):** Information that is delivered as a unit among peer entities of a layered network consisting of protocol control information (typically a header) and possibly user data (payload) of that layer. The scope of this document includes layer 2 and layer 3 networks, where the PDU is respectively termed a frame or a packet (or a cell in ATM). PDU is a general term for any of these. This definition also includes a payload with a shim header lying somewhere between layer 2 & 3.

**Transport:** The end-to-end transmission control function, conventionally considered at layer-4 in the OSI reference model. Given the audience for this document will often use the word transport to mean low level bit carriage, whenever the term is used it will be qualified, e.g. 'L4 transport'.

**Encapsulator:** The link or tunnel endpoint function that adds an outer header to a PDU (also termed the 'link ingress', the 'subnet ingress', the 'ingress tunnel endpoint' or just the 'ingress' where the context is clear).

**Decapsulator:** The link or tunnel endpoint function that removes an outer header from a PDU (also termed the 'link egress', the 'subnet egress', the 'egress tunnel endpoint' or just the 'egress' where the context is clear).

**Incoming header:** The header of an arriving PDU before encapsulation.

**Outer header:** The header added to encapsulate a PDU.

**Inner header:** The header encapsulated by the outer header.

**Outgoing header:** The header forwarded by the decapsulator.

**CE:** Congestion Experienced [RFC3168]

ECT: ECN-Capable Transport [RFC3168]

Not-ECT: Not ECN-Capable Transport [RFC3168]

ECN-PDU: A PDU that is part of a feedback loop within which all the nodes that need to propagate explicit congestion notifications back to the Load Regulator are ECN-capable. An IP packet with a non-zero ECN field implies that the endpoints are ECN-capable, so this would be an ECN-PDU. However, ECN-PDU is intended to be a general term for a PDU at any layer, not just IP.

Not-ECN-PDU: A PDU that is part of a feedback-loop within which some nodes necessary to propagate explicit congestion notifications back to the load regulator are not ECN-capable.

Load Regulator: For each flow of PDUs, the transport function that is capable of controlling the data rate. Typically located at the data source, but in-path nodes can regulate load in some congestion control arrangements (e.g. admission control or policing nodes). Note the term "a function capable of controlling the load" deliberately includes a transport that doesn't actually control the load but ideally it ought to (e.g. a sending application without congestion control that uses UDP).

Congestion Baseline: The location of the function on the path that initialised the values of all congestion notification fields in a sequence of packets, before any are set to the congestion experienced (CE) codepoint if they experience congestion further downstream. Typically the original data source at layer-4.

### 3. Modes of Operation

This section sets down the different modes by which congestion information is passed between the lower layer and the higher one. It acts as a reference framework for the following sections, which give normative guidelines for designers of explicit congestion notification protocols, taking each mode in turn:

Feed-Forward-and-Up: Nodes feed forward congestion notification towards the egress within the lower layer then up and along the layers towards the end-to-end destination at the transport layer. The following local optimisation is possible:

Feed-Up-and-Forward: A lower layer switch feeds-up congestion notification directly into the ECN field in the higher layer (e.g. IP) header, irrespective of whether the node is at the egress of a subnet.

Feed-Backward: Nodes feed back congestion signals towards the ingress of the lower layer and (optionally) attempt to control congestion within their own layer.

Null: Nodes cannot experience congestion at the lower layer except at ingress nodes (which are IP-aware or equivalently higher-layer-aware).

### 3.1. Feed-Forward-and-Up Mode

Like IP and MPLS, many subnet technologies are based on self-contained protocol data units (PDUs) or frames sent unreliably. They provide no feedback channel at the subnetwork layer, instead relying on higher layers (e.g. TCP) to feed back loss signals.

In these cases, ECN may best be supported by standardising explicit notification of congestion into the lower layer protocol that carries the data forwards. It will then also be necessary to define how the egress of the lower layer subnet propagates this explicit signal into the forwarded upper layer (IP) header. It can then continue forwards until it finally reaches the destination transport (at L4). Then typically the destination will feed this congestion notification back to the source transport using an end-to-end protocol (e.g. TCP). This is the arrangement that has already been used to add ECN to IP-in-IP tunnels [RFC6040], IP-in-MPLS and MPLS-in-MPLS [RFC5129].

This mode is illustrated in Figure 1. Along the middle of the figure, layers 2, 3 & 4 of the protocol stack are shown, and one packet is shown along the bottom as it progresses across the network from source to destination, crossing two subnets connected by a router, and crossing two switches on the path across each subnet. Congestion at the output of the first switch (shown as \*) leads to a congestion marking in the L2 header (shown as C in the illustration of the packet). The chevrons show the progress of the resulting congestion indication. It is propagated from link to link across the subnet in the L2 header, then when the router removes the marked L2 header, it propagates the marking up into the L3 (IP) header. The router forwards the marked L3 header into subnet 2, and when it adds a new L2 header it copies the L3 marking into the L2 header as well, as shown by the 'C's in both layers (assuming the technology of subnet 2 also supports explicit congestion marking).

Note that there is no implication that each 'C' marking is encoded the same; a different encoding might be used for the 'C' marking in each protocol.

Finally, for completeness, we show the L3 marking arriving at the destination, where the host transport protocol (e.g. TCP) feeds it



back to the source in the L4 acknowledgement (the 'C' at L4 in the packet at the top of the diagram).

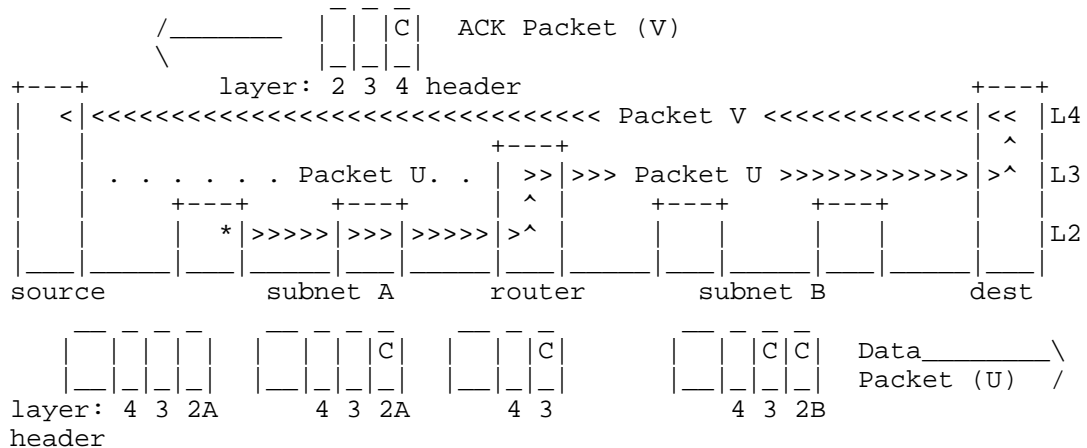


Figure 1: Feed-Forward-and-Up Mode

Of course, modern networks are rarely as simple as this text-book example, often involving multiple nested layers. For example, a 3GPP mobile network may have two IP-in-IP (GTP) tunnels in series and an MPLS backhaul between the base station and the first router. Nonetheless, the example illustrates the general idea of feeding congestion notification forward then upward whenever a header is removed at the egress of a subnet.

Note that the FECN (forward ECN) bit in Frame Relay and the explicit forward congestion indication (EFCI [ITU-T.I.371]) bit in ATM user data cells follow a feed-forward pattern. However, in ATM, this is only as part of a feed-forward-and-backward pattern at the lower layer, not feed-forward-and-up out of the lower layer--the intention was never to interface to IP ECN at the subnet egress. To our knowledge, Frame Relay FECN is solely used to detect where more capacity should be provisioned [Buck00].

### 3.2. Feed-Up-and-Forward Mode

Ethernet is particularly difficult to extend incrementally to support explicit congestion notification. One way to support ECN in such cases has been to use so called 'layer-3 switches'. These are Ethernet switches that bury into the Ethernet payload to find an IP header and manipulate or act on certain IP fields (specifically Diffserv & ECN). For instance, in Data Center TCP [DCTCP], layer-3 switches are configured to mark the ECN field of the IP header within

the Ethernet payload when their output buffer becomes congested. With respect to switching, a layer-3 switch acts solely on the addresses in the Ethernet header; it doesn't use IP addresses, and it doesn't decrement the TTL field in the IP header.

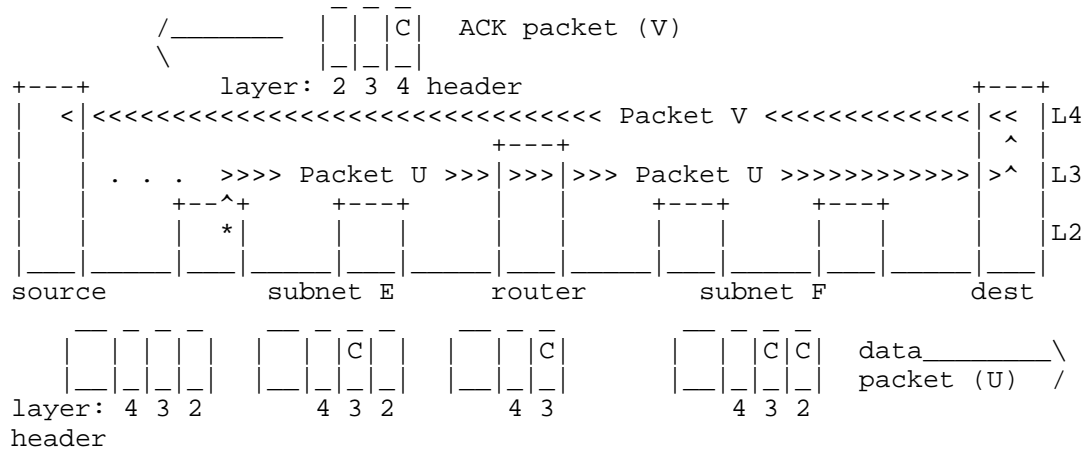


Figure 2: Feed-Up-and-Forward Mode

By comparing Figure 2 with Figure 1, it can be seen that subnet E (perhaps a subnet of layer-3 Ethernet switches) works in feed-up-and-forward mode by notifying congestion directly into L3 at the point of congestion, even though the congested switch does not otherwise act at L3. In this example, the technology in subnet F (e.g. MPLS) does support ECN natively, so when the router adds the layer-2 header it copies the ECN marking from L3 to L2 as well.

### 3.3. Feed-Backward Mode

In some layer 2 technologies, explicit congestion notification has been defined for use internally within the subnet with its own feedback and load regulation, but typically the interface with IP for ECN has not been defined.

For instance, for the available bit-rate (ABR) service in ATM, the relative rate mechanism was one of the more popular mechanisms for managing traffic, tending to supersede earlier designs. In this approach ATM switches send special resource management (RM) cells in both the forward and backward directions to control the ingress rate of user data into a virtual circuit. If a switch buffer is approaching congestion or congested it sends an RM cell back towards the ingress with respectively the No Increase (NI) or Congestion

Indication (CI) bit set in its message type field [ATM-TM-ABR]. The ingress then holds or decreases its sending bit-rate accordingly.

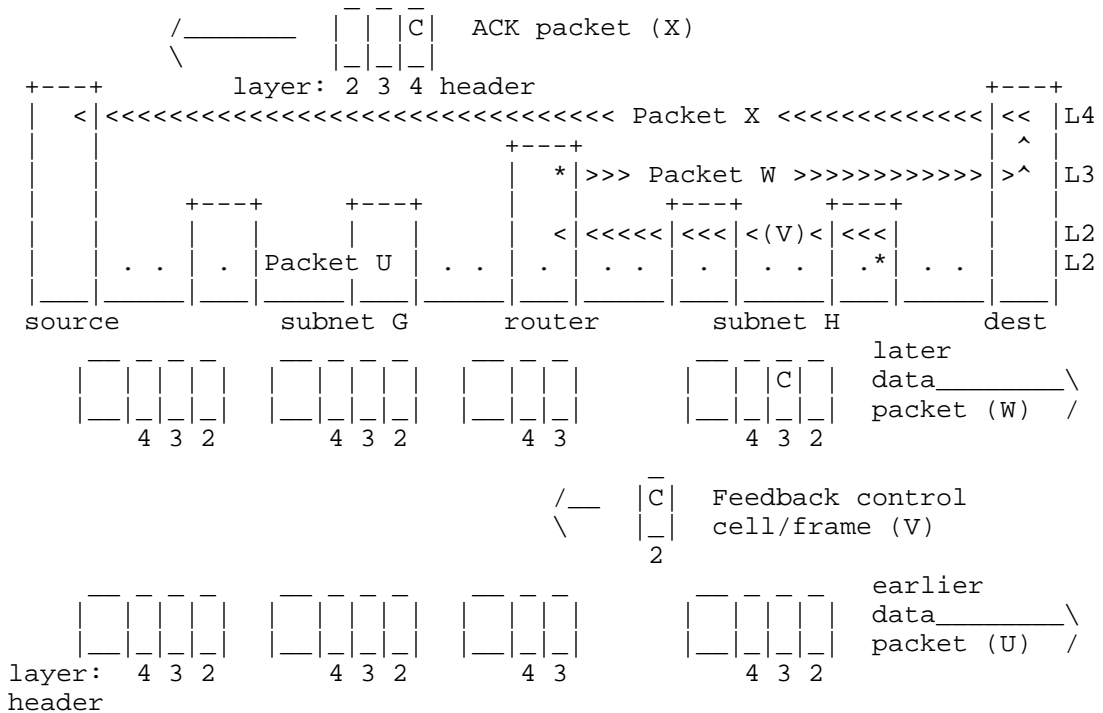


Figure 3: Feed-Backward Mode

ATM's feed-backward approach doesn't fit well when layered beneath IP's feed-forward approach--unless the initial data source is the same node as the ATM ingress. Figure 3 shows the feed-backward approach being used in subnet H. If the final switch on the path is congested (\*), it doesn't feed-forward any congestion indications on packet (U). Instead it sends a control cell (V) back to the router at the ATM ingress.

However, the backward feedback doesn't reach the original data source directly because IP doesn't support backward feedback (and subnet G is independent of subnet H). Instead, the router in the middle throttles down its sending rate but the original data sources don't reduce their rates. The resulting rate mismatch causes the middle router's buffer at layer 3 to back up until it becomes congested, which it signals forwards on later data packets at layer 3 (e.g. packet W). Note that the forward signal from the middle router is not triggered directly by the backward signal. Rather, it is

triggered by congestion resulting from the middle router's mismatched rate response to the backward signal.

In response to this later forward signalling, end-to-end feedback at layer-4 finally completes the tortuous path of congestion indications back to the origin data source, as before.

### 3.4. Null Mode

Often link and physical layer resources are 'non-blocking' by design. In these cases congestion notification may be implemented but it does not need to be deployed at the lower layer; ECN in IP would be sufficient.

A degenerate example is a point-to-point Ethernet link. Excess loading of the link merely causes the queue from the higher layer to back up, while the lower layer remains immune to congestion. Even a whole meshed subnetwork can be made immune to interior congestion by limiting ingress capacity and careful sizing of links, particularly if multi-path routing is used to ensure even worst-case patterns of load cannot congest any link.

## 4. Feed-Forward-and-Up Mode: Guidelines for Adding Congestion Notification

Feed-forward-and-up is the mode already used for signalling ECN up the layers through MPLS into IP [RFC5129] and through IP-in-IP tunnels [RFC6040]. These RFCs take a consistent approach and the following guidelines are designed to ensure this consistency continues as ECN support is added to other protocols that encapsulate IP. The guidelines are also designed to ensure compliance with the more general best current practice for the design of alternate ECN schemes given in [RFC4774].

The rest of this section is structured as follows:

- o Section 4.1 addresses the most straightforward cases, where [RFC6040] can be applied directly to add ECN to tunnels that are effectively the same as IP-in-IP tunnels.
- o The subsequent sections give guidelines for adding ECN to a subnet technology that uses feed-forward-and-up mode like IP, but it is not so similar to IP that [RFC6040] rules can be applied directly. Specifically:
  - \* Sections 4.2, 4.3 and 4.4 respectively address how to add ECN support to the wire protocol and to the encapsulators and decapsulators at the ingress and egress of the subnet.

- \* Section 4.5 deals with the special, but common, case of sequences of tunnels or subnets that all use the same technology
- \* Section 4.6 deals with the question of reframing when IP packets do not map 1:1 into lower layer frames.

#### 4.1. IP-in-IP Tunnels with Tightly Coupled Shim Headers

A common pattern for many tunnelling protocols is to encapsulate an inner IP header with shim header(s) then an outer IP header. In many cases the shim header(s) always have to be tightly coupled to the outer IP header because they are not sufficient as outer headers in their own right. In such cases the shim header(s) and the outer IP header are always added (or removed) in the same operation. Therefore, in all such tightly coupled IP-in-IP tunnelling protocols, the rules in [RFC6040] for propagating the ECN field between the two IP headers SHOULD be applied directly.

Examples of tightly coupled IP-in-IP tunnelling protocols where [RFC6040] can be applied directly are:

- o L2TP [RFC2661]
- o GRE [RFC1701], [RFC2784]
- o PPTP [RFC2637]
- o GTP [GTPv1], [GTPv1-U], [GTPv2-C]
- o VXLAN [vxlan].

#### 4.2. Wire Protocol Design: Indication of ECN Support

This section is intended to guide the redesign of any lower layer protocol that encapsulate IP to add native ECN support at the lower layer. It reflects the approaches used in [RFC6040] and in [RFC5129]. Therefore IP-in-IP tunnels or IP-in-MPLS or MPLS-in-MPLS encapsulations that already comply with [RFC6040] or [RFC5129] will already satisfy this guidance.

A lower layer (or subnet) congestion notification system:

1. SHOULD NOT apply explicit congestion notifications to PDUs that are destined for legacy layer-4 transport implementations that will not understand ECN, and

2. SHOULD NOT apply explicit congestion notifications to PDUs if the egress of the subnet might not propagate congestion notifications onward into the higher layer.

We use the term ECN-PDUs for a PDU on a feedback loop that will propagate congestion notification properly because it meets both the above criteria. And a Not-ECN-PDU is a PDU on a feedback loop that does not meet both criteria, and will therefore not propagate congestion notification properly. A corollary of the above is that a lower layer congestion notification protocol:

3. SHOULD be able to distinguish ECN-PDUs from Not-ECN-PDUs.

Note that there is no need for all interior nodes within a subnet to be able to mark congestion explicitly. A mix of ECN and drop signals from different nodes is fine. However, if any interior nodes might generate ECN markings, guideline 2 above says that all relevant egress node(s) SHOULD be able to propagate those markings up to the higher layer.

In IP, if the ECN field in each PDU is cleared to the Not-ECT (not ECN-capable transport) codepoint, it indicates that the L4 transport will not understand congestion markings. A congested buffer must not mark these Not-ECT PDUs, and therefore drops them instead.

The mechanism a lower layer uses to distinguish the ECN-capability of PDUs need not mimic that of IP. All the above guidelines say is that the lower layer system, as a whole, should achieve the same outcome. For instance, ECN-capable feedback loops might use PDUs that are identified by a particular set of labels or tags. Alternatively, logical link protocols that use flow state might determine whether a PDU can be congestion marked by checking for ECN-support in the flow state. Other protocols might depend on out-of-band control signals.

The per-domain checking of ECN support in MPLS [RFC5129] is a good example of a way to avoid sending congestion markings to transports that will not understand them, without using any header space in the subnet protocol.

In MPLS, header space is extremely limited, therefore RFC5129 does not provide a field in the MPLS header to indicate whether the PDU is an ECN-PDU or a Not-ECN-PDU. Instead, interior nodes in a domain are allowed to set explicit congestion indications without checking whether the PDU is destined for a transport that will understand them. Nonetheless, this is made safe by requiring that the network operator upgrades all decapsulating edges of a whole domain at once, as soon as even one switch within the domain is configured to mark rather than drop during congestion. Therefore, any edge node that

might decapsulate a packet will be capable of checking whether the higher layer transport is ECN-capable. When decapsulating a CE-marked packet, if the decapsulator discovers that the higher layer (inner header) indicates the transport is not ECN-capable, it drops the packet--effectively on behalf of the earlier congested node (see Decapsulation Guideline 1 in Section 4.4).

It was only appropriate to define such an incremental deployment strategy because MPLS is targeted solely at professional operators, who can be expected to ensure that a whole subnetwork is consistently configured. This strategy might not be appropriate for other link technologies targeted at zero-configuration deployment or deployment by the general public (e.g. Ethernet). For such 'plug-and-play' environments it will be necessary to invent a failsafe approach that ensures congestion markings will never fall into black holes, no matter how inconsistently a system is put together. Alternatively, congestion notification relying on correct system configuration could be confined to flavours of Ethernet intended only for professional network operators, such as IEEE 802.1ah Provider Backbone Bridges (PBB).

QCN [IEEE802.1Qau] provides another example of how to indicate to lower layer devices that the end-points will not understand ECN. An operator can define certain 802.1p classes of service to indicate non-QCN frames and an ingress bridge is required to map arriving not-QCN-capable IP packets to one of these non-QCN 802.1p classes.

#### 4.3. Encapsulation Guidelines

This section is intended to guide the redesign of any node that encapsulates IP with a lower layer header when adding native ECN support to the lower layer protocol. It reflects the approaches used in [RFC6040] and in [RFC5129]. Therefore IP-in-IP tunnels or IP-in-MPLS or MPLS-in-MPLS encapsulations that already comply with [RFC6040] or [RFC5129] will already satisfy this guidance.

1. Egress Capability Check: A subnet ingress needs to be sure that the corresponding egress of a subnet will propagate any congestion notification added to the outer header across the subnet. This is necessary in addition to checking that an incoming PDU indicates an ECN-capable (L4) transport. Examples of how this guarantee might be provided include:
  - \* by configuration (e.g. if any label switches in a domain support ECN marking, [RFC5129] requires all egress nodes to have been configured to propagate ECN)

- \* by the ingress explicitly checking that the egress propagates ECN (e.g. TRILL uses IS-IS to check path capabilities before using critical options [trill-rbridge-options])
  - \* by inherent design of the protocol (e.g. by encoding ECN marking on the outer header in such a way that a legacy egress that does not understand ECN will consider the PDU corrupt and discard it, thus at least propagating a form of congestion signal).
2. Egress Fails Capability Check: If the ingress cannot guarantee that the egress will propagate congestion notification, the ingress SHOULD disable ECN when it forwards the PDU at the lower layer. An example of how the ingress might disable ECN at the lower layer would be by setting the outer header of the PDU to identify it as a Not-ECN-PDU, assuming the subnet technology supports such a concept.
  3. Standard Congestion Monitoring Baseline: Once the ingress to a subnet has established that the egress will correctly propagate ECN, on encapsulation it SHOULD encode the same level of congestion in outer headers as is arriving in incoming headers. For example it might copy any incoming congestion notification into the outer header of the lower layer protocol.

This ensures that all outer headers reflect congestion accumulated along the whole upstream path since the Load Regulator, not just since the ingress of the subnet. A node that is not the Load Regulator SHOULD NOT re-initialise the level of CE markings in the outer to zero.

This guideline is intended to ensure that any bulk congestion monitoring of outer headers (e.g. by a network management node monitoring ECN in passing frames) is most meaningful. For instance, if an operator measures CE in 0.4% of passing outer headers, this information is only useful if the operator knows where the proportion of CE markings was last initialised to 0% (the Congestion Baseline). Such monitoring information will not be useful if some subnet ingress nodes reset all outer CE markings while others copy incoming CE markings into the outer.

Most information can be extracted if the Congestion Baseline is standardised at the node that is regulating the load (the Load Regulator--typically the data source). Then the operator can measure both congestion since the Load Regulator, and congestion since the subnet ingress. The latter might be measurable by subtracting the level of CE markings on inner headers from that on outer headers (see Appendix C of [RFC6040]).



#### 4.4. Decapsulation Guidelines

This section is intended to guide the redesign of any node that decapsulates IP from within a lower layer header when adding native ECN support to the lower layer protocol. It reflects the approaches used in [RFC6040] and in [RFC5129]. Therefore IP-in-IP tunnels or IP-in-MPLS or MPLS-in-MPLS encapsulations that already comply with [RFC6040] or [RFC5129] will already satisfy this guidance.

A subnet egress SHOULD NOT simply copy congestion notification from outer headers to the forwarded header. It SHOULD calculate the outgoing congestion notification field from the inner and outer headers using the following guidelines. If there is any conflict, rules earlier in the list take precedence over rules later in the list:

1. If the arriving inner header is a Not-ECN-PDU it implies the L4 transport will not understand explicit congestion markings.  
Then:
  - \* If the outer header carries an explicit congestion marking, the packet SHOULD be dropped--the only indication of congestion that the L4 transport will understand.
  - \* If the outer is an ECN-PDU that carries no indication of congestion or a Not-ECN-PDU the PDU SHOULD be forwarded, but still as a Not-ECN-PDU.
2. If the outer header does not support explicit congestion notification (a Not-ECN-PDU), but the inner header does (an ECN-PDU), the inner header SHOULD be forwarded unchanged.
3. In some lower layer protocols congestion may be signalled as a numerical level, such as in the control frames of quantised congestion notification [IEEE802.1Qau]. If such a multi-bit encoding encapsulates an ECN-capable IP data packet, a function will be needed to convert the quantised congestion level into the frequency of congestion markings in outgoing IP packets.
4. Congestion indications may be encoded by a severity level. For instance increasing levels of congestion might be encoded by numerically increasing indications, e.g. pre-congestion notification (PCN) can be encoded in each PDU at three severity levels in IP or MPLS [RFC6660].

If the arriving inner header is an ECN-PDU, where the inner and outer headers carry indications of congestion of different

severity, the more severe indication SHOULD be forwarded in preference to the less severe.

5. The inner and outer headers might carry a combination of congestion notification fields that should not be possible given any currently used protocol transitions. For instance, if Encapsulation Guideline 3 in Section 4.3 had been followed, it should not be possible to have a less severe indication of congestion in the outer than in the inner. It MAY be appropriate to log unexpected combinations of headers and possibly raise an alarm.

If a safe outgoing codepoint can be defined for such a PDU, the PDU SHOULD be forwarded rather than dropped. Some implementers discard PDUs with currently unused combinations of headers just in case they represent an attack. However, an approach using alarms and policy-mediated drop is preferable to hard-coded drop, so that operators can keep track of possible attacks but currently unused combinations are not precluded from future use through new standards actions.

#### 4.5. Sequences of Similar Tunnels or Subnets

In some deployments, particularly in 3GPP networks, an IP packet may traverse two or more IP-in-IP tunnels in sequence that all use identical technology (e.g. GTP).

In such cases, it would be sufficient for every encapsulation and decapsulation in the chain to comply with RFC 6040. Alternatively, as an optimisation, a node that decapsulates a packet and immediately re-encapsulates it for the next tunnel MAY copy the incoming outer ECN field directly to the outgoing outer and the incoming inner ECN field directly to the outgoing inner. Then the overall behavior across the sequence of tunnel segments would still be consistent with RFC 6040.

Appendix C of RFC6040 describes how a tunnel egress can monitor how much congestion has been introduced within a tunnel. A network operator might want to monitor how much congestion had been introduced within a whole sequence of tunnels. Using the technique in Appendix C of RFC6040 at the final egress, the operator could monitor the whole sequence of tunnels, but only if the above optimisation were used consistently along the sequence of tunnels, in order to make it appear as a single tunnel. Therefore, tunnel endpoint implementations SHOULD allow the operator to configure whether this optimisation is enabled.

When ECN support is added to a subnet technology, consideration SHOULD be given to a similar optimisation between subnets in sequence if they all use the same technology.

#### 4.6. Reframing and Congestion Markings

The guidance in this section is worded in terms of framing boundaries, but it applies equally whether the protocol data units are frames, cells or packets.

Where framing boundaries are different between two layers, congestion indications SHOULD be propagated on the basis that a congestion indication on a PDU applies to all the octets in the PDU. On average, an encapsulator or decapsulator SHOULD approximately preserve the number of marked octets arriving and leaving (counting the size of inner headers, but not added encapsulating headers).

The next departing frame SHOULD be immediately marked even if only enough incoming marked octets have arrived for part of the departing frame. This ensures that any outstanding congestion marked octets are propagated immediately, rather than held back waiting for a frame no bigger than the outstanding marked octets--which might involve a long wait.

For instance, an algorithm for marking departing frames could maintain a counter representing the balance of arriving marked octets minus departing marked octets. It adds the size of every marked frame that arrives and if the counter is positive it marks the next frame to depart and subtracts its size from the counter. This will often leave a negative remainder in the counter, which is deliberate.

#### 5. Feed-Up-and-Forward Mode: Guidelines for Adding Congestion Notification

The guidance in this section is applicable when IP packets:

- o are encapsulated in Ethernet headers;
- o are forwarded by the eNode-B (base station) of a 3GPP radio access network, which is required to apply ECN marking during congestion [LTE-RA].

This guidance also generalises to encapsulation by other subnet technologies with no native support for explicit congestion notification at the lower layer, but with support for finding and processing an IP header. It is unlikely to be applicable or necessary for IP-in-IP encapsulation, where feed-forward-and-up mode based on [RFC6040] would be more appropriate.

Marking the IP header while switching at layer-2 (by using a layer-3 switch) or while forwarding in a radio access network seems to represent a layering violation. However, it can be considered as a benign optimisation if the guidelines below are followed. Feed-up-and-forward is certainly not a general alternative to implementing feed-forward congestion notification in the lower layer, because:

- o IPv4 and IPv6 are not the only layer-3 protocols that might be encapsulated by lower layer protocols
- o Link-layer encryption might be in use, making the layer-2 payload inaccessible
- o Many Ethernet switches do not have 'layer-3 switch' capabilities so they cannot read or modify an IP payload
- o It might be costly to find an IP header (v4 or v6) when it may be encapsulated by more than one lower layer header, e.g. Ethernet MAC in MAC [IEEE802.1Qah].

Nonetheless, configuring lower layer equipment to look for an ECN field in an encapsulated IP header is a useful optimisation. If the implementation follows the guidelines below, this optimisation does not have to be confined to a controlled environment such as within a data centre; it could usefully be applied on any network--even if the operator is not sure whether the above issues will never apply:

1. If a native lower-layer congestion notification mechanism exists for a subnet technology, it is safe to mix feed-up-and-forward with feed-forward-and-up on other switches in the same subnet. However, it will generally be more efficient to use the native mechanism.
2. The depth of the search for an IP header SHOULD be limited. If an IP header is not found soon enough, or an unrecognised or unreadable header is encountered, the switch SHOULD resort to an alternative means of signalling congestion (e.g. drop, or the native lower layer mechanism if available).
3. It is sufficient to use the first IP header found in the stack; the egress of the relevant tunnel can propagate congestion notification upwards to any more deeply encapsulated IP headers later.

## 6. Feed-Backward Mode: Guidelines for Adding Congestion Notification

It can be seen from Section 3.3 that congestion notification in a subnet using feed-backward mode has generally not been designed to be directly coupled with IP layer congestion notification. The subnet attempts to minimise congestion internally, and if the incoming load at the ingress exceeds the capacity somewhere through the subnet, the layer 3 buffer into the ingress backs up. Thus, a feed-backward mode subnet is in some sense similar to a null mode subnet, in that there is no need for any direct interaction between the subnet and higher layer congestion notification. Therefore no detailed protocol design guidelines are appropriate. Nonetheless, a more general guideline is appropriate:

1. A subnetwork technology intended to eventually interface to IP SHOULD NOT be designed using only the feed-backward mode, which is certainly best for a stand-alone subnet, but would need to be modified to work efficiently as part of the wider Internet, because IP uses feed-forward-and-up mode.

The feed-backward approach at least works beneath IP, where the term 'works' is used only in a narrow functional sense because feed-backward can result in very inefficient and sluggish congestion control--except if it is confined to the subnet directly connected to the original data source, when it is faster than feed-forward. It would be valid to design a protocol that could work in feed-backward mode for paths that only cross one subnet, and in feed-forward-and-up mode for paths that cross subnets.

In the early days of TCP/IP, a similar feed-backward approach was tried for explicit congestion signalling, using source-quench (SQ) ICMP control packets. However, SQ fell out of favour and is now formally deprecated [RFC6633]. The main problem was that it is hard for a data source to tell the difference between a spoofed SQ message and a quench request from a genuine buffer on the path. It is also hard for a lower layer buffer to address an SQ message to the original source port number, which may be buried within many layers of headers, and possibly encrypted.

Quantised congestion notification (QCN--also known as backward congestion notification or BCN) [IEEE802.1Qau] uses a feed-backward mode structurally similar to ATM's relative rate mechanism. However, QCN confines its applicability to scenarios such as some data centres where all endpoints are directly attached by the same Ethernet technology. If a QCN subnet were later connected into a wider IP-based internetwork (e.g. when attempting to interconnect multiple data centres) it would suffer the inefficiency shown Figure 3.

## 7. IANA Considerations (to be removed by RFC Editor)

This memo includes no request to IANA.

## 8. Security Considerations

If a lower layer wire protocol is redesigned to include explicit congestion signalling in-band in the protocol header, care SHOULD be taken to ensure that the field used is specified as mutable during transit. Otherwise interior nodes signalling congestion would invalidate any authentication protocol applied to the lower layer header--by altering a header field that had been assumed as immutable.

The redesign of protocols that encapsulate IP in order to propagate congestion signals between layers raises potential signal integrity concerns. Experimental or proposed approaches exist for assuring the end-to-end integrity of in-band congestion signals, e.g.:

- o Congestion exposure (ConEx) for networks to audit that their congestion signals are not being suppressed by other networks or by receivers, and for networks to police that senders are responding sufficiently to the signals, irrespective of the transport protocol used [I-D.ietf-conex-abstract-mech].
- o The ECN nonce [RFC3540] for a TCP sender to detect whether a network or the receiver is suppressing congestion signals.
- o A test with the same goals as the ECN nonce, but without the need for the receiver to co-operate with the protocol [I-D.moncaster-tcpm-rcv-cheat].

Given these end-to-end approaches are already being specified, it would make little sense to attempt to design hop-by-hop congestion signal integrity into a new lower layer protocol, because end-to-end integrity inherently achieves hop-by-hop integrity.

## 9. Conclusions

Following the guidance in the document enables ECN support to be extended to numerous protocols that encapsulate IP (v4 & v6) in a consistent way, so that IP continues to fulfil its role as an end-to-end interoperability layer. This includes:

- o A wide range of tunnelling protocols with various forms of shim header between two IP headers;

- o A wide range of subnet technologies, particularly those that work in the same 'feed-forward-and-up' mode that is used to support ECN in IP and MPLS.

Guidelines have been defined for supporting propagation of ECN between Ethernet and IP on so-called Layer-3 Ethernet switches, using a 'feed-up-and-forward' mode. This approach could enable other subnet technologies to pass ECN signals into the IP layer, even if they do not support ECN natively.

Finally, attempting to add ECN to a subnet technology in feed-backward mode is deprecated except in special cases, due to its likely sluggish response to congestion.

## 10. Acknowledgements

Thanks to Gorry Fairhurst for extensive reviews. Thanks also to the following reviewers: Ingemar Johansson and Piers O'Hanlon and Michael Welzl, who pointed out that lower layer congestion notification signals may have different semantics to those in IP.

Bob Briscoe was part-funded by the European Community under its Seventh Framework Programme through the Trilogy project (ICT-216372) for initial drafts and through the Reducing Internet Transport Latency (RITE) project (ICT-317700) subsequently. The views expressed here are solely those of the authors.

## 11. Comments Solicited

Comments and questions are encouraged and very welcome. They can be addressed to the IETF Transport Area working group mailing list <tsvwg@ietf.org>, and/or to the authors.

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.

- [RFC4774] Floyd, S., "Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field", BCP 124, RFC 4774, November 2006.
- [RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, January 2008.
- [RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, November 2010.

## 12.2. Informative References

- [ATM-TM-ABR] Cisco, "Understanding the Available Bit Rate (ABR) Service Category for ATM VCs", Design Technote 10415, June 2005.
- [Buck00] Buckwalter, J., "Frame Relay: Technology and Practice", Pub. Addison Wesley ISBN-13: 978-0201485240, 2000.
- [DCTCP] Alizadeh, M., Greenberg, A., Maltz, D., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and M. Sridharan, "Data Center TCP (DCTCP)", ACM SIGCOMM CCR 40(4)63--74, October 2010, <<http://portal.acm.org/citation.cfm?id=1851192>>.
- [GTPv1-U] 3GPP, "General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)", Technical Specification TS 29.281, .
- [GTPv1] 3GPP, "GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface", Technical Specification TS 29.060, .
- [GTPv2-C] 3GPP, "Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C)", Technical Specification TS 29.274, .
- [I-D.ietf-conex-abstract-mech] Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx) Concepts and Abstract Mechanism", draft-ietf-conex-abstract-mech-08 (work in progress), October 2013.
- [I-D.moncaster-tcpm-rcv-cheat] Moncaster, T., "A TCP Test to Allow Senders to Identify Receiver Non-Compliance", draft-moncaster-tcpm-rcv-cheat-01 (work in progress), June 2007.



## [IEEE802.1Qah]

IEEE, "IEEE Standard for Local and Metropolitan Area Networks--Virtual Bridged Local Area Networks--Amendment 6: Provider Backbone Bridges", IEEE Std 802.1Qah-2008, August 2008, <<http://www.ieee802.org/1/pages/802.1ah.html>>.

(Access Controlled link within page)

## [IEEE802.1Qau]

Finn, N., Ed., "IEEE Standard for Local and Metropolitan Area Networks--Virtual Bridged Local Area Networks - Amendment 13: Congestion Notification", IEEE Std 802.1Qau-2010, March 2010, <<http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5454061>>.

(Access Controlled link within page)

## [ITU-T.I.371]

ITU-T, "Traffic Control and Congestion Control in B-ISDN", ITU-T Rec. I.371 (03/04), March 2004, <<http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5454061>>.

[LTE-RA] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2", Technical Specification TS 36.300, .

[RFC1323] Jacobson, V., Braden, B., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[RFC1701] Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 1701, October 1994.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

[RFC2637] Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W., and G. Zorn, "Point-to-Point Tunneling Protocol", RFC 2637, July 1999.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.
- [RFC2884] Hadi Salim, J. and U. Ahmed, "Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks", RFC 2884, July 2000.
- [RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.
- [RFC3540] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", RFC 3540, June 2003.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC6633] Gont, F., "Deprecation of ICMP Source Quench Messages", RFC 6633, May 2012.
- [RFC6660] Briscoe, B., Moncaster, T., and M. Menth, "Encoding Three Pre-Congestion Notification (PCN) States in the IP Header Using a Single Diffserv Codepoint (DSCP)", RFC 6660, July 2012.
- [trill-rbridge-options] Eastlake, D., Ghanwani, A., Manral, V., and C. Bestler, "RBridges: Further TRILL Header Extensions", draft-ietf-trill-rbridge-options-07 (work in progress), June 2012.
- [vxlan] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-08 (work in progress), February 2014.

## Appendix A. Outstanding Document Issues

1. [GF] Concern that certain guidelines warrant a MUST (NOT) rather than a SHOULD (NOT). Given the guidelines say that if any SHOULD (NOT)s are not followed, a strong justification will be needed, they have been left as SHOULD (NOT) pending further list discussion. In particular:
  - \* If inner is a Not-ECN-PDU and Outer is CE (or highest severity congestion level), MUST (not SHOULD) drop?
2. Consider whether an IETF Standard Track doc will be needed to Update the IP-in-IP protocols listed in Section 4.1--at least those that the IET

## Appendix B. Changes in This Version (to be removed by RFC Editor)

From briscoe-03 to 04:

- \* Re-arranged the introduction to describe the purpose of the document first before introducing ECN in more depth. And clarified the introduction throughout.
- \* Added applicability to 3GPP TS 36.300.

From briscoe-02 to 03:

- \* Scope section:
  - + Added dependence on correct propagation of traffic class information
  - + For the feed-backward mode, deemed multicast and anycast out of scope
- \* Ensured all guidelines referring to subnet technologies also refer to tunnels and vice versa by adding applicability sentences at the start of sections 4.1, 4.2, 4.3, 4.4, 4.6 and 5.
- \* Added Security Considerations on ensuring congestion signal fields are classed as immutable and on using end-to-end congestion signal integrity technologies rather than hop-by-hop.

From briscoe-01 to 02:

- \* Added authors: JK & PT

- \* Added
  - + Section 4.1 "IP-in-IP Tunnels with Tightly Coupled Shim Headers"
  - + Section 4.5 "Sequences of Similar Tunnels or Subnets"
  - + roadmap at the start of Section 4, given the subsections have become quite fragmented.
  - + Section 9 "Conclusions"
- \* Clarified why transports are starting to be able to saturate interior links
- \* Under Section 1.1, addressed the question of alternative signal semantics and included multicast & anycast.
- \* Under Section 3.1, included a 3GPP example.
- \* Section 4.2. "Wire Protocol Design":
  - + Altered guideline 2. to make it clear that it only applies to the immediate subnet egress, not later ones
  - + Added a reminder that it is only necessary to check that ECN propagates at the egress, not whether interior nodes mark ECN
  - + Added example of how QCN uses 802.1p to indicate support for QCN.
- \* Added references to Appendix C of RFC6040, about monitoring the amount of congestion signals introduced within a tunnel
- \* Appendix A: Added more issues to be addressed, including plan to produce a standards track update to IP-in-IP tunnel protocols.
- \* Updated acks and references

From briscoe-00 to 01:

- \* Intended status: BCP (was Informational) & updates 3819 added.
- \* Briefer Introduction: Introductory para justifying benefits of ECN. Moved all but a brief enumeration of modes of operation

to their own new section (from both Intro & Scope). Introduced incr. deployment as most tricky part.

- \* Tightened & added to terminology section
- \* Structured with Modes of Operation, then Guidelines section for each mode.
- \* Tightened up guideline text to remove vagueness / passive voice / ambiguity and highlight main guidelines as numbered items.
- \* Added Outstanding Document Issues Appendix
- \* Updated references

#### Authors' Addresses

Bob Briscoe  
BT  
B54/77, Adastral Park  
Martlesham Heath  
Ipswich IP5 3RE  
UK

Phone: +44 1473 645196  
EMail: bob.briscoe@bt.com  
URI: <http://bobbbriscoe.net/>

John Kaippallimalil  
Huawei  
5340 Legacy Drive, Suite 175  
Plano, Texas 75024  
USA

EMail: [john.kaippallimalil@huawei.com](mailto:john.kaippallimalil@huawei.com)

Pat Thaler  
Broadcom Corporation  
5025 Keane Drive  
Carmichael, CA 95608  
USA

EMail: [pthaler@broadcom.com](mailto:pthaler@broadcom.com)

TSVWG  
Internet-Draft  
Intended Status: Informational  
Expires: August 25, 2013

K. Carlberg  
G11  
P. O'Hanlon  
UCL  
Feb 25, 2013

Reactions to Signaling from ECN Support for RTP/RTCP  
<draft-carlberg-tsvwg-ecn-reactions-04.txt>

#### Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 25, 2013.

#### Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Abstract

This document presents an examination of various responses to Congestion Experience (CE) notifications by real time applications that have negotiated end-to-end support of Explicit Congestion Notification (ECN). This document is a follow-on effort of [rfc6679], which specifies the signaling used to provide ECN support for RTP/RTCP flows.

## 1. Introduction

This document presents an examination of various responses to Congestion Experience (CE) notifications by real time applications that have negotiated end-to-end support of Explicit Congestion Notification (ECN). [rfc6679] defines the signaling for support of ECN by RTP based sessions and also covers the case where a set of nodes do not respond to CE notifications. A more detailed discussion about how back-off algorithms can be achieved, as well as other potential reactions, is viewed as out of scope of that document and may be addressed by a companion document.

### 1.1 Background

ECN is a mechanism used to explicitly signal the presence of congestion without relying on packet loss. It was initially designed using a dual layer signaling model; negotiation and feedback at the transport layer, and downstream notification of congestion at the network layer. For IP, a new two bit field was used to both indicate the successful negotiated support for ECN signaling, as well as indicate the presence of congestion via the CE flag. In the case of TCP [rfc3168], a new TCP header flag was defined that provides upstream end-to-end indication of congestion occurring somewhere along the downstream path.

There should be no difference in congestion response if ECN-CE marks or packet drops are detected. However it is noted that there MAY be other reactions to ECN-CE specified in the future. Such an alternative reaction MUST be specified and considered to be safe for deployment under any restrictions specified. We specify such an alternative in this document.

With respect to ECN for TCP, [rfc3168] specifies an indication of congestion, but it does so once per Round Trip Time (RTT). [rfc6679] is an effort that proposes a finer grained notification reflecting a more accurate indication of the number of ECN marked packets received within one RTT. It should be noted that there is also other on going work to provide more accurate ECN feedback information for TCP [draft-tcpm-accecn-reqs].

### 1.2 Terminology and Abbreviations

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [RFC2119].

## 2. Issues

The initial discussions and presentation of [draft-rtp-ecn] produced a consensus that the specification of signaling was to be done within the AVTcore working group, and any subsequent discussion on end-to-end reactions to the signaling would be accomplished in the Transport Services (TSV) working group. This draft satisfies the latter effort.

Another issue that needs to be recognized is that the reactions to CE in the context of [rfc6679] are the responsibility of the application. This is in contrast to ECN support for TCP, where explicit signaled feedback of, and reaction to, CE is kept transparent to the application. The issue of placing the feedback responsibility in the application is that each application needs to add specific support for that reaction. On the other hand, multiple reactions may be considered by the application. For this reason, [rfc6679] states the need for a default congestion control reaction that MUST be supported. Section 3 through 5 expands on this topic.

### 3. Congestion Control Algorithms

The transport of any data flow across the Internet produces a need for some form of congestion control to attain a suitable share of the capacity of the path through a network. Most of the existing work on realtime congestion control algorithms has been rooted in TCP-friendly approaches but with smoother adaptation cycles. TCP congestion control is unsuitable for interactive media for a number of reasons including the fact that it is loss-based so it maximizes the latency on a path, it changes its transmit rate to quickly for multimedia, and favors reliability over timeliness. In the case of real time media transport, one requires:

Smoother rate variation: (than for bulk data) to accommodate the underlying media flow's characteristics.

Low latency: Maintaining latencies sufficient to be usable, where 150ms is understood to be a good target [ITU.G114.2003].

Burst handling: Ability to handle bursts due to the nature of the media and codec (e.g. I-frames etc)

#### 3.1 TCP Friendly Rate Control (TFRC)

TFRC has a smoother response to congestion than TCP-like approaches, thus making it more suitable for real-time interactive multimedia applications. It has been cited in a number of other documents within the IETF for use with UDP and media flows [rfc3714, bcpl45] and is seeing full and partial deployment in related solutions such as Empathy/Farsight, and GoogleTalk [googl].



However it should be noted that TFRC is only recommended for real-time media use with ECN response. TFRC is not recommended for non-ECN paths due to its loss based operation which leads to full queues with maximised latencies. It is assumed that ECN markings will usually occur with lower queue occupancy and thus lower latency. However it is understood that ECN marks may not provide for sufficiently low latencies in some situations so other congestion control solutions would be preferable.

[rfc4342] specifies the profile for TFRC for use in the Datagram Congestion Control Protocol (DCCP) [rfc4340] for a half connection. A DCCP half connection is defined as application data sent downstream with corresponding acknowledgements sent upstream. These half-connections can be realized in the form of one-way pre-recorded media, one-way live media, or two-way interactive. A perceived drawback in this profile concerns its application to interactive media that use small packets. [RFC4828] is an experimental protocol defining a variation of TFRC used to address this drawback and achieve the same bandwidth as a TCP flow using packets of size 1500 bytes.

[rfc6679] is an standard that specifies how RTP flows can be supported using the RTP/AVPF profile and the general RTP header extension mechanism.

### 3.2 Related Work

#### 3.2.1 3GPP

Outside of this previous and on-going work with TFRC, it is understood that some parties have issues with the behavior of TFRC under certain conditions. A notable mention of this is made in the 3GPP's document on IP Multimedia Subsystem (IMS) Media handling and interaction [TR26.114], where it is mentioned:

"Note that for IMS networks, which normally have nonzero packet loss and fairly long round-trip delay, the amount of bitrate reduction specified in RFC 3448 is generally too restrictive for video and may, if used as specified, result in very low video bitrates already at (for IMS) moderate packet loss rates."

Though it is unclear exactly what the 3GPP community consider as too restrictive and whether some alteration of the response may be suitable. It should be noted that the 3GPP document only referred to an older version of TFRC defined in [RFC3448]. Given that the current version of TFRC [RFC5348] has made significant changes to the idle and data-limited responses it is unclear whether their assessment is relevant to current TFRC implementations.

Furthermore the specification [TR26.114] only outlines a rudimentary approach to congestion control, providing an example of a 60% back-off reaction to loss within an RTCP reporting period. The proposed signalling employs Temporary Maximum Media Stream Bit Rate Request (TMMBR) [RFC5104] and Codec Mode Request (CMR) [RFC4867] for video and audio respectively, which would only provide for very basic rate control if used as specified. We note that [TR26.114] specifies terminal behavior, while [TS36.300] specifies base station behaviour, though neither specify any standardised congestion control approach.

It is understood that there are a number of proprietary and patented approaches that provide more sophisticated response in the case of 3G/LTE, but since these are neither endorsed nor standardized this document advocates a standardized approach such as TFRC.

We also acknowledge that there are many congestion control algorithms available for implementers to choose from, with a subset that are specifically suited to real time media transmission. However, given a variety of real time applications and their various characteristics (sender-only broadcast, interactive unicast, etc), we need to expand the notion of how back-off can be achieved. Hence, the focus needs to be on an output that would resemble the characteristics of TFRC.

### 3.2.2 RTCweb

Within the RTCweb Working Group the need for a more media friendly congestion control mechanism has been made apparent. Currently, TFRC is perceived as having deficiencies (e.g. its loss-based design, lack of cross-stream congestion control functionality etc) that make it an incomplete or insufficient solution for the envisioned RTCWEB media flows. The RTP Media Congestion Avoidance Techniques (rmcat) working group has now been formed which aims to lead to the formation of a working group on these issues. The group aims to develop one or more congestion control algorithms, associated extensions, and evaluation criteria. Furthermore it has been proposed that certain practices, such as 'circuit-breaker' conditions, to provide operational limits on congestion control algorithms, and feedback messages, may be tackled in other groups such as AVTCORE and AVTEXT respectively.

Thus there is some movement to attempt to develop new algorithms better suited to media transport, but these efforts will clearly take a considerable time to reach fruition.

### 3.3 ECN response

As mentioned above and in accordance to [rfc3168], the actual response to the reception of an ECN-CE marked packet MUST normally be the same as that of a lost packet. However there are a number of contexts where one

may also be interested in more varied approaches. We expand on this in Section 5 below.

#### 4. Application Layer Congestion Response

Whilst the congestion control algorithm may decide to alter the rate at which the application should operate, in the case of media applications this process is not as straightforward as the case of bulk data. The different media engines and codecs in use may only have limited adaptation ranges, thus, this limitation needs to be a consideration when adapting the rate. Furthermore the application needs to be aware of the capability of the specific codecs in terms of their ability to switch configuration mid-stream (without loss of fidelity), which may impose further limits on the modes of operation.

One approach for achieving a lower generation of data is through reduced sampling of the media (e.g., voice or video). In the case of video, this may also involve slower frame rates. Specific recommendations that describe how applications should respond to congestion in the context of supporting the algorithmic characteristics of a congestion control algorithm are outside the scope of this document.

#### 5. Other Reactions

In addition to the activation of congestion control algorithm, other reactions can be used or leveraged by an application in response to CE. We divide these other potential reactions into three categories: signaling, fault tolerance, and reduction. In the first two cases, we note that these other reactions are considered symmetric because they require downstream peer support. We also point out that activation of other reactions represents an example of an on-demand and as-needed approach in responding to CE.

In each case, we discuss issues that should be considered when contemplating a different reaction in the presence of CE feedback.

##### 5.1 Signaling

###### 5.1.1 RSVP

The resource Reservation Protocol (RSVP) can be used to signal a desired set of path characteristics (e.g., bandwidth, delay) in response to CE feedback [rfc2205]. Its operation is based on the use of PATH messages sent downstream hop-by-hop from the source to a destination that specify requested forwarding characteristics. In return, the destination sends a hop-by-hop RESV message upstream towards the source confirming the resources that have been reserved for that flow.

[rfc3181] defines a priority policy element that specifies both an allocation and defending priority. This dual specification supports the use of preemption of existing reservations. [draft-priority-rsvp] is a work-in-progress that defines a new policy element that only conveys priority during reservation establishment. This latter effort also presents several reservation models, including one that describes engineered resources set aside for priority users.

#### 5.1.1.1 Issues

As discussed in [rfc-3583], RSVP presents a difficult challenge of establishing state and effectively and efficiently migrating it during roaming in mobile environments. Its soft state design allows the protocol to attempt re-establishment of reserved resources along new path(s), but there is no guarantee that resources along the new path will be available. In addition, there is at least 1 RTT of delay and the delta in initiating a new PATH message that delays reservation establishment.

Some user groups, such as those found in the military, make a distinction between mobile and transportable environments. The former case resembles scenarios attributed to Mobile IP. The latter case is characterized by wireless hosts operating in a new location, but never moving to the extent that new paths through a network need to be established. In this latter example, the challenges of RSVP in a wireless environment are diminished. In addition, these environments tend to involve a single administrative control of both hosts and routing/forwarding nodes within a network infrastructure.

RSVP is associated with a means of retaining a minimal bound of forwarding characteristics per flow, or aggregate of flows. As such, it can be considered to run contrary to the objectives of ECN. However, in cases where some flows must be reserved, CE feedback could be used to signal the need to lower a pre-existing killer app reservation.

#### 5.1.2 Differentiated Services

Unlike RSVP and its use of a separate signaling mechanism to reserve resources, Differentiated Services (diff-serv) uses code points within the IP header to convey the forwarding behavior of that packet [rfc2474]. This may range from various drop precedence values to a code point that signifies low delay and low loss (i.e., characteristics attributed to real time flows).

As in the case of RSVP, applications could rely on the reception of CE feedback to initiate a subsequent setting of diff-serv code points to provide additional protection or explicit association of forwarding characteristics of a given flow of packets. In addition, the setting of

diff-serv code points would be done on an as-needed basis in reaction to CE feedback. Recommendations concerning specific diff-serv values are outside the scope of this document.

#### 5.1.2.1 Issues

Given the ease by which applications or middle boxes can set diff-serv code points, the issue of trusting values other than best effort can become problematic when hosts and routing/forwarding nodes are not associated with a single administrative authority.

As in the case of RSVP, the effectiveness of diff-serv is dependent on the number of nodes along a path that support the protocol. Thus, as opposed to a single end-point reaction to CE feedback, differentiated services requires additional support in the network to either increase or decrease the probability of traffic being forwarded to its destination.

A symbiotic capability to consider is the use of on-demand/as-needed diff-serv code points to trigger downstream actions by the network. A specific example would be a diff-serv code point sent in reaction to CE feedback that could trigger alternate path routing via MPLS.

#### 5.2 Fault Tolerance

Fault tolerance is another category of reactions that may be used by applications in response to CE feedback. In some cases, these efforts may contribute to an increase in traffic load in order to add protection and resiliency to a flow.

**Redundant Transmissions:** This approach is based on a source sending duplicate payloads that can be used to compensate for lost packets. Its positive value may emerge in cases where a path has several downstream congestion points that increase the probability that a packet will be dropped instead of marked as CE and forwarded downstream.

**Application Layer Forward Error Correction (FEC):** This approach also adds additional overhead to the flow in order to compensate for potential packet loss. And as the case of redundant transmissions, the value of this approach can be realized when there exists multiple downstream congestion points that increase the probability of dropping packets. However, the impact of the overhead is minimized by having one (or a few) additional packet(s) used to compensate for the loss of a set of packets.

**Codec Swapping:** This approach involves changing codecs to either reduce load or achieve an improvement in compensating for lost packets. Depending on the codec, the reduction of load may be a simple step

function, or it may involve a gradual and variable reduction in load based on the rate of congestion feedback received by the source.

Interweaving packets: To Be Done (based on research at UCL)

#### 5.2.1 Issues

The use of redundant transmissions or FEC produces a detrimental impact of contributing to an increase in load and the measure of congestion that triggers CE feedback. In the case of FEC, additional delay is typically incurred through the generation of X amount of erasure packets for each set of original source packets. And while an initial increase in QoS may be observed for these flows, the overall rate of congestion can be expected to increase.

Swapping codecs based on the reception of CE feedback has the positive affect of reducing load at the risk of reducing perceived QoS by the user. As in the case of all options described above regarding fault tolerance, the ability to change to a different codec is depending on end-to-end peer support. In addition, there is no assurance that the different codec reduces load in relation to the amount of congestion experienced over time.

#### 5.3 Alternative Reaction for Emergency Communications

As mentioned in [rfc6679], the default reaction on the reception of these ECN-CE marked packets MUST be to provide the congestion control algorithm with a congestion notification that triggers the algorithm to react as if packet loss had occurred. There MAY be an alternative reaction if it is considered safe for deployment. An example of the need for an alternative reaction would be the case of Emergency Telecommunications Service (ETS) [rfc3689, rfc4190], where an improvement in QoS or a higher probability of session establishment and forwarding of traffic is of high interest.

It is proposed that certain authorized ETS flows may be permitted to employ either a substantially less aggressive back-off algorithm than the default algorithm, or some level of exemption from reacting to ECN marked packets. This alternative reaction will benefit these flows as the marks would normally be considered as equivalent to lost packets, which would effectively increase the loss level, which in turn will generally result in the reduction of flow rate. This applies to all flows that utilize some form of the rate control that is inversely proportional to the loss rate, which includes TCP-like algorithms or equation-based approaches.

Simulations of the use of ECN exemption with TFRC and have found that it has limited effect on the normal flows with low numbers of exempt flows. A half-dumbbell network was used with a RED router queue configured using the

settings recommended by Sally Floyd. The candidate flows are 1Mbit/s each with a backhaul 100Mbit/s link. In the standard case where 1% of flows would be exempt the remaining flows achieve 99.99% of the bandwidth that they would achieve without the presence of the exempt flows. This is what would be expected from the simple calculation of the allocation, given that the exempt flows achieve their full rate (1Mbit/s); With 100 normal plus 1 exempt flow, assuming that the exempt flow uses 1Mbit/s, the remaining capacity is 99Mbit/s which is divided between the 100 normal flows. Whilst when 101 normal flows are run over the 100Mbit/s link they would have to share it evenly, so it work

s out thus:  $((99/100)/(100/101))*100=99.99\%$ . In the case of 5% exempt flows then the proportion is very slightly lower at  $((95/100)/(100/105))*100=99.75\%$ . Bot

h these calculations are borne out in the simulation runs.

The level of exemption employed can be altered in a number of ways. Two simple approaches would be to either set a threshold number of ECN marked packets tha

t could be considered as a loss, and another approach would be to set a percentage threshold of ECN marked packet that would be considered as a loss.

It should be noted that in the simulations the end-to-end delay of the packets within the flows was monitored and the relative delay of the exempt flows apparently rises somewhat when exemption is enacted. However what is actually occurring is that the 'normal' flows are reducing their throughput and are thu

s reducing their latency somewhat. There is normally some limited latency when using loss-based techniques such as TFRC because it fills the queues to ascertain the link capacity and maintains that level of delay throughout a session. However the level of latency is clearly limited by the queue sizes in the network and on media specific links these queue sizes are typically quite small, so the resulting latency is limited.

Furthermore in the case where media flows employing TFRC, or any other congestion control algorithm (e.g. delay-based), are sharing a bottleneck link with TCP flows then the queues will be filled by the TCP flows and the latency will be kept near or at a their maximum despite any other flows.

### 5.3.1 Issues

To Be Done

## 6. IANA Considerations

This document requires no actions from IANA.

## 7. Security Considerations

The reliance on accurate and un-modified RTCP information means that SRTP needs to be used, or any other mechanism that helps prevent modification of RTCP feedback packets.

## 8. Acknowledgements

TBD

## 9. References

### 9.1 Normative

- [rfc2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [rfc2205] Braden, B., et. al., "Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification", RFC 2205, September 1997
- [rfc2209] Braden, R., L. Zhang, "Resource Reservation Protocol (RSVP) Version 1 Message Processing Rules", RFC2209 September 1997
- [rfc2474] Nichols, K., et. al., "Definition of the Differentiated Services Field in the IPv4 and IPv6 Headers", RFC 2474, December 1998
- [rfc3168] Ramakrishnan, K., et. al., "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September, 2001
- [rfc3181] Herzog, S., "Signaled Preemption Priority Policy Element", RFC 3181, October 2001
- [rfc3448] Handley, M., et. al., "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003
- [rfc3583] Chaskar, H., "Requirements of a Quality of Service (QoS) Solution for Mobile IP", RFC 3583, September 2003
- [rfc4867] Sjöberg, J., et. al., "RTP Payload Format and File Storage Format for the AMR and AMR-WB Audio Codecs", RFC 4867, April 2007
- [rfc5104] Wenger, S., et. al., "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008
- [rfc6679] Westerlund, M., et. al., "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, IETF, Aug 2012



## 9.2 Informative

- [draft-rtp-tfrc] Gharai, L., C. Perkins, "RTP with TCP Friendly Rate Control", work-in-progress, Sept 2011
- [draft-tcpm-accecn-reqs] M. Kuehlewind, R. Scheffenegger, "Problem Statement and Requirements for a More Accurate ECN Feedback", work-in-progress, Feb 2013
- [Googl] [http://code.google.com/apis/talk/call\\_signaling.html](http://code.google.com/apis/talk/call_signaling.html)
- [tr26.114] "IMS; Multimedia telephony; Media Handling and Interaction", 3GPP, version 10, April 2011
- [ts36.300] "E-UTRA and E-UTRAN Overall Description, Stage 2", 3GPP, Release 10, September, 2011
- [rfc4340] Kohler, E., et. al, Datagram Congestion Control Protocol (DCCP), RFC4340, March 2006
- [rfc4342] Floyd, S., et. al., "Profile for DCCP Congestion Control ID 3: TFRC", RFC 4342, March 2006
- [rfc4828] Floyd, S., E. Kohler, "TFRC: The Small Packet Variant", RFC 4828, April 2007
- [rfc3689] Carlberg, K., Atkinson, R., "General Requirements for Emergency Telecommunications Service (ETS)", RFC 3689, February 2004
- [rfc4190] Carlberg, K. et, al., "Framework for Supporting Emergency Telecommunications Service (ETS) in IP Telephony", RFC 4190, November 2005
- [rfc3714] Floyd, S., Kempf, J., "IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet", RFC 3714, March 2004
- [bcp145] Eggert, L., Fairhurst, G., "Unicast UDP Usage Guidelines for Application Designers", RFC 5405, BCP 145, November 2008
- [ITU.G114.2003]  
International Telecommunications Union, "One-way transmission time", ITU-T Recommendation G.707, May 2003.

## Author's Addresses

Piers O'Hanlon

University of Oxford  
Oxford Internet Institute  
1 St Giles  
Oxford OX1 3JS  
United Kingdom

Email: piers.ohanlon@oii.ox.ac.uk

Ken Carlberg  
G11  
1600 Clarendon Blvd  
Arlington VA  
USA

Email: carlberg@g11.org.uk

Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: August 29, 2013

G. Fairhurst  
University of Aberdeen  
M. Westerlund  
Ericsson  
February 25, 2013

Applicability Statement for the use of IPv6 UDP Datagrams with Zero  
Checksums  
draft-ietf-6man-udpzero-12

Abstract

This document provides an applicability statement for the use of UDP transport checksums with IPv6. It defines recommendations and requirements for the use of IPv6 UDP datagrams with a zero UDP checksum. It describes the issues and design principles that need to be considered when UDP is used with IPv6 to support tunnel encapsulations and examines the role of the IPv6 UDP transport checksum. The document also identifies issues and constraints for deployment on network paths that include middleboxes. An appendix presents a summary of the trade-offs that were considered in evaluating the safety of the update to RFC 2460 that updates use of the UDP checksum with IPv6.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 29, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

Provisions Relating to IETF Documents  
 (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
1.1. Document Structure . . . . .	5
1.2. Terminology . . . . .	5
1.3. Use of UDP Tunnels . . . . .	5
1.3.1. Motivation for new approaches . . . . .	6
1.3.2. Reducing forwarding cost . . . . .	6
1.3.3. Need to inspect the entire packet . . . . .	7
1.3.4. Interactions with middleboxes . . . . .	7
1.3.5. Support for load balancing . . . . .	8
2. Standards-Track Transports . . . . .	9
2.1. UDP with Standard Checksum . . . . .	9
2.2. UDP-Lite . . . . .	9
2.2.1. Using UDP-Lite as a Tunnel Encapsulation . . . . .	10
2.3. General Tunnel Encapsulations . . . . .	10
2.4. Relation to UDP-Lite and UDP with checksum . . . . .	10
3. Issues Requiring Consideration . . . . .	12
3.1. Effect of packet modification in the network . . . . .	13
3.1.1. Corruption of the destination IP address . . . . .	14
3.1.2. Corruption of the source IP address . . . . .	15
3.1.3. Corruption of Port Information . . . . .	16
3.1.4. Delivery to an unexpected port . . . . .	16
3.1.5. Corruption of Fragmentation Information . . . . .	17
3.2. Where Packet Corruption Occurs . . . . .	19
3.3. Validating the network path . . . . .	20
3.4. Applicability of method . . . . .	21
3.5. Impact on non-supporting devices or applications . . . . .	21
4. Constraints on implementation of IPv6 nodes supporting zero checksum . . . . .	22
5. Requirements on usage of the zero UDP checksum . . . . .	24
6. Summary . . . . .	26
7. Acknowledgements . . . . .	28
8. IANA Considerations . . . . .	28
9. Security Considerations . . . . .	28
10. References . . . . .	29
10.1. Normative References . . . . .	29
10.2. Informative References . . . . .	29

Appendix A. Evaluation of proposal to update RFC 2460 to	
support zero checksum . . . . .	31
A.1. Alternatives to the Standard Checksum . . . . .	31
A.2. Comparison . . . . .	33
A.2.1. Middlebox Traversal . . . . .	33
A.2.2. Load Balancing . . . . .	34
A.2.3. Ingress and Egress Performance Implications . . . . .	34
A.2.4. Deployability . . . . .	34
A.2.5. Corruption Detection Strength . . . . .	35
A.2.6. Comparison Summary . . . . .	35
Appendix B. Document Change History . . . . .	38
Authors' Addresses . . . . .	41

## 1. Introduction

The User Datagram Protocol (UDP) [RFC0768] transport is defined for the Internet Protocol (IPv4) [RFC0791] and is defined in "Internet Protocol, Version 6 (IPv6) [RFC2460] for IPv6 hosts and routers. The UDP transport protocol has a minimal set of features. This limited set has enabled a wide range of applications to use UDP, but these application do need to provide many important transport functions on top of UDP. The UDP Usage Guidelines [RFC5405] provides overall guidance for application designers, including the use of UDP to support tunneling. The key difference between UDP usage with IPv4 and IPv6 is that RFC 2460 mandates use of a calculated UDP checksum, i.e. a non-zero value, due to the lack of an IPv6 header checksum. The inclusion of the pseudo header in the checksum computation provides a statistical check that datagrams have been delivered to the intended IPv6 destination node. Algorithms for checksum computation are described in [RFC1071].

The lack of a possibility to use an IPv6 datagram with a zero UDP checksum has been observed as a real problem for certain classes of application, primarily tunnel applications. This class of application has been deployed with a zero UDP checksum using IPv4. The design of IPv6 raises different issues when considering the safety of using a UDP checksum with IPv6. These issues can significantly affect applications, both when an endpoint is the intended user and when an innocent bystander (when a packet is received by a different endpoint to that intended).

This document examines the issues and an appendix compares the strengths and weaknesses of a number of proposed solutions. This identifies a set of issues that must be considered and mitigated to be able to safely deploy IPv6 applications that use a zero UDP checksum. The provided comparison of methods is expected to also be useful when considering applications that have different goals from the ones that initiated the writing of this document, especially the use of already standardized methods. The analysis concludes that using a zero UDP checksum is the best method of the proposed alternatives to meet the goals for certain tunnel applications.

This document defines recommendations and requirements for use of IPv6 datagrams with a zero UDP checksum. This usage is expected to have initial deployment issues related to middleboxes, limiting the usability more than desired in the currently deployed Internet. However, this limitation will be largest initially and will reduce as updates are provided in middleboxes that support the zero UDP checksum for IPv6. The document therefore derives a set of constraints required to ensure safe deployment of a zero UDP checksum.

Finally, the document also identifies some issues that require future consideration and possibly additional research.

#### 1.1. Document Structure

Section 1 provides a background to key issues, and introduces the use of UDP as a tunnel transport protocol.

Section 2 describes a set of standards-track datagram transport protocols that may be used to support tunnels.

Section 3 discusses issues with a zero UDP checksum for IPv6. It considers the impact of corruption, the need for validation of the path and when it is suitable to use a zero UDP checksum.

Section 4 is an applicability statement that defines requirements and recommendations on the implementation of IPv6 nodes that support the use of a zero UDP checksum.

Section 5 provides an applicability statement that defines requirements and recommendations for protocols and tunnel encapsulations that are transported over an IPv6 transport that does not perform a UDP checksum calculation to verify the integrity at the transport endpoints.

Section 6 provides the recommendations for standardization of zero UDP checksum with a summary of the findings and notes remaining issues needing future work.

Appendix A evaluates the set of proposals to update the UDP transport behaviour and other alternatives intended to improve support for tunnel protocols. It concludes by assessing the trade-offs of the various methods, identifying advantages and disadvantages for each method.

#### 1.2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

#### 1.3. Use of UDP Tunnels

One increasingly popular use of UDP is as a tunneling protocol, where a tunnel endpoint encapsulates the packets of another protocol inside UDP datagrams and transmits them to another tunnel endpoint. Using UDP as a tunneling protocol is attractive when the payload protocol is not supported by the middleboxes that may exist along the path,

because many middleboxes support transmission using UDP. In this use, the receiving endpoint decapsulates the UDP datagrams and forwards the original packets contained in the payload [RFC5405]. Tunnels establish virtual links that appear to directly connect locations that are distant in the physical Internet topology and can be used to create virtual (private) networks.

#### 1.3.1. Motivation for new approaches

A number of tunnel encapsulations deployed over IPv4 have used the UDP transport with a zero checksum. Users of these protocols expect a similar solution for IPv6.

A number of tunnel protocols are also currently being defined (e.g. Automated Multicast Tunnels, AMT [I-D.ietf-mboned-auto-multicast], and the Locator/Identifier Separation Protocol, LISP [LISP]). These protocols motivated an update to IPv6 UDP checksum processing to benefit from simpler checksum processing for various reasons:

- o Reducing forwarding costs, motivated by redundancy present in the encapsulated packet header, since in tunnel encapsulations, payload integrity and length verification may be provided by higher layer encapsulations (often using the IPv4, UDP, UDP-Lite, or TCP checksums).
- o Eliminating a need to access the entire packet when forwarding the packet by a tunnel endpoint.
- o Enhancing ability to traverse and function with middleboxes.
- o A desire to use the port number space to enable load-sharing.

#### 1.3.2. Reducing forwarding cost

It is a common requirement to terminate a large number of tunnels on a single router/host. The processing cost per tunnel includes both state (memory requirements) and per-packet processing at the tunnel ingress and egress.

Automatic IP Multicast Tunneling, known as AMT [I-D.ietf-mboned-auto-multicast] currently specifies UDP as the transport protocol for packets carrying tunneled IP multicast packets. The current specification for AMT states that the UDP checksum in the outer packet header should be zero (see Section 6.6 of [I-D.ietf-mboned-auto-multicast]). This argues that the computation of an additional checksum is an unwarranted burden on nodes implementing lightweight tunneling protocols when an inner packet is already adequately protected, . The AMT protocol needs to



replicate a multicast packet to each gateway tunnel. In this case, the outer IP addresses are different for each tunnel and therefore require a different pseudo header to be built for each UDP replicated encapsulation.

The argument concerning redundant processing costs is valid regarding the integrity of a tunneled packet. In some architectures (e.g. PC-based routers), other mechanisms may also significantly reduce checksum processing costs: There are implementations that have optimised checksum processing algorithms, including the use of checksum-offloading. This processing is readily available for IPv4 packets at high line rates. Such processing may be anticipated for IPv6 endpoints, allowing receivers to reject corrupted packets without further processing. However, there are certain classes of tunnel end-points where this off-loading is not available and unlikely to become available in the near future.

#### 1.3.3. Need to inspect the entire packet

The currently-deployed hardware in many routers uses a fast-path processing that only provides the first  $n$  bytes of a packet to the forwarding engine, where typically  $n \leq 128$ .

When this design is used to support a tunnel ingress and egress, it prevents fast processing of a transport checksum over an entire (large) packet. Hence the currently defined IPv6 UDP checksum is poorly suited to use within a router that is unable to access the entire packet and does not provide checksum-offloading. Thus enabling checksum calculation over the complete packet can impact router design, performance improvement, energy consumption and/or cost.

#### 1.3.4. Interactions with middleboxes

Many paths in the Internet include one or more middleboxes of various types. There exist large classes of middleboxes that will handle zero UDP checksum packets, which would not support UDP-Lite or the other investigated proposals. These middleboxes includes load balancers (see Section 1.3.5) including Equal Cost Multipath Routing, traffic classifiers and other functions that reads some fields in the UDP headers but does not validate the UDP checksum.

There are also middleboxes that either validates or modify the UDP checksum. The two most common classes are Firewalls and NATs. In IPv4, UDP-encapsulation may be desirable for NAT traversal, since UDP support is commonly provided. It is also necessary due to the almost ubiquitous deployment of IPv4 NATs. There has also been discussion of NAT for IPv6, although not for the same reason as in IPv4. If

IPv6 NAT becomes a reality they hopefully do not present the same protocol issues as for IPv4. If NAT is defined for IPv6, it should take into consideration the use of a zero UDP checksum.

The requirements for IPv6 firewall traversal are likely to be similar to those for IPv4. In addition, it can be reasonably expected that a firewall conforming to RFC 2460 will not regard datagrams with a zero UDP checksum as valid. Use of a zero UDP checksum with IPv6 requires firewalls to be updated before the full utility of the change is available.

It can be expected that datagrams with zero UDP checksum will initially not have the same middlebox traversal characteristics as regular UDP (RFC 2460). However when implementations follow the requirements specified in this document, we expect the traversal capabilities to improve over time. We also note that deployment of IPv6-capable middleboxes is still in its initial phases. Thus, it might be that the number of non-updated boxes quickly become a very small percentage of the deployed middleboxes.

#### 1.3.5. Support for load balancing

The UDP port number fields have been used as a basis to design load-balancing solutions for IPv4. This approach has also been leveraged for IPv6. An alternate method would be to utilise the IPv6 Flow Label [RFC6437] as a basis for entropy for load balancing. This would have the desirable effect of releasing IPv6 load-balancing devices from the need to assume semantics for the use of the transport port field and also works for all type of transport protocols.

This use of the flow-label for load balancing is consistent with the intended use, although further clarity was needed to ensure the field can be consistently used for this purpose, therefore an updated IPv6 Flow Label [RFC6437] and Equal-Cost Multi-Path routing usage, (ECMP) [RFC6438] was produced. Router vendors could be encouraged to start using the IPv6 Flow Label as a part of the flow hash, providing support for ECMP without requiring use of UDP.

However, the method for populating the outer IPv6 header with a value for the flow label is not trivial: If the inner packet uses IPv6, then the flow label value could be copied to the outer packet header. However, many current end-points set the flow label to a zero value (thus no entropy). The ingress of a tunnel seeking to provide good entropy in the flow label field would therefore need to create a random flow label value and keep corresponding state, so that all packets that were associated with a flow would be consistently given the same flow label. Although possible, this complexity may not be

desirable in a tunnel ingress.

The end-to-end use of flow labels for load balancing is a long-term solution. Even if the usage of the flow label is clarified, there would be a transition time before a significant proportion of end-points start to assign a good quality flow label to the flows that they originate, with continued use of load balancing using the transport header fields until any widespread deployment is finally achieved.

## 2. Standards-Track Transports

The IETF has defined a set of transport protocols that may be applicable for tunnels with IPv6. There are also a set of network layer encapsulation tunnels such as IP-in-IP and GRE. These already standardized solutions are discussed here prior to the issues, as background for the issue description and some comparison of where the issue may already occur.

### 2.1. UDP with Standard Checksum

UDP [RFC0768] with standard checksum behaviour, as defined in RFC 2460, has already been discussed. UDP usage guidelines are provided in [RFC5405].

### 2.2. UDP-Lite

UDP-Lite [RFC3828] offers an alternate transport to UDP, specified as a proposed standard, RFC 3828. A MIB is defined in [RFC5097] and unicast usage guidelines in [RFC5405]. There is at least one open source implementation as a part of the Linux kernel since version 2.6.20.

UDP-Lite provides a checksum with optional partial coverage. When using this option, a datagram is divided into a sensitive part (covered by the checksum) and an insensitive part (not covered by the checksum). When the checksum covers the entire packet, UDP-Lite is fully equivalent with UDP, with the exception that it uses a different value in the Next Header field in the IPv6 header. Errors/corruption in the insensitive part will not cause the datagram to be discarded by the transport layer at the receiving endpoint. A minor side-effect of using UDP-Lite is that this was specified for damage-tolerant payloads and some link-layers may employ different link encapsulations when forwarding UDP-Lite segments (e.g. radio access bearers). Most link-layers will cover the insensitive part with the same strong layer 2 frame CRC that covers the sensitive part.

#### 2.2.1. Using UDP-Lite as a Tunnel Encapsulation

Tunnel encapsulations can use UDP-Lite (e.g. Control And Provisioning of Wireless Access Points, CAPWAP [RFC5415]), since UDP-Lite provides a transport-layer checksum, including an IP pseudo header checksum, in IPv6, without the need for a router/middlebox to traverse the entire packet payload. This provides most of the verification required for delivery and still keeps a low complexity for the checksumming operation. UDP-Lite may set the length of checksum coverage on a per packet basis. This feature could be used if a tunnel protocol is designed to only verify delivery of the tunneled payload and uses a calculated checksum for control information.

There is currently poor support for middlebox traversal using UDP-Lite, because UDP-Lite uses a different IPv6 network-layer Next Header value to that of UDP, and few middleboxes are able to interpret UDP-Lite and take appropriate actions when forwarding the packet. This makes UDP-Lite less suited to protocols needing general Internet support, until such time that UDP-Lite has achieved better support in middleboxes and end-points.

#### 2.3. General Tunnel Encapsulations

The IETF has defined a set of tunneling protocols or network layer encapsulations, e.g., IP-in-IP and GRE. These either do not include a checksum or use a checksum that is optional, since tunnel encapsulations are typically layered directly over the Internet layer (identified by the upper layer type in the IPv6 Next Header field) and are also not used as endpoint transport protocols. There is little chance of confusing a tunnel-encapsulated packet with other application data that could result in corruption of application state or data.

From the end-to-end perspective, the principal difference is that the network-layer Next Header field identifies a separate transport, which reduces the probability that corruption could result in the packet being delivered to the wrong endpoint or application. Specifically, packets are only delivered to protocol modules that process a specific Next Header value. The Next Header field therefore provides a first-level check of correct demultiplexing. In contrast, the UDP port space is shared by many diverse applications and therefore UDP demultiplexing relies solely on the port numbers.

#### 2.4. Relation to UDP-Lite and UDP with checksum

The operation of IPv6 with UDP with a zero-checksum is not the same as IPv4 with UDP with a zero-checksum. Protocol designers should not

be fooled into thinking the two are the same. The requirements below list a set of additional considerations.

Where possible, existing general tunnel encapsulations, such as GRE, IP-in-IP, should be used. This section assumes that such existing tunnel encapsulations do not offer the functionality required to satisfy the protocol designer's goals. The section considers the standardized alternative solutions, rather than the full set of ideas evaluated in Appendix A. The alternatives to UDP with a zero checksum are UDP with a (calculated) checksum, and UDP-Lite.

UDP with a checksum has the advantage of close to universal support in both endpoints and middleboxes. It also provides statistical verification of delivery to the intended destination (address and port). However, some classes of device have limited support for calculation of a checksum that covers a full datagram. For these devices, this can incur significant processing cost (e.g. requiring processing in the router slow-path) and can hence reduce capacity or fail to function.

UDP-Lite has the advantage of using a checksum that is calculated only over the pseudo header and the UDP header. This provides a statistical verification of delivery to the intended destination (address and port). The checksum can be calculated without access to the datagram payload, only requiring access to the part to be protected. A drawback is that UDP-Lite has currently limited support in both end-points (i.e. is not supported on all operating system platforms) and middleboxes (that require support for the UDP-Lite header type). A path verification method is therefore recommended.

IPv6 and UDP with a zero-checksum can also be used by nodes that do not permit calculation of a payload checksum. Many existing classes of middleboxes do not verify or change the transport checksum. For these middleboxes, IPv6 with a zero UDP checksum is expected to function where UDP-Lite would not. However, support for the zero UDP checksum in middleboxes that do change or verify the checksum is currently limited, and this may result in datagrams with a zero UDP checksum being discarded, therefore a path verification method is recommended.

There are sets of constraints for which no solution exist: A protocol designer that needs to originate or receive datagrams on a device that can not efficiently calculate a checksum over a full datagram and also needs these packets to pass through a middlebox that verifies or changes a UDP checksum, but does not support a zero UDP checksum, can not use the zero UDP checksum method. Similarly, one that originates datagrams on a device with UDP-Lite support, but needs the packets to pass through a middlebox that does not support

UDP-Lite, can not use UDP-Lite. For such cases, there is no optimal solution and the current recommendation is to use or fall-back to using UDP with full checksum coverage.

### 3. Issues Requiring Consideration

This informative section evaluates issues around the proposal to update IPv6 [RFC2460], to enable the UDP transport checksum to be set to zero. Some of the identified issues are shared with other protocols already in use. The section also provides background to the requirements and recommendations that follow.

The decision in RFC 2460 to omit an integrity check at the network level meant that the IPv6 transport checksum was overloaded with many functions, including validating:

- o the endpoint address was not corrupted within a router, i.e., a packet was intended to be received by this destination and validate that the packet does not consist of a wrong header spliced to a different payload;
- o that extension header processing is correctly delimited - i.e., the start of data has not been corrupted. In this case, reception of a valid Next Header value provides some protection;
- o reassembly processing, when used;
- o the length of the payload;
- o the port values - i.e., the correct application receives the payload (applications should also check the expected use of source ports/addresses);
- o the payload integrity.

In IPv4, the first four checks are performed using the IPv4 header checksum.

In IPv6, these checks occur within the endpoint stack using the UDP checksum information. An IPv6 node also relies on the header information to determine whether to send an ICMPv6 error message [RFC4443] and to determine the node to which this is sent. Corrupted information may lead to misdelivery to an unintended application socket on an unexpected host.

### 3.1. Effect of packet modification in the network

IP packets may be corrupted as they traverse an Internet path. Older evidence in "When the CRC and TCP Checksum Disagree" [Sigcomm2000] show that this was once an issue in year 2000 with IPv4 routers, and occasional corruption could result from bad internal router processing in routers or hosts. These errors are not detected by the strong frame checksums employed at the link-layer [RFC3819]. During the development of this document in 2009, individuals provided reports of observed rates for received UDP datagrams using IPv4 where the UDP checksum had been detected as corrupt. These rates were as high as 1.39E-4 for some paths, but also close to zero for some other paths.

There is extensive experience of deployment using tunnel protocols in well-managed networks (e.g. corporate networks or service provider core networks). This has shown the robustness of methods such as PWE and MPLS that do not employ a transport protocol checksum and have not specified mechanisms to protect from corruption of the unprotected headers (such as the VPN Identifier in MPLS). Reasons for the robustness may include:

- o A reduced probability of corruption on paths through well-managed networks.
- o IP forms the majority of the inner traffic carried by these tunnels. Hence from a transport perspective, endpoint verification is already being performed when processing a received IPv4 packet or by the transport pseudo-header for an IPv6 packet. This update to UDP does not change this behaviour.
- o In certain cases, a combination of additional filtering (e.g. filter of a MAC destination address in a L2 tunnel) significantly reduces the probability of final mis-delivery to the IP stack.
- o The tunnel protocols did not use a UDP transport header, any corruption is therefore unlikely to result in misdelivery to another UDP-based application. This concern is specific to the use of UDP with IPv6.

While this experience can guide the present recommendations, any update to UDP must preserve operation in the general Internet. This is heterogeneous and can include links and systems of very varying characteristics. Transport protocols used by hosts need to be designed with this in mind, especially when there is need to traverse edge networks, where middlebox deployments are common.

For the general Internet, there is no current evidence that

corruption is rare, nor that this may not be applicable to IPv6. It therefore seems prudent not to relax checks on misdelivery. The emergence of low-end IPv6 routers and the proposed use of NAT with IPv6 further motivate the need to protect from misdelivery.

Corruption in the network may result in:

- o A datagram being misdelivered to the wrong host/router or the wrong transport entity within an endpoint. Such a datagram needs to be discarded;
- o A datagram payload being corrupted, but still delivered to the intended host/router transport entity. Such a datagram needs to be either discarded or correctly processed by an application that provides its own integrity checks;
- o A datagram payload being truncated by corruption of the length field. Such a datagram needs to be discarded.

When a checksum is used, this significantly reduces the impact of errors, reducing the probability of undetected corruption of state (and data) on both the host stack and the applications using the transport service.

The following sections examine the impact of modifying each of these header fields.

#### 3.1.1. Corruption of the destination IP address

An IPv6 endpoint destination address could be modified in the network (e.g. corrupted by an error). This is not a concern for IPv4, because the IP header checksum will result in this packet being discarded by the receiving IP stack. Such modification in the network can not be detected at the network layer when using IPv6. Detection of this corruption by a UDP receiver relies on the IPv6 pseudo header incorporated in the transport checksum.

There are two possible outcomes:

- o Delivery to a destination address that is not in use (the packet will not be delivered, but could result in an error report);
- o Delivery to a different destination address. This modification will normally be detected by the transport checksum, resulting in silent discard. Without a computed checksum, the packet would be passed to the endpoint port demultiplexing function. If an application is bound to the associated ports, the packet payload will be passed to the application (see the subsequent section on



port processing).

### 3.1.2. Corruption of the source IP address

This section examines what happens when the source address is corrupted in transit. This is not a concern in IPv4, because the IP header checksum will normally result in this packet being discarded by the receiving IP stack. Detection of this corruption by a UDP receiver relies on the IPv6 pseudo header incorporated in the transport checksum.

Corruption of an IPv6 source address does not result in the IP packet being delivered to a different endpoint protocol or destination address. If only the source address is corrupted, the datagram will likely be processed in the intended context, although with erroneous origin information. When using Unicast Reverse Path Forwarding [RFC2827], a change in address may result in the router discarding the packet when the route to the modified source address is different to that of the source address of the original packet.

The result will depend on the application or protocol that processes the packet. Some examples are:

- o An application that requires a per-established context may disregard the datagram as invalid, or could map this to another context (if a context for the modified source address was already activated).
- o A stateless application will process the datagram outside of any context, a simple example is the ECHO server, which will respond with a datagram directed to the modified source address. This would create unwanted additional processing load, and generate traffic to the modified endpoint address.
- o Some datagram applications build state using the information from packet headers. A previously unused source address would result in receiver processing and the creation of unnecessary transport-layer state at the receiver. For example, Real Time Protocol (RTP) [RFC3550] sessions commonly employ a source independent receiver port. State is created for each received flow. Reception of a datagram with a corrupted source address will therefore result in accumulation of unnecessary state in the RTP state machine, including collision detection and response (since the same synchronization source, SSRC, value will appear to arrive from multiple source IP addresses).
- o ICMP messages relating to a corrupted packet can be misdirected to the wrong source node.

In general, the effect of corrupting the source address will depend upon the protocol that processes the packet and its robustness to this error. For the case where the packet is received by a tunnel endpoint, the tunnel application is expected to correctly handle a corrupted source address.

The impact of source address modification is more difficult to quantify when the receiving application is not that originally intended and several fields have been modified in transit.

#### 3.1.3. Corruption of Port Information

This section describes what happens if one or both of the UDP port values are corrupted in transit. This can also happen with IPv4 is used with a zero UDP checksum, but not when UDP checksums are calculated or when UDP-Lite is used. If the ports carried in the transport header of an IPv6 packet were corrupted in transit, packets may be delivered to the wrong application process (on the intended machine) and/or responses or errors sent to the wrong application process (on the intended machine).

#### 3.1.4. Delivery to an unexpected port

If one combines the corruption effects, such as destination address and ports, there is a number of potential outcomes when traffic arrives at an unexpected port. This section discusses these possibilities and their outcomes for a packet that does not use the UDP checksum validation:

- o Delivery to a port that is not in use. The packet is discarded, but could generate an ICMPv6 message (e.g. port unreachable).
- o It could be delivered to a different node that implements the same application, where the packet may be accepted, generating side-effects or accumulated state.
- o It could be delivered to an application that does not implement the tunnel protocol, where the packet may be incorrectly parsed, and may be misinterpreted, generating side-effects or accumulated state.

The probability of each outcome depends on the statistical probability that the address or the port information for the source or destination becomes corrupt in the datagram such that they match those of an existing flow or server port. Unfortunately, such a match may be more likely for UDP than for connection-oriented transports, because:

1. There is no handshake prior to communication and no sequence numbers (as in TCP, DCCP, or SCTP). Together, this makes it hard to verify that an application process is given only the application data associated with a specific transport session.
2. Applications writers often bind to wild-card values in endpoint identifiers and do not always validate correctness of datagrams they receive (guidance on this topic is provided in [RFC5405]).

While these rules could, in principle, be revised to declare naive applications as "Historic". This remedy is not realistic: the transport owes it to the stack to do its best to reject bogus datagrams.

If checksum coverage is suppressed, the application therefore needs to provide a method to detect and discard the unwanted data. A tunnel protocol would need to perform its own integrity checks on any control information if transported in datagrams with a zero UDP checksum. If the tunnel payload is another IP packet, the packets requiring checksums can be assumed to have their own checksums provided that the rate of corrupted packets is not significantly larger due to the tunnel encapsulation. If a tunnel transports other inner payloads that do not use IP, the assumptions of corruption detection for that particular protocol must be fulfilled, this may require an additional checksum/CRC and/or integrity protection of the payload and tunnel headers.

A protocol that uses a zero UDP checksum can not assume that it is the only protocol using a zero UDP checksum. Therefore, it needs to gracefully handle misdelivery. It must be robust to reception of malformed packets received on a listening port and expect that these packets may contain corrupted data or data associated with a completely different protocol.

#### 3.1.5. Corruption of Fragmentation Information

The fragmentation information in IPv6 employs a 32-bit identity field, compared to only a 16-bit field in IPv4, a 13-bit fragment offset and a 1-bit flag, indicating if there are more fragments. Corruption of any of these field may result in one of two outcomes:

Reassembly failure: An error in the "More Fragments" field for the last fragment will for example result in the packet never being considered complete and will eventually be timed out and discarded. A corruption in the ID field will result in the fragment not being delivered to the intended context thus leaving the rest incomplete, unless that packet has been duplicated prior to corruption. The incomplete packet will eventually be timed out

and discarded.

Erroneous reassembly: The re-assembled packet did not match the original packet. This can occur when the ID field of a fragment is corrupted, resulting in a fragment becoming associated with another packet and taking the place of another fragment. Corruption in the offset information can cause the fragment to be misaligned in the reassembly buffer, resulting in incorrect reassembly. Corruption can cause the packet to become shorter or longer, however completion of reassembly is much less probable, since this would require consistent corruption of the IPv6 headers payload length field and the offset field. The possibility of mis-assembly requires the reassembling stack to provide strong checks that detect overlap or missing data, note however that this is not guaranteed and has been clarified in "Handling of Overlapping IPv6 Fragments" [RFC5722].

The erroneous reassembly of packets is a general concern and such packets should be discarded instead of being passed to higher layer processes. The primary detector of packet length changes is the IP payload length field, with a secondary check by the transport checksum. The Upper-Layer Packet length field included in the pseudo header assists in verifying correct reassembly, since the Internet checksum has a low probability of detecting insertion of data or overlap errors (due to misplacement of data). The checksum is also incapable of detecting insertion or removal of all zero-data that occurs in a multiple of a 16-bit chunk.

The most significant risk of corruption results following mis-association of a fragment with a different packet. This risk can be significant, since the size of fragments is often the same (e.g. fragments resulting when the path MTU results in fragmentation of a larger packet, common when addition of a tunnel encapsulation header expands the size of a packet). Detection of this type of error requires a checksum or other integrity check of the headers and the payload. Such protection is anyway desirable for tunnel encapsulations using IPv4, since the small fragmentation ID can easily result in wrap-around [RFC4963], this is especially the case for tunnels that perform flow aggregation [I-D.ietf-intarea-tunnels].

Tunnel fragmentation behavior matters. There can be outer or inner fragmentation "Tunnels in the Internet Architecture" [I-D.ietf-intarea-tunnels]. If there is inner fragmentation by the tunnel, the outer headers will never be fragmented and thus a zero UDP checksum in the outer header will not affect the reassembly process. When a tunnel performs outer header fragmentation, the tunnel egress needs to perform reassembly of the outer fragments into an inner packet. The inner packet is either a complete packet or a

fragment. If it is a fragment, the destination endpoint of the fragment will perform reassembly of the received fragments. The complete packet or the reassembled fragments will then be processed according to the packet Next Header field. The receiver may only detect reassembly anomalies when it uses a protocol with a checksum. The larger the number of reassembly processes to which a packet has been subjected, the greater the probability of an error.

- o An IP-in-IP tunnel that performs inner fragmentation has similar properties to a UDP tunnel with a zero UDP checksum that also performs inner fragmentation.
- o An IP-in-IP tunnel that performs outer fragmentation has similar properties to a UDP tunnel with a zero UDP checksum that performs outer fragmentation.
- o A tunnel that performs outer fragmentation can result in a higher level of corruption due to both inner and outer fragmentation, enabling more chances for reassembly errors to occur.
- o Recursive tunneling can result in fragmentation at more than one header level, even for inner fragmentation unless it goes to the inner-most IP header.
- o Unless there is verification at each reassembly, the probability for undetected error will increase with the number of times fragmentation is recursively applied, making IP-in-IP and UDP with zero UDP checksum both vulnerable to undetected errors.

In conclusion, fragmentation of datagrams with a zero UDP checksum does not worsen the performance compared to some other commonly used tunnel encapsulations. However, caution is needed for recursive tunneling without any additional verification at the different tunnel layers.

### 3.2. Where Packet Corruption Occurs

Corruption of IP packets can occur at any point along a network path, during packet generation, during transmission over the link, in the process of routing and switching, etc. Some transmission steps include a checksum or Cyclic Redundancy Check (CRC) that reduces the probability for corrupted packets being forwarded, but there still exists a probability that errors may propagate undetected.

Unfortunately the community lacks reliable information to identify the most common functions or equipment that result in packet corruption. However, there are indications that the place where corruption occurs can vary significantly from one path to another.

There is therefore a risk in applying evidence from one domain of usage to infer characteristics for another. Methods intended for general Internet usage must therefore assume that corruption can occur and deploy mechanisms to mitigate the effect of corruption and/or resulting misdelivery.

### 3.3. Validating the network path

IP transports designed for use in the general Internet should not assume specific path characteristics. Network protocols may reroute packets that change the set of routers and middleboxes along a path. Therefore transports such as TCP, SCTP and DCCP have been designed to negotiate protocol parameters, adapt to different network path characteristics, and receive feedback to verify that the current path is suited to the intended application. Applications using UDP and UDP-Lite need to provide their own mechanisms to confirm the validity of the current network path.

A zero value in the UDP checksum field is explicitly disallowed in RFC2460. Thus it may be expected that any device on the path that has a reason to look beyond the IP header, for example to validate the UDP checksum, will consider such a packet as erroneous or illegal and may discard it, unless the device is updated to support the new behavior. Any middlebox that modifies the UDP checksum, for example a NAT that changes the values of the IP and UDP header in such a way that the checksum over the pseudo header changes value, will need to be updated to support this behavior. Until then, a zero UDP checksum packet is likely to be discarded either directly in the middlebox or at the destination, when a zero UDP checksum has been modified to a non-zero by an incremental update.

A pair of end-points intending to use a new behavior will therefore not only need to ensure support at each end-point, but also that the path between them will deliver packets with the new behavior. This may require using negotiation or an explicit mandate to use the new behavior by all nodes that support the new protocol.

Enabling the use of a zero checksum places new requirements on equipment deployed within the network, such as middleboxes. A middlebox (e.g. Firewalls, Network Address Translators) may enable zero checksum usage for a particular range of ports. Note that checksum off-loading and operating system design may result in all IPv6 UDP traffic being sent with a calculated checksum. This requires middleboxes that are configured to enable a zero UDP checksum to continue to work with bidirectional UDP flows that use a zero UDP checksum in only one direction, and therefore they must not maintain separate state for a UDP flow based on its checksum usage.

Support along the path between end points can be guaranteed in limited deployments by appropriate configuration. In general, it can be expected to take time for deployment of any updated behaviour to become ubiquitous.

A sender will need to probe the path to verify the expected behavior. Path characteristics may change, and usage therefore should be robust and able to detect a failure of the path under normal usage and re-negotiate. Note that a bidirectional path does not necessarily support the same checksum usage in both the forward and return directions: Receipt of a datagram with a zero UDP checksum, does not imply that the remote endpoint can also receive a datagram with a zero UDP checksum. This will require periodic validation of the path, adding complexity to any solution using the new behavior.

#### 3.4. Applicability of method

The update to the IPv6 specification defined in [I-D.ietf-6man-udpchecksums] only modifies IPv6 nodes that implement specific protocols designed to permit omission of a UDP checksum. This document therefore provides an applicability statement for the updated method indicating when the mechanism can (and can not) be used. Enabling this, and ensuring correct interactions with the stack, implies much more than simply disabling the checksum algorithm for specific packets at the transport interface.

When the method is widely available, it may be expected to be used by applications that are perceived to gain benefit. Any solution that uses an end-to-end transport protocol, rather than an IP-in-IP encapsulation, needs to minimise the possibility that application processes could confuse a corrupted or wrongly delivered UDP datagram with that of data addressed to the application running on their endpoint.

The protocol or application that uses the zero checksum method must ensure that the lack of checksum does not affect the protocol operation. This includes being robust to receiving a unintended packet from another protocol or context following corruption of a destination or source address and/or port value. It also includes considering the need for additional implicit protection mechanisms required when using the payload of a UDP packet received with a zero checksum.

#### 3.5. Impact on non-supporting devices or applications

It is important to consider the potential impact of using a zero UDP checksum on end-point devices or applications that are not modified to support the new behavior or by default or preference, use the

regular behavior. These applications must not be significantly impacted by the update.

To illustrate why this necessary, consider the implications of a node that enables use of a zero UDP checksum at the interface level: This would result in all applications that listen to a UDP socket receiving datagrams where the checksum was not verified. This could have a significant impact on an application that was not designed with the additional robustness needed to handle received packets with corruption, creating state or destroying existing state in the application.

A zero UDP checksum therefore needs to be enabled only for individual ports using an explicit request by the application. In this case, applications using other ports would maintain the current IPv6 behavior, discarding incoming datagrams with a zero UDP checksum. These other applications would not be affected by this changed behavior. An application that allows the changed behavior should be aware of the risk of corruption and the increased level of misdirected traffic, and can be designed robustly to handle this risk.

#### 4. Constraints on implementation of IPv6 nodes supporting zero checksum

This section is an applicability statement that defines requirements and recommendations on the implementation of IPv6 nodes that support use of a zero value in the checksum field of a UDP datagram.

All implementations that support this zero UDP checksum method **MUST** conform to the requirements defined below.

1. An IPv6 sending node **MAY** use a calculated RFC 2460 checksum for all datagrams that it sends. This explicitly permits an interface that supports checksum offloading to insert an updated UDP checksum value in all UDP datagrams that it forwards, however note that sending a calculated checksum requires the receiver to also perform the checksum calculation. Checksum offloading can normally be switched off for a particular interface to ensure that datagrams are sent with a zero UDP checksum.
2. IPv6 nodes **SHOULD** by default **NOT** allow the zero UDP checksum method for transmission.
3. IPv6 nodes **MUST** provide a way for the application/protocol to indicate the set of ports that will be enabled to send datagrams with a zero UDP checksum. This may be implemented by enabling a



transport mode using a socket API call when the socket is established, or a similar mechanism. It may also be implemented by enabling the method for a pre-assigned static port used by a specific tunnel protocol.

4. IPv6 nodes MUST provide a method to allow an application/protocol to indicate that a particular UDP datagram is required to be sent with a UDP checksum. This needs to be allowed by the operating system at any time (e.g. to send keep-alive datagrams), not just when a socket is established in the zero checksum mode.
5. The default IPv6 node receiver behaviour MUST discard all IPv6 packets carrying datagrams with a zero UDP checksum.
6. IPv6 nodes MUST provide a way for the application/protocol to indicate the set of ports that will be enabled to receive datagrams with a zero UDP checksum. This may be implemented via a socket API call, or similar mechanism. It may also be implemented by enabling the method for a pre-assigned static port used by a specific tunnel protocol.
7. IPv6 nodes supporting usage of zero UDP checksums MUST also allow reception using a calculated UDP checksum on all ports configured to allow zero UDP checksum usage. (The sending endpoint, e.g. encapsulating ingress, may choose to compute the UDP checksum, or may calculate this by default.) The receiving endpoint MUST use the reception method specified in RFC2460 when the checksum field is not zero.
8. RFC 2460 specifies that IPv6 nodes SHOULD log received datagrams with a zero UDP checksum. This remains the case for any datagram received on a port that does not explicitly enable processing of a zero UDP checksum. A port for which the zero UDP checksum has been enabled MUST NOT log the datagram solely because the checksum value is zero.
9. IPv6 nodes MAY separately identify received UDP datagrams that are discarded with a zero UDP checksum. It SHOULD NOT add these to the standard log, since the endpoint has not been verified. This may be used to support other functions (such as a security policy).
10. IPv6 nodes that receive ICMPv6 messages that refer to packets with a zero UDP checksum MUST provide appropriate checks concerning the consistency of the reported packet to verify that the reported packet actually originated from the node, before acting upon the information (e.g. validating the address and

port numbers in the ICMPv6 message body).

#### 5. Requirements on usage of the zero UDP checksum

This section is an applicability statement that identifies requirements and recommendations for protocols and tunnel encapsulations that are transported over an IPv6 transport flow (e.g. tunnel) that does not perform a UDP checksum calculation to verify the integrity at the transport endpoints. Before deciding to use the zero UDP checksum and loose the integrity verification provided, a protocol developer should seriously consider if they can use checksummed UDP packets or UDP-Lite [RFC3828], because IPv6 with a zero UDP checksum is not equivalent in behavior to IPv4 with zero UDP checksum.

The requirements and recommendations for protocols and tunnel encapsulations using an IPv6 transport flow that does not perform a UDP checksum calculation to verify the integrity at the transport endpoints are:

1. Transported protocols that enable the use of zero UDP checksum MUST only enable this for a specific port or port-range. This needs to be enabled at the sending and receiving endpoints for a UDP flow.
2. An integrity mechanism is always RECOMMENDED at the transported protocol layer to ensure that corruption rates of the delivered payload is not increased (e.g. the inner-most packet of a UDP tunnel). A mechanism that isolates the causes of corruption (e.g. identifying misdelivery, IPv6 header corruption, tunnel header corruption) is expected to also provide additional information about the status of the tunnel (e.g. to suggest a security attack).
3. A transported protocol that encapsulates Internet Protocol (IPv4 or IPv6) packets MAY rely on the inner packet integrity checks, provided that the tunnel protocol will not significantly increase the rate of corruption of the inner IP packet. If a significantly increased corruption rate can occur, then the tunnel protocol MUST provide an additional integrity verification mechanism. Early detection is desirable to avoid wasting unnecessary computation, transmission capacity or storage for packets that will subsequently be discarded.
4. A transported protocol that supports use of a zero UDP checksum, MUST be designed so that corruption of this information does not result in accumulated state for the protocol.

5.    A transported protocol with a non-tunnel payload or one that encapsulates non-IP packets **MUST** have a CRC or other mechanism for checking packet integrity, unless the non-IP packet is specifically designed for transmission over a lower layer that does not provide a packet integrity guarantee.
6.    A transported protocol with control feedback **SHOULD** be robust to changes in the network path, since the set of middleboxes on a path may vary during the life of an association. The UDP endpoints need to discover paths with middleboxes that drop packets with a zero UDP checksum. Therefore, transported protocols **SHOULD** send keep-alive messages with a zero UDP checksum. An endpoint that discovers an appreciable loss rate for keep-alive packets **MAY** terminate the UDP flow (e.g. tunnel). Section 3.1.3 of RFC 5405 describes requirements for congestion control when using a UDP-based transport.
7.    A protocol with control feedback that can fall-back to using UDP with a calculated RFC 2460 checksum is expected to be more robust to changes in the network path. Therefore, keep-alive messages **SHOULD** include both UDP datagrams with a checksum and datagrams with a zero UDP checksum. This will enable the remote endpoint to distinguish between a path failure and dropping of datagrams with a zero UDP checksum.
8.    A middlebox implementation **MUST** allow forwarding of an IPv6 UDP datagram with both a zero and standard UDP checksum using the same UDP port.
9.    A middlebox **MAY** configure a restricted set of specific port ranges that forward UDP datagrams with a zero UDP checksum. The middlebox **MAY** drop IPv6 datagrams with a zero UDP checksum that are outside a configured range.
10.   When a middlebox forwards an IPv6 UDP flow containing datagrams with both a zero and standard UDP checksum, the middlebox **MUST** NOT maintain separate state for flows depending on the value of their UDP checksum field. (This requirement is necessary to enable a sender that always calculates a checksum to communicate via a middlebox with a remote endpoint that uses a zero UDP checksum.)

Special considerations are required when designing a UDP tunnel protocol, where the tunnel ingress or egress may be a router that may not have access to the packet payload. When the node is acting as a host (i.e., sending or receiving a packet addressed to itself), the checksum processing is similar to other hosts. However, when the node (e.g. a router) is acting as a tunnel ingress or egress that

forwards a packet to or from a UDP tunnel, there may be restricted access to the packet payload. This prevents calculating (or verifying) a UDP checksum. In this case, the tunnel protocol may use a zero UDP checksum and must:

- o Ensure that tunnel ingress and tunnel egress router are both configured to use a zero UDP checksum. For example, this may include ensuring that hardware checksum offloading is disabled.
- o The tunnel operator must ensure that middleboxes on the network path are updated to support use of a zero UDP checksum.
- o A tunnel egress should implement appropriate security techniques to protect from overload, including source address filtering to prevent traffic injection by an attacker, and rate-limiting of any packets that incur additional processing, such as UDP datagrams used for control functions that require verification of a calculated checksum to verify the network path. Usage of common control traffic for multiple tunnels between a pair of nodes can assist in reducing the number of packets to be processed.

## 6. Summary

This document provides an applicability statement for the use of UDP transport checksums with IPv6.

It examines the role of the UDP transport checksum when used with IPv6 and presents a summary of the trade-offs in evaluating the safety of updating RFC 2460 to permit an IPv6 endpoint to use a zero UDP checksum field to indicate that no checksum is present.

Application designers should first examine whether their transport goals may be met using standard UDP (with a calculated checksum) or by using UDP-Lite. The use of UDP with a zero UDP checksum has merits for some applications, such as tunnel encapsulation, and is widely used in IPv4. However, there are different dangers for IPv6: There is an increased risk of corruption and misdelivery when using zero UDP checksum in IPv6 compared to using IPv4 due to the lack of an IPv6 header checksum. Thus, applications need to evaluate the risks of enabling use of a zero UDP checksum and consider a solution that at least provides the same delivery protection as for IPv4, for example by utilizing UDP-Lite, or by enabling the UDP checksum. The use of checksum off-loading may help alleviate the cost of checksum processing and permit use of a checksum using method defined in RFC 2460.

Tunnel applications using UDP for encapsulation can in many cases use

a zero UDP checksum without significant impact on the corruption rate. A well-designed tunnel application should include consistency checks to validate the header information encapsulated with a received packet. In most cases, tunnels encapsulating IP packets can rely on the integrity protection provided by the transported protocol (or tunneled inner packet). When correctly implemented, such an endpoint will not be negatively impacted by omission of the transport-layer checksum. Recursive tunneling and fragmentation is a potential issue that can raise corruption rates significantly, and requires careful consideration.

Other UDP applications at the intended destination node or another node can be impacted if they are allowed to receive datagrams that have a zero UDP checksum. It is important that already deployed applications are not impacted by a change at the transport layer. If these applications execute on nodes that implement RFC 2460, they will discard (and log) all datagrams with a zero UDP checksum. This is not an issue.

In general, UDP-based applications need to employ a mechanism that allows a large percentage of the corrupted packets to be removed before they reach an application, both to protect the data stream of the application and the control plane of higher layer protocols. These checks are currently performed by the UDP checksum for IPv6, or the reduced checksum for UDP-Lite when used with IPv6.

The transport of recursive tunneling and the use of fragmentation pose difficult issues that need to be considered in the design of tunnel protocols. There is an increased risk of an error in the inner-most packet when fragmentation when several layers of tunneling and several different reassembly processes are run without verification of correctness. This requires extra thought and careful consideration in the design of transported tunnels.

Any use of the updated method must consider the implications on firewalls, NATs and other middleboxes. It is not expected that IPv6 NATs handle IPv6 UDP datagrams in the same way that they handle IPv4 UDP datagrams. In many deployed cases this will require an update to support an IPv6 zero UDP checksum. Firewalls are intended to be configured, and therefore may need to be explicitly updated to allow new services or protocols. IPv6 middlebox deployment is not yet as prolific as it is in IPv4, and therefore new devices are expected to follow the methods specified in this document.

Each application should consider the implications of choosing an IPv6 transport that uses a zero UDP checksum, and consider whether other standard methods may be more appropriate, and may simplify application design.

## 7. Acknowledgements

Brian Haberman, Brian Carpenter, Margaret Wasserman, Lars Eggert, others in the TSV directorate. Barry Leiba, Ronald Bonica, Pete Resnick, and Stewart Bryant are thanked for resulting in a document with much greater applicability. Thanks to P.F. Chimento for careful review and editorial corrections.

Thanks also to: Remi Denis-Courmont, Pekka Savola, Glen Turner, and many others who contributed comments and ideas via the 6man, behave, lisp and mboned lists.

## 8. IANA Considerations

This document does not require any actions by IANA.

## 9. Security Considerations

Transport checksums provide the first stage of protection for the stack, although they can not be considered authentication mechanisms. These checks are also desirable to ensure packet counters correctly log actual activity, and can be used to detect unusual behaviours.

Depending on the hardware design, the processing requirements may differ for tunnels that have a zero UDP checksum and those that calculate a checksum. This processing overhead may need to be considered when deciding whether to enable a tunnel and to determine an acceptable rate for transmission. This can become a security risk for designs that can handle a significantly larger number of packets with zero UDP checksums compared to datagrams with a non-zero checksum, such as tunnel egress. An attacker could attempt to inject non-zero checksummed UDP packets into a tunnel forwarding zero checksum UDP packets and cause overload in the processing of the non-zero checksums, e.g. if this happens in a routers slow path. Protection mechanisms should therefore be employed when this threat exists. Protection may include source address filtering to prevent an attacker injecting traffic, as well as throttling the amount of non-zero checksum traffic. The latter may impact the function of the tunnel protocol.

Transmission of IPv6 packets with a zero UDP checksum could reveal additional information to an on-path attacker to identify the operating system or configuration of a sending node. There is a need to probe the network path to determine whether the current path supports using IPv6 packets with a zero UDP checksum. The details of the probing mechanism may differ for different tunnel encapsulations

and if visible in the network (e.g. if not using IPsec in encryption mode) could reveal additional information to an on-path attacker to identify the type of tunnel being used.

IP-in-IP or GRE tunnels offer good traversal of middleboxes that have not been designed for security, e.g. firewalls. However, firewalls may be expected to be configured to block general tunnels as they present a large attack surface. This applicability statement therefore permits this method to be enabled only for specific ranges of ports.

When the zero UDP checksum mode is enabled for a range of ports, nodes and middleboxes must forward received UDP datagrams that have either a calculated checksum or a zero checksum.

## 10. References

### 10.1. Normative References

- [I-D.ietf-6man-udpchecksums]  
Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", draft-ietf-6man-udpchecksums-08 (work in progress), February 2013.
- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

### 10.2. Informative References

- [I-D.ietf-intarea-tunnels]  
Touch, J. and M. Townsley, "Tunnels in the Internet Architecture", draft-ietf-intarea-tunnels-00 (work in progress), March 2010.
- [I-D.ietf-mboned-auto-multicast]  
Bumgardner, G., "Automatic Multicast Tunneling", draft-ietf-mboned-auto-multicast-14 (work in progress),

June 2012.

- [LISP]        D. Farinacci et al, "Locator/ID Separation Protocol (LISP)", November 2012.
- [RFC1071]    Braden, R., Borman, D., Partridge, C., and W. Plummer, "Computing the Internet checksum", RFC 1071, September 1988.
- [RFC1141]    Mallory, T. and A. Kullberg, "Incremental updating of the Internet checksum", RFC 1141, January 1990.
- [RFC1624]    Rijssinghani, A., "Computation of the Internet Checksum via Incremental Update", RFC 1624, May 1994.
- [RFC2827]    Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC3550]    Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3819]    Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.
- [RFC3828]    Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.
- [RFC4443]    Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC4963]    Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly Errors at High Data Rates", RFC 4963, July 2007.
- [RFC5097]    Renker, G. and G. Fairhurst, "MIB for the UDP-Lite protocol", RFC 5097, January 2008.
- [RFC5405]    Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.
- [RFC5415]    Calhoun, P., Montemurro, M., and D. Stanley, "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol



Specification", RFC 5415, March 2009.

[RFC5722]    Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, December 2009.

[RFC6437]    Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.

[RFC6438]    Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, November 2011.

[Sigcomm2000]    Jonathan Stone and Craig Partridge , "When the CRC and TCP Checksum Disagree", 2000.

[UDPTT]    G Fairhurst, "The UDP Tunnel Transport mode", Feb 2010.

#### Appendix A.    Evaluation of proposal to update RFC 2460 to support zero checksum

This informative appendix documents the evaluation of the proposal to update IPv6 [RFC2460], to provide the option that some nodes may suppress generation and checking of the UDP transport checksum. It also compares the proposal with other alternatives, and notes that for a particular application some standard methods may be more appropriate than using IPv6 with a zero UDP checksum.

##### A.1.    Alternatives to the Standard Checksum

There are several alternatives to the normal method for calculating the UDP Checksum [RFC1071] that do not require a tunnel endpoint to inspect the entire packet when computing a checksum. These include (in decreasing order of complexity):

- o    Delta computation of the checksum from an encapsulated checksum field. Since the checksum is a cumulative sum [RFC1624], an encapsulating header checksum can be derived from the new pseudo header, the inner checksum and the sum of the other network-layer fields not included in the pseudo header of the encapsulated packet, in a manner resembling incremental checksum update [RFC1141]. This would not require access to the whole packet, but does require fields to be collected across the header, and arithmetic operations on each packet. The method would only work for packets that contain a 2's complement transport checksum (i.e., it would not be appropriate for SCTP or when IP fragmentation is used).

- o UDP-Lite with the checksum coverage set to only the header portion of a packet. This requires a pseudo header checksum calculation only on the encapsulating packet header. The computed checksum value may be cached (before adding the Length field) for each flow/destination and subsequently combined with the Length of each packet to minimise per-packet processing. This value is combined with the UDP payload length for the pseudo header, however this length is expected to be known when performing packet forwarding.
- o The proposed UDP Tunnel Transport [UDPTT] suggested a method where UDP would be modified to derive the checksum only from the encapsulating packet protocol header. This value does not change between packets in a single flow. The value may be cached per flow/destination to minimise per-packet processing.
- o There has been a proposal to simply ignore the UDP checksum value on reception at the tunnel egress, allowing a tunnel ingress to insert any value correct or false. For tunnel usage, a non standard checksum value may be used, forcing an RFC 2460 receiver to drop the packet. The main downside is that it would be impossible to identify a UDP datagram (in the network or an endpoint) that is treated in this way compared to a packet that has actually been corrupted.
- o A method has been proposed that uses a new (to be defined) IPv6 Destination Options Header to provide an end-to-end validation check at the network layer. This would allow an endpoint to verify delivery to an appropriate end point, but would also require IPv6 nodes to correctly handle the additional header, and would require changes to middlebox behavior (e.g. when used with a NAT that always adjusts the checksum value).
- o UDP modified to disable checksum processing [I-D.ietf-6man-udpchecksums]. This eliminates the need for a checksum calculation, but would require constraints on appropriate usage and updates to end-points and middleboxes.
- o IP-in-IP tunneling. As this method completely dispenses with a transport protocol in the outer-layer it has reduced overhead and complexity, but also reduced functionality. There is no outer checksum over the packet and also no ports to perform demultiplexing between different tunnel types. This reduces the information available upon which a load balancer may act.

These options are compared and discussed further in the following sections.

## A.2. Comparison

This section compares the above listed methods to support datagram tunneling. It includes proposals for updating the behaviour of UDP.

While this comparison focuses on applications that are expected to execute on routers, the distinction between a router and a host is not always clear, especially at the transport level. Systems (such as unix-based operating systems) routinely provide both functions. There is no way to identify the role of the receiving node from a received packet.

### A.2.1. Middlebox Traversal

Regular UDP with a standard checksum or the delta encoded optimization for creating correct checksums have the best possibilities for successful traversal of a middlebox. No new support is required.

A method that ignores the UDP checksum on reception is expected to have a good probability of traversal, because most middleboxes perform an incremental checksum update. UDPTT would also have been able to traverse a middlebox with this behaviour. However, a middlebox on the path that attempts to verify a standard checksum will not forward packets using either of these methods, preventing traversal. A method that ignores the checksum has an additional downside in that it prevents improvement of middlebox traversal, because there is no way to identify UDP datagrams that use the modified checksum behaviour.

IP-in-IP or GRE tunnels offer good traversal of middleboxes that have not been designed for security, e.g. firewalls. However, firewalls may be expected to be configured to block general tunnels as they present a large attack surface.

A new IPv6 Destination Options header will suffer traversal issues with middleboxes, especially Firewalls and NATs, and will likely require them to be updated before the extension header is passed.

Datagrams with a zero UDP checksum will not be passed by any middlebox that validates the checksum using RFC 2460 or updates the checksum field, such as NAT or firewalls. This would require an update to correctly handle a datagram with a zero UDP checksum.

UDP-Lite will require an update of almost all type of middleboxes, because it requires support for a separate network-layer protocol number. Once enabled, the method to support incremental checksum update would be identical to that for UDP, but different for checksum

validation.

#### A.2.2. Load Balancing

The usefulness of solutions for load balancers depends on the difference in entropy in the headers for different flows that can be included in a hash function. All the proposals that use the UDP protocol number have equal behavior. UDP-Lite has the potential for equally good behavior as for UDP. However, UDP-Lite is currently unlikely to be supported by deployed hashing mechanisms, which could cause a load balancer to not use the transport header in the computed hash. A load balancer that only uses the IP header will have low entropy, but could be improved by including the IPv6 the flow label, providing that the tunnel ingress ensures that different flow labels are assigned to different flows. However, a transition to the common use of good quality flow labels is likely to take time to deploy.

#### A.2.3. Ingress and Egress Performance Implications

IP-in-IP tunnels are often considered efficient, because they introduce very little processing and low data overhead. The other proposals introduce a UDP-like header incurring associated data overhead. Processing is minimised for the method that uses a zero UDP checksum, ignoring the UDP checksum on reception, and only slightly higher for UDPTT, the extension header and UDP-Lite. The delta-calculation scheme operates on a few more fields, but also introduces serious failure modes that can result in a need to calculate a checksum over the complete datagram. Regular UDP is clearly the most costly to process, always requiring checksum calculation over the entire datagram.

It is important to note that the zero UDP checksum method, ignoring checksum on reception, the Option Header, UDPTT and UDP-Lite will likely incur additional complexities in the application to incorporate a negotiation and validation mechanism.

#### A.2.4. Deployability

The major factors influencing deployability of these solutions are a need to update both end-points, a need for negotiation and the need to update middleboxes. These are summarised below:

- o The solution with the best deployability is regular UDP. This requires no changes and has good middlebox traversal characteristics.
- o The next easiest to deploy is the delta checksum solution. This does not modify the protocol on the wire and only needs changes in

tunnel ingress.

- o IP-in-IP tunnels should not require changes to the end-points, but raise issues when traversing firewalls and other security devices, which are expected to require updates.
- o Ignoring the checksum on reception will require changes at both end-points. The never ceasing risk of path failure requires additional checks to ensure this solution is robust and will require changes or additions to the tunnel control protocol to negotiate support and validate the path.
- o The remaining solutions (including the zero checksum method) offer similar deployability. UDP-Lite requires support at both end-points and in middleboxes. UDPTT and the zero UDP checksum method with or without an extension header require support at both end-points and in middleboxes. UDP-Lite, UDPTT, and the zero UDP checksum method and use of extension headers may additionally require changes or additions to the tunnel control protocol to negotiate support and path validation.

#### A.2.5. Corruption Detection Strength

The standard UDP checksum and the delta checksum can both provide some verification at the tunnel egress. This can significantly reduce the probability that a corrupted inner packet is forwarded. UDP-Lite, UDPTT and the extension header all provide some verification against corruption, but do not verify the inner packet. They only provide a strong indication that the delivered packet was intended for the tunnel egress and was correctly delimited.

The methods using a zero UDP checksum, ignoring the UDP checksum on reception and IP-and-IP encapsulation all provide no verification that a received datagram was intended to be processed by a specific tunnel egress or that the inner encapsulated packet was correct. Section 3.1 discusses experience using specific protocols in well-managed networks.

#### A.2.6. Comparison Summary

The comparisons above may be summarised as "there is no silver bullet that will slay all the issues". One has to select which down side(s) can best be lived with. Focusing on the existing solutions, this can be summarized as:

Regular UDP:    The method defined in RFC 2460 has good middlebox traversal and load balancing and multiplexing, requiring a checksum in the outer headers covering the whole packet.

IP in IP:    A low complexity encapsulation, with limited middlebox traversal, no multiplexing support, and currently poor load balancing support that could improve over time.

UDP-Lite:    A medium complexity encapsulation, with good multiplexing support, limited middlebox traversal, but possible to improve over time, currently poor load balancing support that could improve over time, in most cases requiring application level negotiation to select the protocol and validation to confirm the path forwards UDP-Lite.

The delta-checksum is an optimization in the processing of UDP, as such it exhibits some of the drawbacks of using regular UDP.

The remaining proposals may be described in similar terms:

Zero-Checksum:    A low complexity encapsulation, with good multiplexing support, limited middlebox traversal that could improve over time, good load balancing support, in most cases requiring application level negotiation and validation to confirm the path forwards a zero UDP checksum.

UDPTT:    A medium complexity encapsulation, with good multiplexing support, limited middlebox traversal, but possible to improve over time, good load balancing support, in most cases requiring application level negotiation to select the transport and validation to confirm the path forwards UDPTT datagrams.

IPv6 Destination Option IP in IP tunneling:    A medium complexity, with no multiplexing support, limited middlebox traversal, currently poor load balancing support that could improve over time, in most cases requiring negotiation to confirm the option is supported and validation to confirm the path forwards the option.

IPv6 Destination Option combined with UDP Zero-checksumming:    A medium complexity encapsulation, with good multiplexing support, limited load balancing support that could improve over time, in most cases requiring negotiation to confirm the option is supported and validation to confirm the path forwards the option.

Ignore the checksum on reception:    A low complexity encapsulation, with good multiplexing support, medium middlebox traversal that never can improve, good load balancing support, in most cases requiring negotiation to confirm the option is supported by the

remote endpoint and validation to confirm the path forwards a zero UDP checksum.

There is no clear single optimum solution. If the most important need is to traverse middleboxes, then the best choice is to stay with regular UDP and consider the optimizations that may be required to perform the checksumming. If one can live with limited middlebox traversal, low complexity is necessary and one does not require load balancing, then IP-in-IP tunneling is the simplest. If one wants strengthened error detection, but with currently limited middlebox traversal and load-balancing. UDP-Lite is appropriate. Zero UDP checksum addresses another set of constraints, low complexity and a need for load balancing from the current Internet, providing it can live with currently limited middlebox traversal.

Techniques for load balancing and middlebox traversal do continue to evolve. Over a long time, developments in load balancing have good potential to improve. This time horizon is long since it requires both load balancer and end-point updates to get full benefit. The challenges of middlebox traversal are also expected to change with time, as device capabilities evolve. Middleboxes are very prolific with a larger proportion of end-user ownership, and therefore may be expected to take long time cycles to evolve.

One potential advantage is that the deployment of IPv6-capable middleboxes are still in its initial phase and the quicker a new method becomes standardized, the fewer boxes will be non-compliant.

Thus, the question of whether to permit use of datagrams with a zero UDP checksum for IPv6 under reasonable constraints, is therefore best viewed as a trade-off between a number of more subjective questions:

- o Is there sufficient interest in using a zero UDP checksum with the given constraints (summarised below)?
- o Are there other avenues of change that will resolve the issue in a better way and sufficiently quickly ?
- o Do we accept the complexity cost of having one more solution in the future?

The analysis concludes that the IETF should carefully consider constraints on sanctioning the use of any new transport mode. The 6man working group of the IETF has determined that the answer to the above questions are sufficient to update IPv6 to standardise use of a zero UDP checksum for use by tunnel encapsulations for specific applications.

Each application should consider the implications of choosing an IPv6 transport that uses a zero UDP checksum. In many cases, standard methods may be more appropriate, and may simplify application design. The use of checksum off-loading may help alleviate the checksum processing cost and permit use of a checksum using method defined in RFC 2460.

## Appendix B. Document Change History

{RFC EDITOR NOTE: This section must be deleted prior to publication}

Individual Draft 00    This is the first DRAFT of this document - It contains a compilation of various discussions and contributions from a variety of IETF WGs, including: mboned, tsv, 6man, lisp, and behave. This includes contributions from Magnus with text on RTP, and various updates.

### Individual Draft 01

- \* This version corrects some typos and editorial NiTs and adds discussion of the need to negotiate and verify operation of a new mechanism (3.3.4).

### Individual Draft 02

- \* Version -02 corrects some typos and editorial NiTs.
- \* Added reference to ECMP for tunnels.
- \* Clarifies the recommendations at the end of the document.

### Working Group Draft 00

- \* Working Group Version -00 corrects some typos and removes much of rationale for UDPTT. It also adds some discussion of IPv6 extension header.

### Working Group Draft 01

- \* Working Group Version -01 updates the rules and incorporates off-list feedback. This version is intended for wider review within the 6man working group.



Working Group Draft 02

- \* This version is the result of a major rewrite and re-ordering of the document.
- \* A new section comparing the results have been added.
- \* The constraints list has been significantly altered by removing some and rewording other constraints.
- \* This contains other significant language updates to clarify the intent of this draft.

Working Group Draft 03

- \* Editorial updates

Working Group Draft 04

- \* Resubmission only updating the AMT and RFC2765 references.

Working Group Draft 05

- \* Resubmission to correct editorial NiTs - thanks to Bill Atwood for noting these. Group Draft 05.

Working Group Draft 06

- \* Resubmission to keep draft alive (spelling updated from 05).

Working Group Draft 07

- \* Interim Version
- \* Submission after IESG Feedback Added
- \* Updates to enable the document to become a PS Applicability Statement

Working Group Draft 08

- \* First Version written as a PS Applicability Statement
- \* Changes to reflect decision to update RFC 2460, rather than recommend decision
- \* Updates to requirements for middleboxes

- \* Inclusion of requirements for security, API, and tunnel
- \* Move of the rationale for the update to an Annex (former section 4)

Working Group Draft 09

- \* Submission after second WGLC (note mistake corrected in -09).
- \* Clarified role of API for supporting full checksum.
- \* Clarified that full checksum is required in security considerations, and therefore noting that full checksum should not be treated as an attack - consistent with remainder of document.
- \* Added mention that API can set a mode in transport stack - to link to similar statement in RFC 2460 update.
- \* Fixed typos.

Working Group Draft 10

- \* Submission to correct unwanted removal of text from section 5 bullets 5-7 by GF.
- \* Replaced section 5 text with the text from 08, and reapplied the editorial correction.
- \* Note to reviewers: Please compare this revision with -08 used in the IETF LC).

Working Group Draft 11

- \* Added REF for 5097 (Noted by S.Turner)
- \* Added text in response to P. Resnick on place where checksum is calculated.
- \* Added text to note experience with MPLS/PWE; Appendix updated to refer to this (S. Bryant)
- \* Added text in response to P.Resnick's 2nd comments.
- \* Request to make UDP-Lite more clearly recommended (J Touch, P.Resnick)

- \* Added considerations around usage of zero checksum in routers.
- \* Added text in response to Stewart Bryant's comments on router requirements.

#### Authors' Addresses

Godred Fairhurst  
University of Aberdeen  
School of Engineering  
Aberdeen, AB24 3UE  
Scotland, UK

Email: [gorry@erg.abdn.ac.uk](mailto:gorry@erg.abdn.ac.uk)  
URI: <http://www.erg.abdn.ac.uk/users/gorry>

Magnus Westerlund  
Ericsson  
Farogatan 6  
Stockholm, SE-164 80  
Sweden

Phone: +46 8 719 0000  
Email: [magnus.westerlund@ericsson.com](mailto:magnus.westerlund@ericsson.com)



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 28 April 2022

R. R. Stewart  
Netflix, Inc.  
M. Tüxen  
I. Rüngeler  
Münster Univ. of Appl. Sciences  
25 October 2021

Stream Control Transmission Protocol (SCTP) Network Address Translation  
Support  
draft-ietf-tsvwg-natsupp-23

Abstract

The Stream Control Transmission Protocol (SCTP) provides a reliable communications channel between two end-hosts in many ways similar to the Transmission Control Protocol (TCP). With the widespread deployment of Network Address Translators (NAT), specialized code has been added to NAT functions for TCP that allows multiple hosts to reside behind a NAT function and yet share a single IPv4 address, even when two hosts (behind a NAT function) choose the same port numbers for their connection. This additional code is sometimes classified as Network Address and Port Translation (NAPT).

This document describes the protocol extensions needed for the SCTP endpoints and the mechanisms for NAT functions necessary to provide similar features of NAPT in the single point and multipoint traversal scenario.

Finally, a YANG module for SCTP NAT is defined.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 28 April 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Conventions . . . . .	5
3. Terminology . . . . .	5
4. Motivation and Overview . . . . .	6
4.1. SCTP NAT Traversal Scenarios . . . . .	6
4.1.1. Single Point Traversal . . . . .	7
4.1.2. Multipoint Traversal . . . . .	7
4.2. Limitations of Classical NAPT for SCTP . . . . .	8
4.3. The SCTP-Specific Variant of NAT . . . . .	8
5. Data Formats . . . . .	13
5.1. Modified Chunks . . . . .	13
5.1.1. Extended ABORT Chunk . . . . .	13
5.1.2. Extended ERROR Chunk . . . . .	14
5.2. New Error Causes . . . . .	14
5.2.1. VTag and Port Number Collision Error Cause . . . . .	14
5.2.2. Missing State Error Cause . . . . .	15
5.2.3. Port Number Collision Error Cause . . . . .	15
5.3. New Parameters . . . . .	16
5.3.1. Disable Restart Parameter . . . . .	16
5.3.2. VTags Parameter . . . . .	17
6. Procedures for SCTP Endpoints and NAT Functions . . . . .	18
6.1. Association Setup Considerations for Endpoints . . . . .	19
6.2. Handling of Internal Port Number and Verification Tag Collisions . . . . .	19
6.2.1. NAT Function Considerations . . . . .	19
6.2.2. Endpoint Considerations . . . . .	20
6.3. Handling of Internal Port Number Collisions . . . . .	20
6.3.1. NAT Function Considerations . . . . .	20
6.3.2. Endpoint Considerations . . . . .	21
6.4. Handling of Missing State . . . . .	21
6.4.1. NAT Function Considerations . . . . .	22
6.4.2. Endpoint Considerations . . . . .	22

6.5.	Handling of Fragmented SCTP Packets by NAT Functions . .	24
6.6.	Multi Point Traversal Considerations for Endpoints . . .	24
7.	SCTP NAT YANG Module . . . . .	24
7.1.	Tree Structure . . . . .	24
7.2.	YANG Module . . . . .	25
8.	Various Examples of NAT Traversals . . . . .	27
8.1.	Single-homed Client to Single-homed Server . . . . .	28
8.2.	Single-homed Client to Multi-homed Server . . . . .	30
8.3.	Multihomed Client and Server . . . . .	32
8.4.	NAT Function Loses Its State . . . . .	35
8.5.	Peer-to-Peer Communications . . . . .	37
9.	Socket API Considerations . . . . .	42
9.1.	Get or Set the NAT Friendliness (SCTP_NAT_FRIENDLY) . . .	43
10.	IANA Considerations . . . . .	43
10.1.	New Chunk Flags for Two Existing Chunk Types . . . . .	43
10.2.	Three New Error Causes . . . . .	45
10.3.	Two New Chunk Parameter Types . . . . .	46
10.4.	One New URI . . . . .	46
10.5.	One New YANG Module . . . . .	46
11.	Security Considerations . . . . .	46
12.	Normative References . . . . .	47
13.	Informative References . . . . .	48
	Acknowledgments . . . . .	51
	Authors' Addresses . . . . .	51

## 1. Introduction

Stream Control Transmission Protocol (SCTP) [RFC4960] provides a reliable communications channel between two end-hosts in many ways similar to TCP [RFC0793]. With the widespread deployment of Network Address Translators (NAT), specialized code has been added to NAT functions for TCP that allows multiple hosts to reside behind a NAT function using private-use addresses (see [RFC6890]) and yet share a single IPv4 address, even when two hosts (behind a NAT function) choose the same port numbers for their connection. This additional code is sometimes classified as Network Address and Port Translation (NAPT). Please note that this document focuses on the case where the NAT function maps a single or multiple internal addresses to a single external address and vice versa.

To date, specialized code for SCTP has not yet been added to most NAT functions so that only a translation of IP addresses is supported. The end result of this is that only one SCTP-capable host can successfully operate behind such a NAT function and this host can only be single-homed. The only alternative for supporting legacy NAT functions is to use UDP encapsulation as specified in [RFC6951].

The NAT function in the document refers to NAPT functions described in Section 2.2 of [RFC3022], NAT64 [RFC6146], or DS-Lite AFTR [RFC6333].

This document specifies procedures allowing a NAT function to support SCTP by providing similar features to those provided by a NAPT for TCP (see [RFC5382] and [RFC7857]), UDP (see [RFC4787] and [RFC7857]), and ICMP (see [RFC5508] and [RFC7857]). This document also specifies a set of data formats for SCTP packets and a set of SCTP endpoint procedures to support NAT traversal. An SCTP implementation supporting these procedures can assure that in both single-homed and multi-homed cases a NAT function will maintain the appropriate state without the NAT function needing to change port numbers.

It is possible and desirable to make these changes for a number of reasons:

- \* It is desirable for SCTP internal end-hosts on multiple platforms to be able to share a NAT function's external IP address in the same way that a TCP session can use a NAT function.
- \* If a NAT function does not need to change any data within an SCTP packet, it will reduce the processing burden of NAT'ing SCTP by not needing to execute the CRC32c checksum used by SCTP.
- \* Not having to touch the IP payload makes the processing of ICMP messages by NAT functions easier.

An SCTP-aware NAT function will need to follow these procedures for generating appropriate SCTP packet formats.

When considering SCTP-aware NAT it is possible to have multiple levels of support. At each level, the Internal Host, Remote Host, and NAT function does or does not support the procedures described in this document. The following table illustrates the results of the various combinations of support and if communications can occur between two endpoints.



Internal Host	NAT Function	Remote Host	Communication
Support	Support	Support	Yes
Support	Support	No Support	Limited
Support	No Support	Support	None
Support	No Support	No Support	None
No Support	Support	Support	Limited
No Support	Support	No Support	Limited
No Support	No Support	Support	None
No Support	No Support	No Support	None

Table 1: Communication possibilities

From the table it can be seen that no communication can occur when a NAT function does not support SCTP-aware NAT. This assumes that the NAT function does not handle SCTP packets at all and all SCTP packets sent from behind a NAT function are discarded by the NAT function. In some cases, where the NAT function supports SCTP-aware NAT, but one of the two hosts does not support the feature, communication can possibly occur in a limited way. For example, only one host can have a connection when a collision case occurs.

## 2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Terminology

This document uses the following terms, which are depicted in Figure 1. Familiarity with the terminology used in [RFC4960] and [RFC5061] is assumed.

Internal-Address (Int-Addr)

An internal address that is known to the internal host.

**Internal-Port (Int-Port)**

The port number that is in use by the host holding the Internal-Address.

**Internal-VTag (Int-VTag)**

The SCTP Verification Tag (VTag) (see Section 3.1 of [RFC4960]) that the internal host has chosen for an association. The VTag is a unique 32-bit tag that accompanies any incoming SCTP packet for this association to the Internal-Address.

**Remote-Address (Rem-Addr)**

The address that an internal host is attempting to contact.

**Remote-Port (Rem-Port)**

The port number used by the host holding the Remote-Address.

**Remote-VTag (Rem-VTag)**

The Verification Tag (VTag) (see Section 3.1 of [RFC4960]) that the host holding the Remote-Address has chosen for an association. The VTag is a unique 32-bit tag that accompanies any outgoing SCTP packet for this association to the Remote-Address.

**External-Address (Ext-Addr)**

An external address assigned to the NAT function, that it uses as a source address when sending packets towards a Remote-Address.

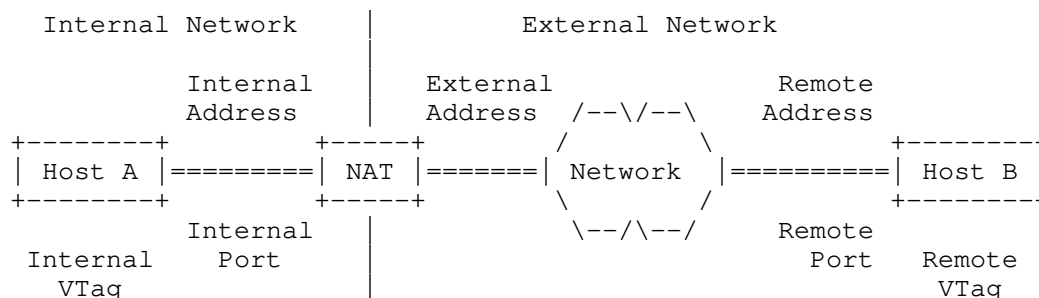


Figure 1: Basic Network Setup

## 4. Motivation and Overview

### 4.1. SCTP NAT Traversal Scenarios

This section defines the notion of single and multipoint NAT traversal.

#### 4.1.1. Single Point Traversal

In this case, all packets in the SCTP association go through a single NAT function, as shown in Figure 2.

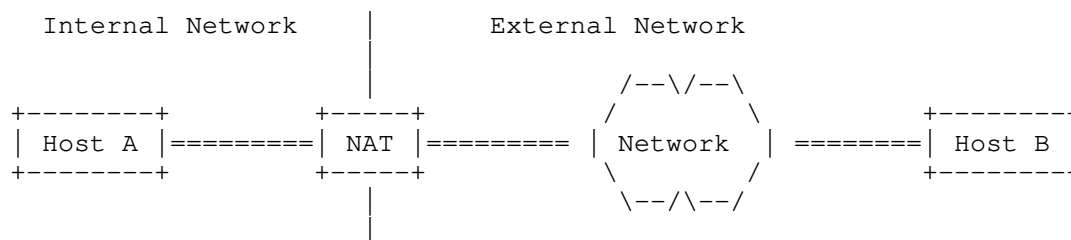


Figure 2: Single NAT Function Scenario

A variation of this case is shown in Figure 3, i.e., multiple NAT functions in the forwarding path between two endpoints.

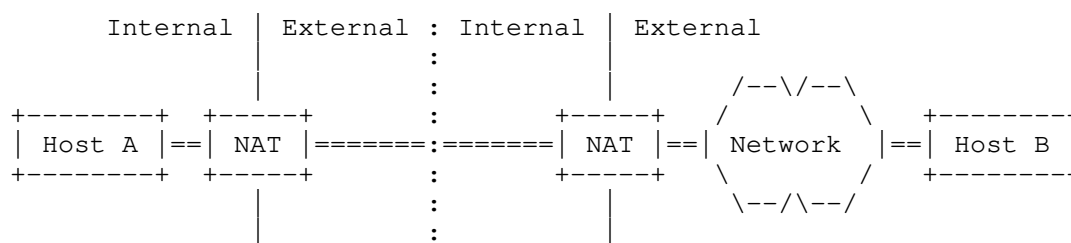


Figure 3: Serial NAT Functions Scenario

Although one of the main benefits of SCTP multi-homing is redundant paths, in the single point traversal scenario the NAT function represents a single point of failure in the path of the SCTP multi-homed association. However, the rest of the path can still benefit from path diversity provided by SCTP multi-homing.

The two SCTP endpoints in this case can be either single-homed or multi-homed. However, the important thing is that the NAT function in this case sees all the packets of the SCTP association.

#### 4.1.2. Multipoint Traversal

This case involves multiple NAT functions and each NAT function only sees some of the packets in the SCTP association. An example is shown in Figure 4.

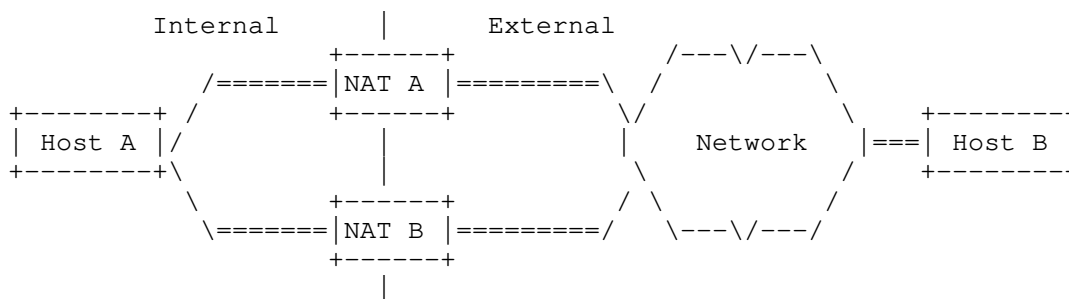


Figure 4: Parallel NAT Functions Scenario

This case does not apply to a single-homed SCTP association (i.e., both endpoints in the association use only one IP address). The advantage here is that the existence of multiple NAT traversal points can preserve the path diversity of a multi-homed association for the entire path. This in turn can improve the robustness of the communication.

#### 4.2. Limitations of Classical NAPT for SCTP

Using classical NAPT possibly results in changing one of the SCTP port numbers during the processing, which requires the recomputation of the transport layer checksum by the NAPT function. Whereas for UDP and TCP this can be done very efficiently, for SCTP the checksum (CRC32c) over the entire packet needs to be recomputed (see Appendix B of [RFC4960] for details of the CRC32c computation). This would considerably add to the NAT computational burden, however hardware support can mitigate this in some implementations.

An SCTP endpoint can have multiple addresses but only has a single port number to use. To make multipoint traversal work, all the NAT functions involved need to recognize the packets they see as belonging to the same SCTP association and perform port number translation in a consistent way. One possible way of doing this is to use a pre-defined table of port numbers and addresses configured within each NAT function. Other mechanisms could make use of NAT to NAT communication. Such mechanisms have not been deployed on a wide scale base and thus are not a preferred solution. Therefore an SCTP variant of NAT function has been developed (see Section 4.3).

#### 4.3. The SCTP-Specific Variant of NAT

In this section it is allowed that there are multiple SCTP capable hosts behind a NAT function that share one External-Address. Furthermore, this section focuses on the single point traversal scenario (see Section 4.1.1).

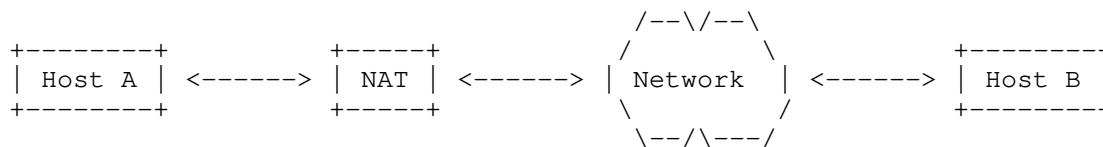
The modification of outgoing SCTP packets sent from an internal host is simple: the source address of the packets has to be replaced with the External-Address. It might also be necessary to establish some state in the NAT function to later handle incoming packets.

Typically, the NAT function has to maintain a NAT binding table of Internal-VTag, Internal-Port, Remote-VTag, Remote-Port, Internal-Address, and whether the restart procedure is disabled or not. An entry in that NAT binding table is called a NAT-State control block. The function Create() obtains the just mentioned parameters and returns a NAT-State control block. A NAT function MAY allow creating NAT-State control blocks via a management interface.

For SCTP packets coming from the external realm of the NAT function the destination address of the packets has to be replaced with the Internal-Address of the host to which the packet has to be delivered, if a NAT state entry is found. The lookup of the Internal-Address is based on the Remote-VTag, Remote-Port, Internal-VTag and the Internal-Port.

The entries in the NAT binding table need to fulfill some uniqueness conditions. There can not be more than one entry NAT binding table with the same pair of Internal-Port and Remote-Port. This rule can be relaxed, if all NAT binding table entries with the same Internal-Port and Remote-Port have the support for the restart procedure disabled (see Section 5.3.1). In this case there can not be no more than one entry with the same Internal-Port, Remote-Port and Remote-VTag and no more than one NAT binding table entry with the same Internal-Port, Remote-Port, and Int-VTag.

The processing of outgoing SCTP packets containing an INIT chunk is illustrated in the following figure. This scenario is valid for all message flows in this section.



```

INIT[Initiate-Tag]
Int-Addr:Int-Port -----> Rem-Addr:Rem-Port
Rem-VTag=0

Create(Initiate-Tag, Int-Port, 0, Rem-Port, Int-Addr,
      IsRestartDisabled)
Returns(NAT-State control block)

```

Translate To:

```

INIT[Initiate-Tag]
Ext-Addr:Int-Port -----> Rem-Addr:Rem-Port
Rem-VTag=0

```

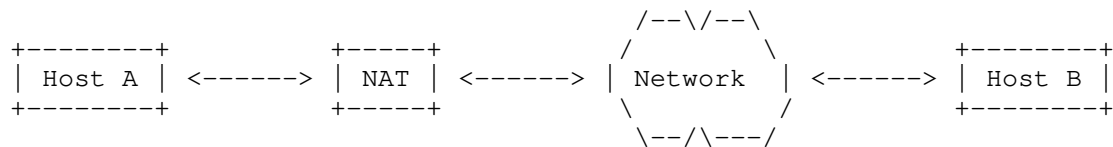
Normally a NAT binding table entry will be created.

However, it is possible that there is already a NAT binding table entry with the same Remote-Port, Internal-Port, and Internal-VTag but different Internal-Address and the restart procedure is disabled. In this case the packet containing the INIT chunk MUST be dropped by the NAT and a packet containing an ABORT chunk SHOULD be sent to the SCTP host that originated the packet with the M bit set and 'VTag and Port Number Collision' error cause (see Section 5.1.1 for the format). The source address of the packet containing the ABORT chunk MUST be the destination address of the packet containing the INIT chunk.

If an outgoing SCTP packet contains an INIT or ASCONF chunk and a matching NAT binding table entry is found, the packet is processed as a normal outgoing packet.

It is also possible that a NAT binding table entry with the same Remote-Port and Internal-Port exists without an Internal-VTag conflict but there exists a NAT binding table entry with the same port numbers but a different Internal-Address and the restart procedure is not disabled. In such a case the packet containing the INIT chunk MUST be dropped by the NAT function and a packet containing an ABORT chunk SHOULD be sent to the SCTP host that originated the packet with the M bit set and 'Port Number Collision' error cause (see Section 5.1.1 for the format).

The processing of outgoing SCTP packets containing no INIT chunks is described in the following figure.

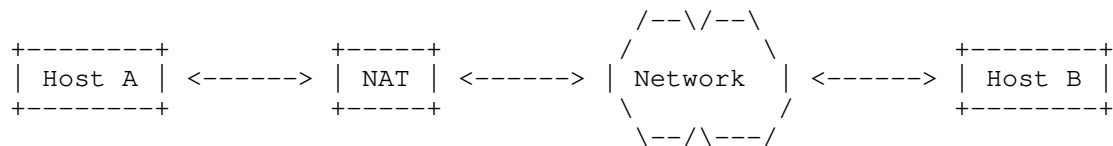


Int-Addr:Int-Port -----> Rem-Addr:Rem-Port  
                                   Rem-VTag

Translate To:

Ext-Addr:Int-Port -----> Rem-Addr:Rem-Port  
                                   Rem-VTag

The processing of incoming SCTP packets containing an INIT ACK chunk is illustrated in the following figure. The Lookup() function has as input the Internal-VTag, Internal-Port, Remote-VTag, and Remote-Port. It returns the corresponding entry of the NAT binding table and updates the Remote-VTag by substituting it with the value of the Initiate-Tag of the INIT ACK chunk. The wildcard character signifies that the parameter's value is not considered in the Lookup() function or changed in the Update() function, respectively.



INIT ACK[Initiate-Tag]  
 Ext-Addr:Int-Port <---- Rem-Addr:Rem-Port  
                                   Int-VTag

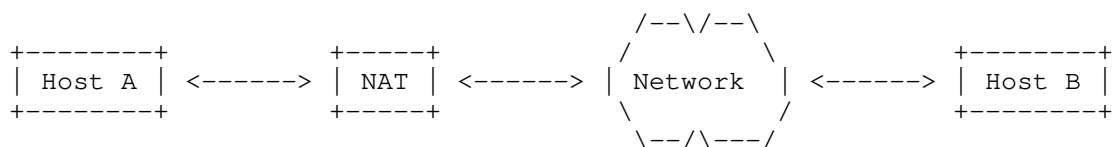
Lookup(Int-VTag, Int-Port, \*, Rem-Port)  
 Update(\*, \*, Initiate-Tag, \*)

Returns(NAT-State control block containing Int-Addr)

INIT ACK[Initiate-Tag]  
 Int-Addr:Int-Port <----- Rem-Addr:Rem-Port  
                                   Int-VTag

In the case where the Lookup function fails because it does not find an entry, the SCTP packet is dropped. If it succeeds, the Update routine inserts the Remote-VTag (the Initiate-Tag of the INIT ACK chunk) in the NAT-State control block.

The processing of incoming SCTP packets containing an ABORT or SHUTDOWN COMPLETE chunk with the T bit set is illustrated in the following figure.



Ext-Addr:Int-Port <----- Rem-Addr:Rem-Port  
Rem-VTag

Lookup(\*, Int-Port, Rem-VTag, Rem-Port)

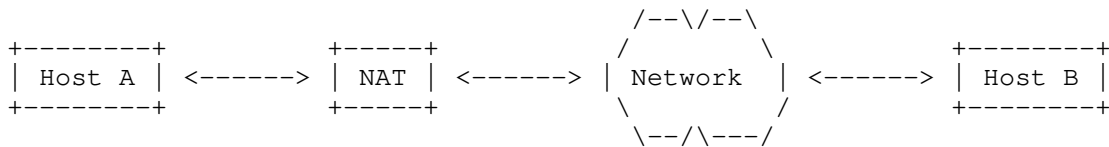
Returns (NAT-State control block containing Int-Addr)

Int-Addr:Int-Port <----- Rem-Addr:Rem-Port  
Rem-VTag

For an incoming packet containing an INIT chunk a table lookup is made only based on the addresses and port numbers. If an entry with a Remote-VTag of zero is found, it is considered a match and the Remote-VTag is updated. If an entry with a non-matching Remote-VTag is found or no entry is found, the incoming packet is silently dropped. If an entry with a matching Remote-VTag is found, the incoming packet is forwarded. This allows the handling of INIT collision through NAT functions.

The processing of other incoming SCTP packets is described in the following figure.





Ext-Addr:Int-Port <----- Rem-Addr:Rem-Port  
Int-VTag

Lookup(Int-VTag, Int-Port, \*, Rem-Port)

Returns(NAT-State control block containing Internal-Address)

Int-Addr:Int-Port <----- Rem-Addr:Rem-Port  
Int-VTag

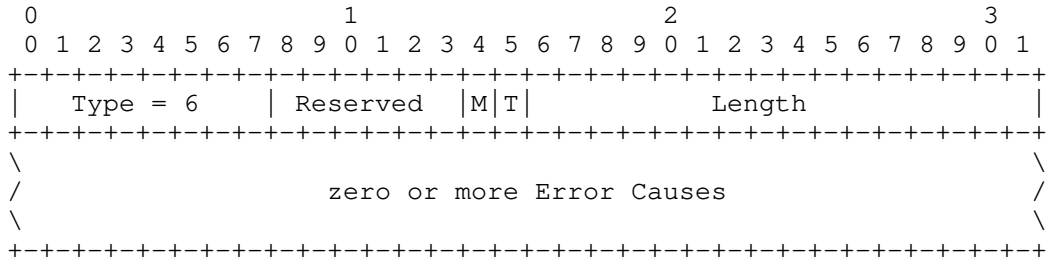
5. Data Formats

This section defines the formats used to support NAT traversal. Section 5.1 and Section 5.2 describe chunks and error causes sent by NAT functions and received by SCTP endpoints. Section 5.3 describes parameters sent by SCTP endpoints and used by NAT functions and SCTP endpoints.

5.1. Modified Chunks

This section presents existing chunks defined in [RFC4960] for which additional flags are specified by this document.

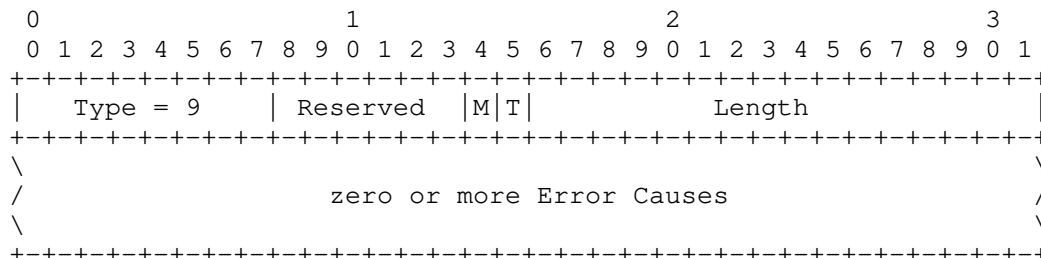
5.1.1. Extended ABORT Chunk



The ABORT chunk is extended to add the new 'M bit'. The M bit indicates to the receiver of the ABORT chunk that the chunk was not generated by the peer SCTP endpoint, but instead by a middle box (e.g., NAT).

[NOTE to RFC-Editor: Assignment of M bit to be confirmed by IANA.]

## 5.1.2. Extended ERROR Chunk



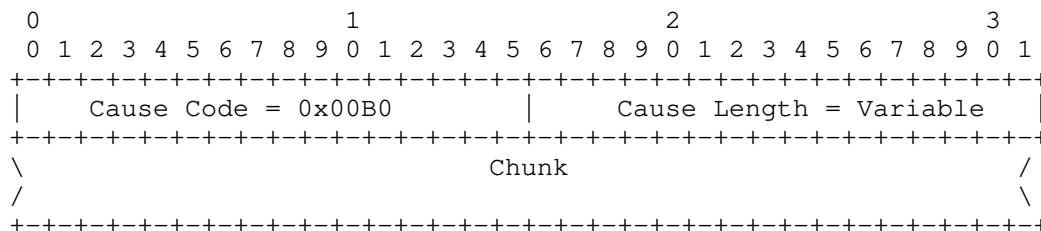
The ERROR chunk defined in [RFC4960] is extended to add the new 'M bit'. The M bit indicates to the receiver of the ERROR chunk that the chunk was not generated by the peer SCTP endpoint, but instead by a middle box.

[NOTE to RFC-Editor: Assignment of M bit to be confirmed by IANA.]

## 5.2. New Error Causes

This section defines the new error causes added by this document.

## 5.2.1. VTag and Port Number Collision Error Cause



Cause Code: 2 bytes (unsigned integer)

This field holds the IANA defined cause code for the 'VTag and Port Number Collision' Error Cause. IANA is requested to assign the value 0x00B0 for this cause code.

Cause Length: 2 bytes (unsigned integer)

This field holds the length in bytes of the error cause. The value MUST be the length of the Cause-Specific Information plus 4.

Chunk: variable length

The Cause-Specific Information is filled with the chunk that caused this error. This can be an INIT, INIT ACK, or ASCONF chunk. Note that if the entire chunk will not fit in the ERROR chunk or ABORT chunk being sent then the bytes that do not fit are truncated.

[NOTE to RFC-Editor: Assignment of cause code to be confirmed by IANA.]

#### 5.2.2. Missing State Error Cause

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Cause Code = 0x00B1										Cause Length = Variable																													
Original Packet																																							

Cause Code: 2 bytes (unsigned integer)

This field holds the IANA defined cause code for the 'Missing State' Error Cause. IANA is requested to assign the value 0x00B1 for this cause code.

Cause Length: 2 bytes (unsigned integer)

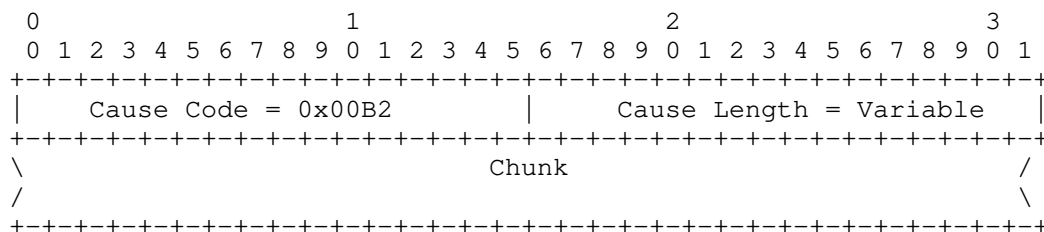
This field holds the length in bytes of the error cause. The value MUST be the length of the Cause-Specific Information plus 4.

Original Packet: variable length

The Cause-Specific Information is filled with the IPv4 or IPv6 packet that caused this error. The IPv4 or IPv6 header MUST be included. Note that if the packet will not fit in the ERROR chunk or ABORT chunk being sent then the bytes that do not fit are truncated.

[NOTE to RFC-Editor: Assignment of cause code to be confirmed by IANA.]

#### 5.2.3. Port Number Collision Error Cause



Cause Code: 2 bytes (unsigned integer)

This field holds the IANA defined cause code for the 'Port Number Collision' Error Cause. IANA is requested to assign the value 0x00B2 for this cause code.

Cause Length: 2 bytes (unsigned integer)

This field holds the length in bytes of the error cause. The value MUST be the length of the Cause-Specific Information plus 4.

Chunk: variable length

The Cause-Specific Information is filled with the chunk that caused this error. This can be an INIT, INIT ACK, or ASCONF chunk. Note that if the entire chunk will not fit in the ERROR chunk or ABORT chunk being sent then the bytes that do not fit are truncated.

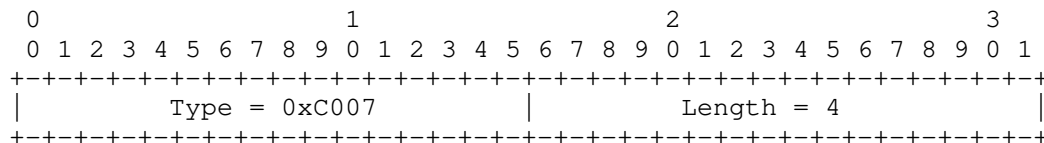
[NOTE to RFC-Editor: Assignment of cause code to be confirmed by IANA.]

### 5.3. New Parameters

This section defines new parameters and their valid appearance defined by this document.

#### 5.3.1. Disable Restart Parameter

This parameter is used to indicate that the restart procedure is requested to be disabled. Both endpoints of an association MUST include this parameter in the INIT chunk and INIT ACK chunk when establishing an association and MUST include it in the ASCONF chunk when adding an address to successfully disable the restart procedure.



Parameter Type: 2 bytes (unsigned integer)

This field holds the IANA defined parameter type for the Disable Restart Parameter. IANA is requested to assign the value 0xC007 for this parameter type.

Parameter Length: 2 bytes (unsigned integer)

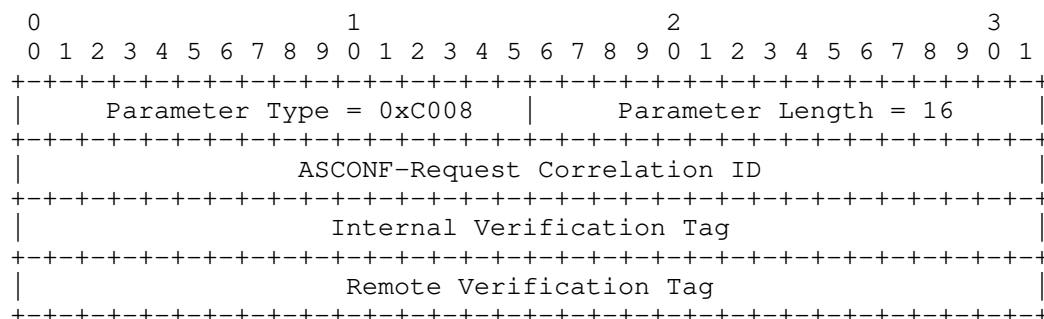
This field holds the length in bytes of the parameter. The value MUST be 4.

[NOTE to RFC-Editor: Assignment of parameter type to be confirmed by IANA.]

The Disable Restart Parameter MAY appear in INIT, INIT ACK and ASCONF chunks and MUST NOT appear in any other chunk.

### 5.3.2. VTags Parameter

This parameter is used to help a NAT function to recover from state loss.



Parameter Type: 2 bytes (unsigned integer)

This field holds the IANA defined parameter type for the VTags Parameter. IANA is requested to assign the value 0xC008 for this parameter type.

Parameter Length: 2 bytes (unsigned integer)

This field holds the length in bytes of the parameter. The value MUST be 16.

ASCONF-Request Correlation ID: 4 bytes (unsigned integer)

This is an opaque integer assigned by the sender to identify each request parameter. The receiver of the ASCONF Chunk will copy this 32-bit value into the ASCONF Response Correlation ID field of the ASCONF ACK response parameter. The sender of the packet containing the ASCONF chunk can use this same value in the ASCONF ACK chunk to find which request the response is for. The receiver MUST NOT change the value of the ASCONF-Request Correlation ID.

Internal Verification Tag: 4 bytes (unsigned integer)

The Verification Tag that the internal host has chosen for the association. The Verification Tag is a unique 32-bit tag that accompanies any incoming SCTP packet for this association to the Internal-Address.

Remote Verification Tag: 4 bytes (unsigned integer)

The Verification Tag that the host holding the Remote-Address has chosen for the association. The VTag is a unique 32-bit tag that accompanies any outgoing SCTP packet for this association to the Remote-Address.

[NOTE to RFC-Editor: Assignment of parameter type to be confirmed by IANA.]

The VTags Parameter MAY appear in ASCONF chunks and MUST NOT appear in any other chunk.

## 6. Procedures for SCTP Endpoints and NAT Functions

If an SCTP endpoint is behind an SCTP-aware NAT, a number of problems can arise as it tries to communicate with its peers:

- \* IP addresses can not be included in the SCTP packet. This is discussed in Section 6.1.
- \* More than one host behind a NAT function could select the same VTag and source port number when communicating with the same peer server. This creates a situation where the NAT function will not be able to tell the two associations apart. This situation is discussed in Section 6.2.
- \* If an SCTP endpoint is a server communicating with multiple peers and the peers are behind the same NAT function, then these peers cannot be distinguished by the server. This case is discussed in Section 6.3.
- \* A restart of a NAT function during a conversation could cause a loss of its state. This problem and its solution is discussed in Section 6.4.
- \* NAT functions need to deal with SCTP packets being fragmented at the IP layer. This is discussed in Section 6.5.
- \* An SCTP endpoint can be behind two NAT functions in parallel providing redundancy. The method to set up this scenario is discussed in Section 6.6.

The mechanisms to solve these problems require additional chunks and parameters, defined in this document, and modified handling procedures from those specified in [RFC4960] as described below.

#### 6.1. Association Setup Considerations for Endpoints

The association setup procedure defined in [RFC4960] allows multi-homed SCTP endpoints to exchange its IP-addresses by using IPv4 or IPv6 address parameters in the INIT and INIT ACK chunks. However, this does not work when NAT functions are present.

Every association setup from a host behind a NAT function MUST NOT use multiple internal addresses. The INIT chunk MUST NOT contain an IPv4 Address parameter, IPv6 Address parameter, or Supported Address Types parameter. The INIT ACK chunk MUST NOT contain any IPv4 Address parameter or IPv6 Address parameter using non-global addresses. The INIT chunk and the INIT ACK chunk MUST NOT contain any Host Name parameters.

If the association is intended to be finally multi-homed, the procedure in Section 6.6 MUST be used.

The INIT and INIT ACK chunk SHOULD contain the Disable Restart parameter defined in Section 5.3.1.

#### 6.2. Handling of Internal Port Number and Verification Tag Collisions

Consider the case where two hosts in the Internal-Address space want to set up an SCTP association with the same service provided by some remote hosts. This means that the Remote-Port is the same. If they both choose the same Internal-Port and Internal-VTag, the NAT function cannot distinguish between incoming packets anymore. However, this is unlikely. The Internal-VTags are chosen at random and if the Internal-Ports are also chosen from the ephemeral port range at random (see [RFC6056]) this gives a 46-bit random number that has to match.

The same can happen with the Remote-VTag when a packet containing an INIT ACK chunk or an ASCONF chunk is processed by the NAT function.

##### 6.2.1. NAT Function Considerations

If the NAT function detects a collision of internal port numbers and verification tags, it SHOULD send a packet containing an ABORT chunk with the M bit set if the collision is triggered by a packet containing an INIT or INIT ACK chunk. If such a collision is triggered by a packet containing an ASCONF chunk, it SHOULD send a packet containing an ERROR chunk with the M bit. The M bit is a new

bit defined by this document to express to SCTP that the source of this packet is a "middle" box, not the peer SCTP endpoint (see Section 5.1.1). If a packet containing an INIT ACK chunk triggers the collision, the corresponding packet containing the ABORT chunk MUST contain the same source and destination address and port numbers as the packet containing the INIT ACK chunk. If a packet containing an INIT chunk or an ASCONF chunk, the source and destination address and port numbers MUST be swapped.

The sender of the packet containing an ERROR or ABORT chunk MUST include the error cause with cause code 'VTag and Port Number Collision' (see Section 5.2.1).

#### 6.2.2. Endpoint Considerations

The sender of the packet containing the INIT chunk or the receiver of a packet containing the INIT ACK chunk, upon reception of a packet containing an ABORT chunk with M bit set and the appropriate error cause code for colliding NAT binding table state is included, SHOULD reinitiate the association setup procedure after choosing a new initiate tag, if the association is in COOKIE-WAIT state. In any other state, the SCTP endpoint MUST NOT respond.

The sender of the packet containing the ASCONF chunk, upon reception of a packet containing an ERROR chunk with M bit set, MUST stop adding the path to the association.

#### 6.3. Handling of Internal Port Number Collisions

When two SCTP hosts are behind an SCTP-aware NAT it is possible that two SCTP hosts in the Internal-Address space will want to set up an SCTP association with the same server running on the same remote host. If the two hosts choose the same internal port, this is considered an internal port number collision.

For the NAT function, appropriate tracking can be performed by assuring that the VTags are unique between the two hosts.

##### 6.3.1. NAT Function Considerations

The NAT function, when processing the packet containing the INIT ACK chunk, SHOULD note in its NAT binding table if the association supports the disable restart extension. This note is used when establishing future associations (i.e. when processing a packet containing an INIT chunk from an internal host) to decide if the connection can be allowed. The NAT function does the following when processing a packet containing an INIT chunk:



- \* If the packet containing the INIT chunk is originating from an internal port to a remote port for which the NAT function has no matching NAT binding table entry, it MUST allow the packet containing the INIT chunk creating an NAT binding table entry.
- \* If the packet containing the INIT chunk matches an existing NAT binding table entry, it MUST validate that the disable restart feature is supported and, if it does, allow the packet containing the INIT chunk to be forwarded.
- \* If the disable restart feature is not supported, the NAT function SHOULD send a packet containing an ABORT chunk with the M bit set.

The 'Port Number Collision' error cause (see Section 5.2.3) MUST be included in the ABORT chunk sent in response to the packet containing an INIT chunk.

If the collision is triggered by a packet containing an ASCONF chunk, a packet containing an ERROR chunk with the 'Port Number Collision' error cause SHOULD be sent in response to the packet containing the ASCONF chunk.

#### 6.3.2. Endpoint Considerations

For the remote SCTP server this means that the Remote-Port and the Remote-Address are the same. If they both have chosen the same Internal-Port the server cannot distinguish between both associations based on the address and port numbers. For the server it looks like the association is being restarted. To overcome this limitation the client sends a Disable Restart parameter in the INIT chunk.

When the server receives this parameter it does the following:

- \* It MUST include a Disable Restart parameter in the INIT ACK to inform the client that it will support the feature.
- \* It MUST disable the restart procedures defined in [RFC4960] for this association.

Servers that support this feature will need to be capable of maintaining multiple connections to what appears to be the same peer (behind the NAT function) differentiated only by the VTags.

#### 6.4. Handling of Missing State

#### 6.4.1. NAT Function Considerations

If the NAT function receives a packet from the internal network for which the lookup procedure does not find an entry in the NAT binding table, a packet containing an ERROR chunk SHOULD be sent back with the M bit set. The source address of the packet containing the ERROR chunk MUST be the destination address of the packet received from the internal network. The verification tag is reflected and the T bit is set. Such a packet containing an ERROR chunk SHOULD NOT be sent if the received packet contains an ASCONF chunk with the VTags parameter or an ABORT, SHUTDOWN COMPLETE or INIT ACK chunk. A packet containing an ERROR chunk MUST NOT be sent if the received packet contains an ERROR chunk with the M bit set. In any case, the packet SHOULD NOT be forwarded to the remote address.

If the NAT function receives a packet from the internal network for which it has no NAT binding table entry and the packet contains an ASCONF chunk with the VTags parameter, the NAT function MUST update its NAT binding table according to the verification tags in the VTags parameter and, if present, the Disable Restart parameter.

When sending a packet containing an ERROR chunk, the error cause 'Missing State' (see Section 5.2.2) MUST be included and the M bit of the ERROR chunk MUST be set (see Section 5.1.2).

#### 6.4.2. Endpoint Considerations

Upon reception of this packet containing the ERROR chunk by an SCTP endpoint the receiver takes the following actions:

- \* It SHOULD validate that the verification tag is reflected by looking at the VTag that would have been included in an outgoing packet. If the validation fails, discard the received packet containing the ERROR chunk.
- \* It SHOULD validate that the peer of the SCTP association supports the dynamic address extension. If the validation fails, discard the received packet containing the ERROR chunk.
- \* It SHOULD generate a packet containing a new ASCONF chunk containing the VTags parameter (see Section 5.3.2) and the Disable Restart parameter (see Section 5.3.1) if the association is using the disable restart feature. By processing this packet the NAT function can recover the appropriate state. The procedures for generating an ASCONF chunk can be found in [RFC5061].

The peer SCTP endpoint receiving such a packet containing an ASCONF chunk SHOULD add the address and respond with an acknowledgment if the address is new to the association (following all procedures defined in [RFC5061]). If the address is already part of the association, the SCTP endpoint MUST NOT respond with an error, but instead SHOULD respond with a packet containing an ASCONF ACK chunk acknowledging the address and take no action (since the address is already in the association).

Note that it is possible that upon receiving a packet containing an ASCONF chunk containing the VTags parameter the NAT function will realize that it has an 'Internal Port Number and Verification Tag collision'. In such a case the NAT function SHOULD send a packet containing an ERROR chunk with the error cause code set to 'VTag and Port Number Collision' (see Section 5.2.1).

If an SCTP endpoint receives a packet containing an ERROR chunk with 'Internal Port Number and Verification Tag collision' as the error cause and the packet in the Error Chunk contains an ASCONF with the VTags parameter, careful examination of the association is necessary. The endpoint does the following:

- \* It MUST validate that the verification tag is reflected by looking at the VTag that would have been included in the outgoing packet. If the validation fails, it MUST discard the packet.
- \* It MUST validate that the peer of the SCTP association supports the dynamic address extension. If the peer does not support this extension, it MUST discard the received packet containing the ERROR chunk.
- \* If the association is attempting to add an address (i.e. following the procedures in Section 6.6) then the endpoint MUST NOT consider the address part of the association and SHOULD make no further attempt to add the address (i.e. cancel any ASCONF timers and remove any record of the path), since the NAT function has a VTag collision and the association cannot easily create a new VTag (as it would if the error occurred when sending a packet containing an INIT chunk).
- \* If the endpoint has no other path, i.e. the procedure was executed due to missing a state in the NAT function, then the endpoint MUST abort the association. This would occur only if the local NAT function restarted and accepted a new association before attempting to repair the missing state (Note that this is no different than what happens to all TCP connections when a NAT function loses its state).

### 6.5. Handling of Fragmented SCTP Packets by NAT Functions

SCTP minimizes the use of IP-level fragmentation. However, it can happen that using IP-level fragmentation is needed to continue an SCTP association. For example, if the path MTU is reduced and there are still some DATA chunk in flight, which require packets larger than the new path MTU. If IP-level fragmentation can not be used, the SCTP association will be terminated in a non-graceful way. See [RFC8900] for more information about IP fragmentation.

Therefore, a NAT function MUST be able to handle IP-level fragmented SCTP packets. The fragments MAY arrive in any order.

When an SCTP packet can not be forwarded by the NAT function due to MTU issues and the IP header forbids fragmentation, the NAT MUST send back a "Fragmentation needed and DF set" ICMPv4 or PTB ICMPv6 message to the internal host. This allows for a faster recovery from this packet drop.

### 6.6. Multi Point Traversal Considerations for Endpoints

If a multi-homed SCTP endpoint behind a NAT function connects to a peer, it MUST first set up the association single-homed with only one address causing the first NAT function to populate its state. Then it SHOULD add each IP address using packets containing ASCONF chunks sent via their respective NAT functions. The address used in the Add IP address parameter is the wildcard address (0.0.0.0 or ::0) and the address parameter in the ASCONF chunk SHOULD also contain the VTags parameter and optionally the Disable Restart parameter.

## 7. SCTP NAT YANG Module

This section defines a YANG module for SCTP NAT.

The terminology for describing YANG data models is defined in [RFC7950]. The meaning of the symbols in tree diagrams is defined in [RFC8340].

### 7.1. Tree Structure

This module augments NAT YANG module [RFC8512] with SCTP specifics. The module supports both classical SCTP NAT (that is, rewrite port numbers) and SCTP-specific variant where the ports numbers are not altered. The YANG "feature" is used to indicate whether SCTP-specific variant is supported.

The tree structure of the SCTP NAT YANG module is provided below:

```

module: ietf-nat-sctp
  augment /nat:nat/nat:instances/nat:instance
    /nat:policy/nat:timers:
      +--rw sctp-timeout?  uint32
  augment /nat:nat/nat:instances/nat:instance
    /nat:mapping-table/nat:mapping-entry:
      +--rw int-VTag?      uint32 {sctp-nat}?
      +--rw rem-VTag?      uint32 {sctp-nat}?

```

Concretely, the SCTP NAT YANG module augments the NAT YANG module (policy, in particular) with the following:

- \* The sctp-timeout is used to control the SCTP inactivity timeout. That is, the time an SCTP mapping will stay active without SCTP packets traversing the NAT. This timeout can be set only for SCTP. Hence, `"/nat:nat/nat:instances/nat:instance/nat:policy/nat:transport-protocols/nat:protocol-id"` MUST be set to `'132'` (SCTP).

In addition, the SCTP NAT YANG module augments the mapping entry with the following parameters defined in Section 3. These parameters apply only for SCTP NAT mapping entries (i.e., `"/nat/instances/instance/mapping-table/mapping-entry/transport-protocol"` MUST be set to `'132'`);

- \* The Internal Verification Tag (Int-VTag)
- \* The Remote Verification Tag (Rem-VTag)

## 7.2. YANG Module

```

<CODE BEGINS> file "ietf-nat-sctp@2020-11-02.yang"
module ietf-nat-sctp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-nat-sctp";
  prefix nat-sctp;

  import ietf-nat {
    prefix nat;
    reference
      "RFC 8512: A YANG Module for Network Address Translation
       (NAT) and Network Prefix Translation (NPT)";
  }

  organization
    "IETF TSVWG Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/tsvwg/>

```

WG List: <mailto:tsvwg@ietf.org>

Author: Mohamed Boucadair  
<mailto:mohamed.boucadair@orange.com>;

description

"This module augments NAT YANG module with Stream Control Transmission Protocol (SCTP) specifics. The extension supports both a classical SCTP NAT (that is, rewrite port numbers) and a, SCTP-specific variant where the ports numbers are not altered.

Copyright (c) 2020 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

revision 2019-11-18 {

description

"Initial revision.";

reference

"RFC XXXX: Stream Control Transmission Protocol (SCTP) Network Address Translation Support";

}

feature sctp-nat {

description

"This feature means that SCTP-specific variant of NAT is supported. That is, avoid rewriting port numbers.";

reference

"Section 4.3 of RFC XXXX.";

}

augment "/nat:nat/nat:instances/nat:instance"

+ "/nat:policy/nat:timers" {

when "/nat:nat/nat:instances/nat:instance"

+ "/nat:policy/nat:transport-protocols"

+ "/nat:protocol-id = 132";

description

"Extends NAT policy with a timeout for SCTP mapping entries.";

```
    leaf sctp-timeout {
      type uint32;
      units "seconds";
      description
        "SCTP inactivity timeout. That is, the time an SCTP
        mapping entry will stay active without packets
        traversing the NAT.";
    }
  }

  augment "/nat:nat/nat:instances/nat:instance"
    + "/nat:mapping-table/nat:mapping-entry" {
    when "nat:transport-protocol = 132";
    if-feature "sctp-nat";
    description
      "Extends the mapping entry with SCTP specifics.";

    leaf int-VTag {
      type uint32;
      description
        "The Internal Verification Tag that the internal
        host has chosen for this communication.";
    }
    leaf rem-VTag {
      type uint32;
      description
        "The Remote Verification Tag that the remote
        peer has chosen for this communication.";
    }
  }
}
<CODE ENDS>
```

## 8. Various Examples of NAT Traversals

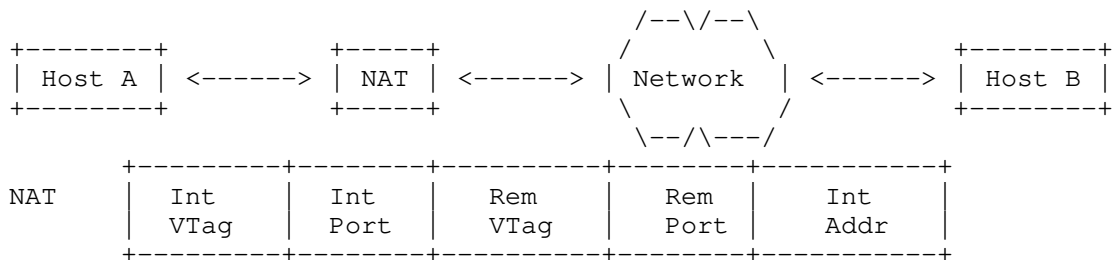
Please note that this section is informational only.

The addresses being used in the following examples are IPv4 addresses for private-use networks and for documentation as specified in [RFC6890]. However, the method described here is not limited to this NAT44 case.

The NAT binding table entries shown in the following examples do not include the flag indicating whether the restart procedure is supported or not. This flag is not relevant for these examples.

## 8.1. Single-homed Client to Single-homed Server

The internal client starts the association with the remote server via a four-way-handshake. Host A starts by sending a packet containing an INIT chunk.



```
INIT[Initiate-Tag = 1234]
10.0.0.1:1 -----> 203.0.113.1:2
    Rem-VTtag = 0
```

A NAT binding tabled entry is created, the source address is substituted and the packet is sent on:

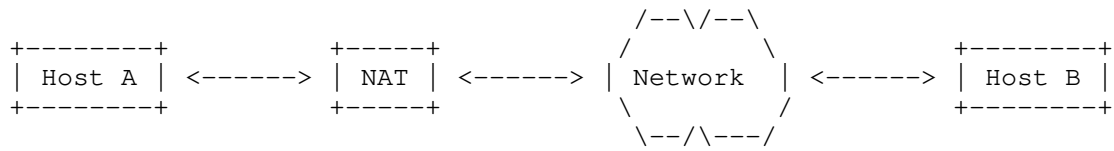
NAT function creates entry:

NAT	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	0	2	10.0.0.1

```
INIT[Initiate-Tag = 1234]
192.0.2.1:1 -----> 203.0.113.1:2
    Rem-VTtag = 0
```

Host B receives the packet containing an INIT chunk and sends a packet containing an INIT ACK chunk with the NAT's Remote-address as destination address.





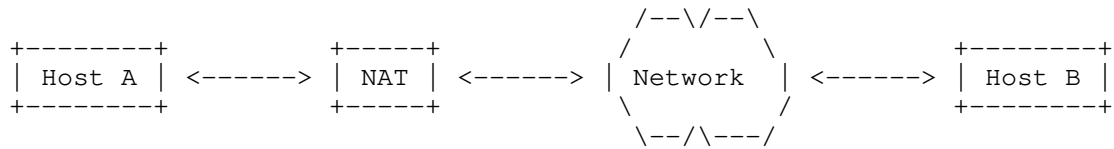
INIT ACK[Initiate-Tag = 5678]  
 192.0.2.1:1 <----- 203.0.113.1:2  
 Int-VTag = 1234

NAT function updates entry:

NAT	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	5678	2	10.0.0.1

INIT ACK[Initiate-Tag = 5678]  
 10.0.0.1:1 <----- 203.0.113.1:2  
 Int-VTag = 1234

The handshake finishes with a COOKIE ECHO acknowledged by a COOKIE ACK.



COOKIE ECHO  
 10.0.0.1:1 -----> 203.0.113.1:2  
 Rem-VTag = 5678

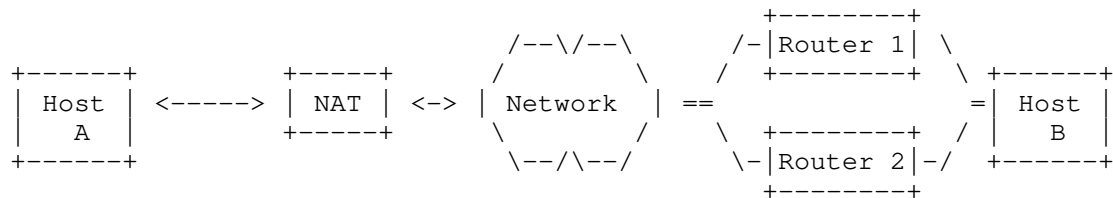
COOKIE ECHO  
 192.0.2.1:1 -----> 203.0.113.1:2  
 Rem-VTag = 5678

COOKIE ACK  
 192.0.2.1:1 <----- 203.0.113.1:2  
 Int-VTag = 1234

COOKIE ACK  
 10.0.0.1:1 <----- 203.0.113.1:2  
 Int-VTag = 1234

## 8.2. Single-homed Client to Multi-homed Server

The internal client is single-homed whereas the remote server is multi-homed. The client (Host A) sends a packet containing an INIT chunk like in the single-homed case.



NAT	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
-----	-------------	-------------	-------------	-------------	-------------

```

INIT[Initiate-Tag = 1234]
10.0.0.1:1 ---> 203.0.113.1:2
    Rem-VTag = 0
  
```

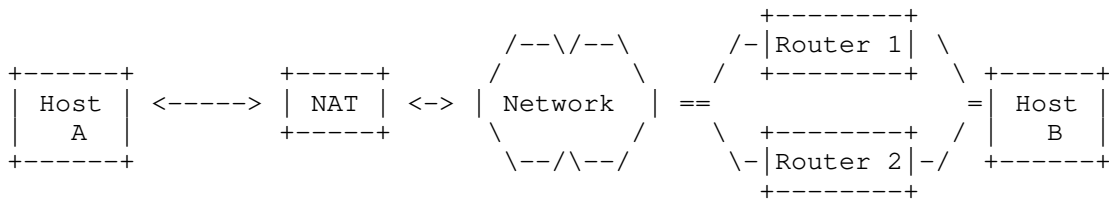
NAT function creates entry:

NAT	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	0	2	10.0.0.1

```

                                INIT[Initiate-Tag = 1234]
192.0.2.1:1 -----> 203.0.113.1:2
                        Rem-VTag = 0
  
```

The server (Host B) includes its two addresses in the INIT ACK chunk.



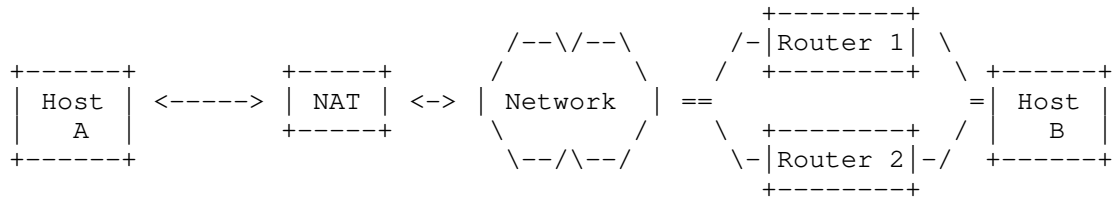
```
INIT ACK[Initiate-tag = 5678, IP-Addr = 203.0.113.129]
192.0.2.1:1 <----- 203.0.113.1:2
                        Int-VTag = 1234
```

The NAT function does not need to change the NAT binding table for the second address:

NAT	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	5678	2	10.0.0.1

```
INIT ACK[Initiate-Tag = 5678]
10.0.0.1:1 <--- 203.0.113.1:2
      Int-VTag = 1234
```

The handshake finishes with a COOKIE ECHO acknowledged by a COOKIE ACK.



COOKIE ECHO  
 10.0.0.1:1 ---> 203.0.113.1:2  
 Rem-VTag = 5678

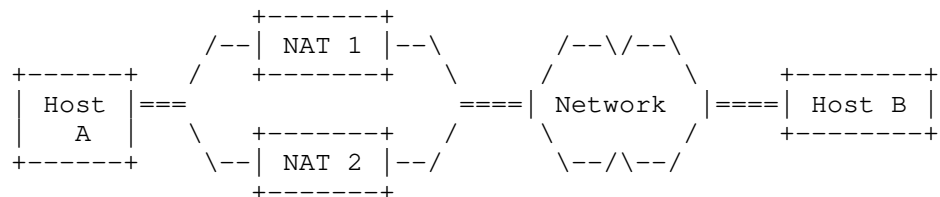
COOKIE ECHO  
 192.0.2.1:1 -----> 203.0.113.1:2  
 Rem-VTag = 5678

COOKIE ACK  
 192.0.2.1:1 <----- 203.0.113.1:2  
 Int-VTag = 1234

COOKIE ACK  
 10.0.0.1:1 <--- 203.0.113.1:2  
 Int-VTag = 1234

### 8.3. Multihomed Client and Server

The client (Host A) sends a packet containing an INIT chunk to the server (Host B), but does not include the second address.



NAT 1	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr

INIT[Initiate-Tag = 1234]  
 10.0.0.1:1 -----> 203.0.113.1:2  
 Rem-VTag = 0

NAT function 1 creates entry:

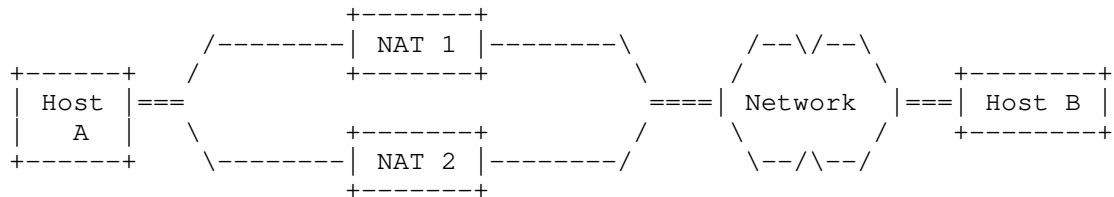
NAT 1	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	0	2	10.0.0.1

```

                                INIT[Initiate-Tag = 1234]
192.0.2.1:1 -----> 203.0.113.1:2
                                Rem-VTag = 0

```

Host B includes its second address in the INIT ACK.



```

INIT ACK[Initiate-Tag = 5678, IP-Addr = 203.0.113.129]
192.0.2.1:1 <----- 203.0.113.1:2
                                Int-VTag = 1234

```

NAT function 1 does not need to update the NAT binding table for the second address:

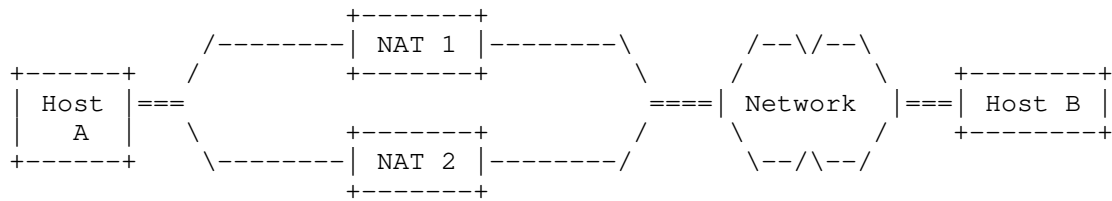
NAT 1	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	5678	2	10.0.0.1

```

INIT ACK[Initiate-Tag = 5678]
10.0.0.1:1 <----- 203.0.113.1:2
                                Int-VTag = 1234

```

The handshake finishes with a COOKIE ECHO acknowledged by a COOKIE ACK.



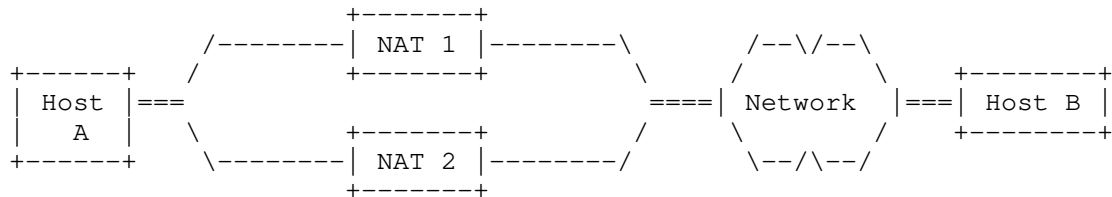
COOKIE ECHO  
 10.0.0.1:1 -----> 203.0.113.1:2  
 Rem-VTag = 5678

COOKIE ECHO  
 192.0.2.1:1 -----> 203.0.113.1:2  
 Rem-VTag = 5678

COOKIE ACK  
 192.0.2.1:1 <----- 203.0.113.1:2  
 Int-VTag = 1234

COOKIE ACK  
 10.0.0.1:1 <----- 203.0.113.1:2  
 Int-VTag = 1234

Host A announces its second address in an ASCONF chunk. The address parameter contains a wildcard address (0.0.0.0 or ::0) to indicate that the source address has to be added. The address parameter within the ASCONF chunk will also contain the pair of VTags (remote and internal) so that the NAT function can populate its NAT binding table entry completely with this single packet.



ASCONF [ADD-IP=0.0.0.0, INT-VTag=1234, Rem-VTag = 5678]  
 10.1.0.1:1 -----> 203.0.113.129:2  
 Rem-VTag = 5678

NAT function 2 creates a complete entry:

NAT 2	+-----+				
	Int	Int	Rem	Rem	Int
	VTag	Port	VTag	Port	Addr
	1234	1	5678	2	10.1.0.1
	+-----+				

```

ASCONF [ADD-IP, Int-VTag=1234, Rem-VTag = 5678]
192.0.2.129:1 -----> 203.0.113.129:2
                        Rem-VTag = 5678

```

```

                        ASCONF ACK
192.0.2.129:1 <----- 203.0.113.129:2
                        Int-VTag = 1234

```

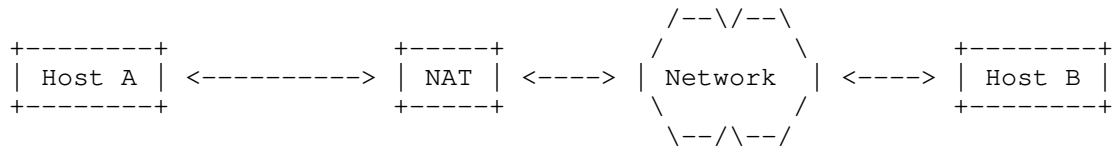
```

                        ASCONF ACK
10.1.0.1:1 <----- 203.0.113.129:2
                        Int-VTag = 1234

```

#### 8.4. NAT Function Loses Its State

Association is already established between Host A and Host B, when the NAT function loses its state and obtains a new external address. Host A sends a DATA chunk to Host B.



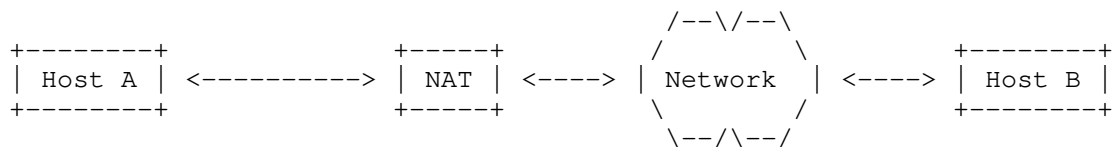
NAT	+-----+				
	Int	Int	Rem	Rem	Int
	VTag	Port	VTag	Port	Addr
	+-----+				

```

                        DATA
10.0.0.1:1 -----> 203.0.113.1:2
                        Rem-VTag = 5678

```

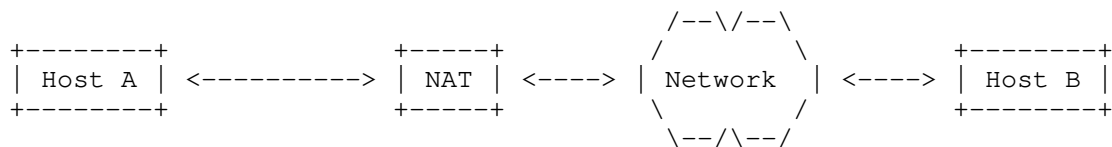
The NAT function cannot find an entry in the NAT binding table for the association. It sends a packet containing an ERROR chunk with the M bit set and the cause "NAT state missing".



```

ERROR [M bit, NAT state missing]
10.0.0.1:1 <----- 203.0.113.1:2
      Rem-VTag = 5678
  
```

On reception of the packet containing the ERROR chunk, Host A sends a packet containing an ASCONF chunk indicating that the former information has to be deleted and the source address of the actual packet added.



```

ASCONF [ADD-IP, DELETE-IP, Int-VTag=1234, Rem-VTag = 5678]
10.0.0.1:1 -----> 203.0.113.129:2
      Rem-VTag = 5678
  
```

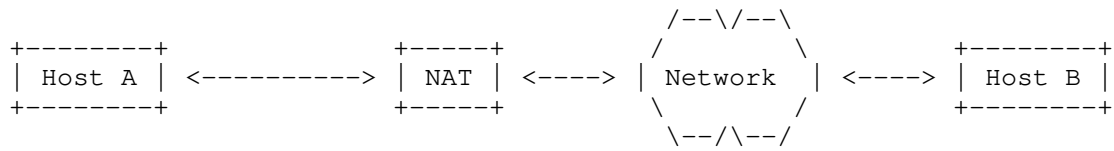
NAT	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	5678	2	10.0.0.1

```

ASCONF [ADD-IP, DELETE-IP, Int-VTag=1234, Rem-VTag = 5678]
      192.0.2.2:1 -----> 203.0.113.129:2
      Rem-VTag = 5678
  
```

Host B adds the new source address to this association and deletes all other addresses from this association.





ASCONF ACK  
 192.0.2.2:1 <----- 203.0.113.129:2  
 Int-VTag = 1234

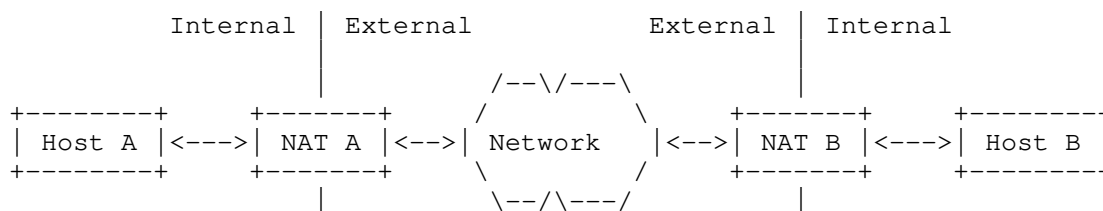
ASCONF ACK  
 10.1.0.1:1 <----- 203.0.113.129:2  
 Int-VTag = 1234

DATA  
 10.0.0.1:1 -----> 203.0.113.1:2  
 Rem-VTag = 5678

DATA  
 192.0.2.2:1 -----> 203.0.113.129:2  
 Rem-VTag = 5678

#### 8.5. Peer-to-Peer Communications

If two hosts, each of them behind a NAT function, want to communicate with each other, they have to get knowledge of the peer's external address. This can be achieved with a so-called rendezvous server. Afterwards the destination addresses are external, and the association is set up with the help of the INIT collision. The NAT functions create their entries according to their internal peer's point of view. Therefore, NAT function A's Internal-VTag and Internal-Port are NAT function B's Remote-VTag and Remote-Port, respectively. The naming (internal/remote) of the verification tag in the packet flow is done from the sending host's point of view.



## NAT Binding Tables

NAT A	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
-------	-------------	-------------	-------------	-------------	-------------

NAT B	Int v-tag	Int port	Rem v-tag	Rem port	Int Addr
-------	--------------	-------------	--------------	-------------	-------------

```
INIT[Initiate-Tag = 1234]
10.0.0.1:1 --> 203.0.113.1:2
      Rem-VTag = 0
```

NAT function A creates entry:

NAT A	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	0	2	10.0.0.1

```

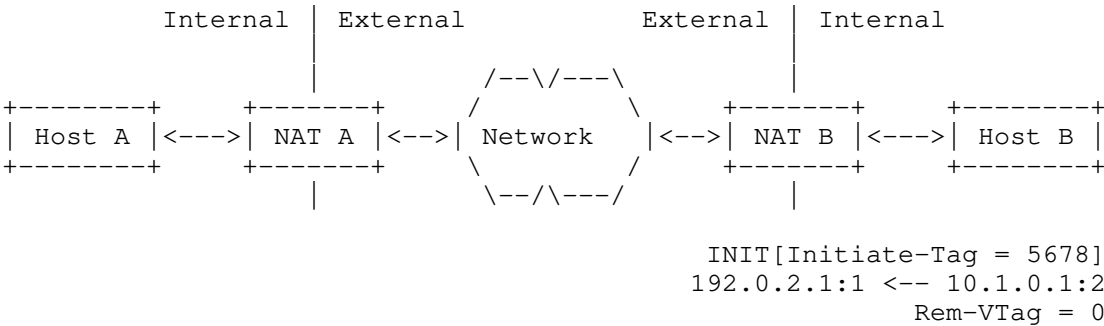
INIT[Initiate-Tag = 1234]
192.0.2.1:1 -----> 203.0.113.1:2
Rem-VTag = 0

```

NAT function B processes the packet containing the INIT chunk, but cannot find an entry. The SCTP packet is silently discarded and leaves the NAT binding table of NAT function B unchanged.

NAT B	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
-------	-------------	-------------	-------------	-------------	-------------

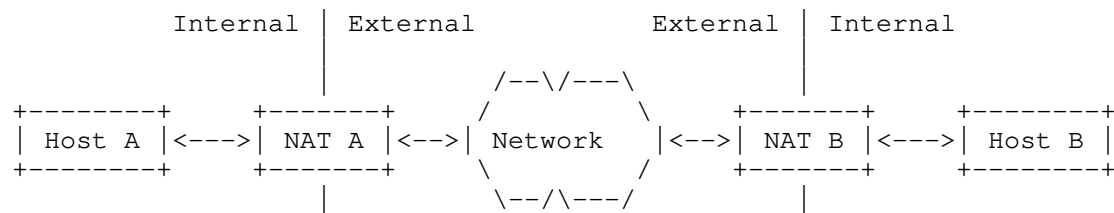
Now Host B sends a packet containing an INIT chunk, which is processed by NAT function B. Its parameters are used to create an entry.



NAT B	Int	Int	Rem	Rem	Int
	VTag	Port	VTag	Port	Addr
	5678	2	0	1	10.1.0.1

INIT[Initiate-Tag = 5678]  
192.0.2.1:1 <----- 203.0.113.1:2  
Rem-VTag = 0

NAT function A processes the packet containing the INIT chunk. As the outgoing packet containing an INIT chunk of Host A has already created an entry, the entry is found and updated:

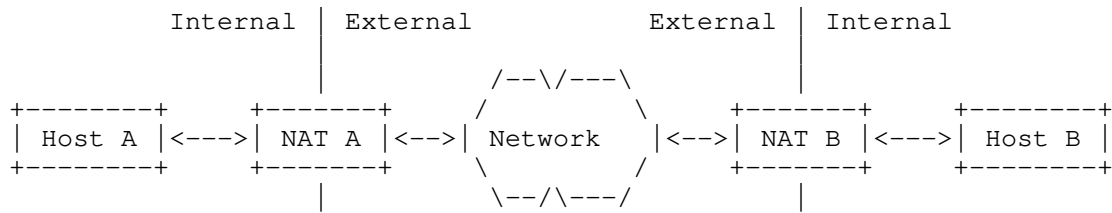


VTag != Int-VTag, but Rem-VTag == 0, find entry.

NAT A	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	5678	2	10.0.0.1

```
INIT[Initiate-tag = 5678]
10.0.0.1:1 <-- 203.0.113.1:2
    Rem-VTag = 0
```

Host A sends a packet containing an INIT ACK chunk, which can pass through NAT function B:



```

INIT ACK[Initiate-Tag = 1234]
10.0.0.1:1 --> 203.0.113.1:2
    Rem-VTag = 5678
  
```

```

          INIT ACK[Initiate-Tag = 1234]
192.0.2.1:1 -----> 203.0.113.1:2
          Rem-VTag = 5678
  
```

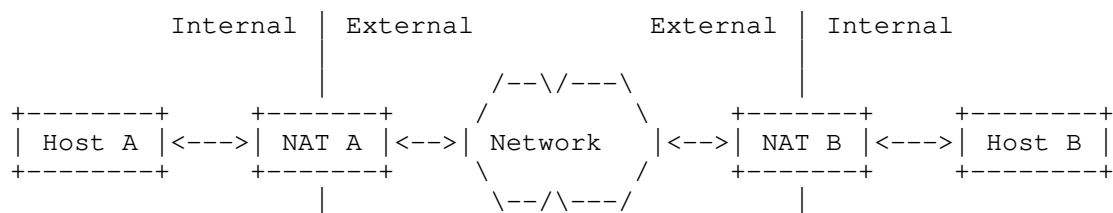
NAT function B updates entry:

NAT B	Int	Int	Rem	Rem	Int
	VTag	Port	VTag	Port	Addr
	5678	2	1234	1	10.1.0.1

```

INIT ACK[Initiate-Tag = 1234]
192.0.2.1:1 --> 10.1.0.1:2
    Rem-VTag = 5678
  
```

The lookup for COOKIE ECHO and COOKIE ACK is successful.



COOKIE ECHO  
 192.0.2.1:1 <-- 10.1.0.1:2  
 Rem-VTag = 1234

COOKIE ECHO  
 192.0.2.1:1 <----- 203.0.113.1:2  
 Rem-VTag = 1234

COOKIE ECHO  
 10.0.0.1:1 <-- 203.0.113.1:2  
 Rem-VTag = 1234

COOKIE ACK  
 10.0.0.1:1 --> 203.0.113.1:2  
 Rem-VTag = 5678

COOKIE ACK  
 192.0.2.1:1 -----> 203.0.113.1:2  
 Rem-VTag = 5678

COOKIE ACK  
 192.0.2.1:1 --> 10.1.0.1:2  
 Rem-VTag = 5678

## 9. Socket API Considerations

This section describes how the socket API defined in [RFC6458] is extended to provide a way for the application to control NAT friendliness.

Please note that this section is informational only.

A socket API implementation based on [RFC6458] is extended by supporting one new read/write socket option.

### 9.1. Get or Set the NAT Friendliness (SCTP\_NAT\_FRIENDLY)

This socket option uses the option\_level IPPROTO\_SCTP and the option\_name SCTP\_NAT\_FRIENDLY. It can be used to enable/disable the NAT friendliness for future associations and retrieve the value for future and specific ones.

```
struct sctp_assoc_value {  
    sctp_assoc_t assoc_id;  
    uint32_t assoc_value;  
};
```

assoc\_id

This parameter is ignored for one-to-one style sockets. For one-to-many style sockets the application can fill in an association identifier or SCTP\_FUTURE\_ASSOC for this query. It is an error to use SCTP\_{CURRENT|ALL}\_ASSOC in assoc\_id.

assoc\_value

A non-zero value indicates a NAT-friendly mode.

## 10. IANA Considerations

[NOTE to RFC-Editor: "RFCXXXX" is to be replaced by the RFC number you assign this document.]

[NOTE to RFC-Editor: The requested values for the chunk type and the chunk parameter types are tentative and to be confirmed by IANA.]

This document (RFCXXXX) is the reference for all registrations described in this section. The requested changes are described below.

### 10.1. New Chunk Flags for Two Existing Chunk Types

As defined in [RFC6096] two chunk flags have to be assigned by IANA for the ERROR chunk. The requested value for the T bit is 0x01 and for the M bit is 0x02.

This requires an update of the "ERROR Chunk Flags" registry for SCTP:

ERROR Chunk Flags

Chunk Flag Value	Chunk Flag Name	Reference
0x01	T bit	[RFCXXXX]
0x02	M bit	[RFCXXXX]
0x04	Unassigned	
0x08	Unassigned	
0x10	Unassigned	
0x20	Unassigned	
0x40	Unassigned	
0x80	Unassigned	

Table 2

As defined in [RFC6096] one chunk flag has to be assigned by IANA for the ABORT chunk. The requested value of the M bit is 0x02.

This requires an update of the "ABORT Chunk Flags" registry for SCTP:

ABORT Chunk Flags



Chunk Flag Value	Chunk Flag Name	Reference
0x01	T bit	[RFC4960]
0x02	M bit	[RFCXXXX]
0x04	Unassigned	
0x08	Unassigned	
0x10	Unassigned	
0x20	Unassigned	
0x40	Unassigned	
0x80	Unassigned	

Table 3

#### 10.2. Three New Error Causes

Three error causes have to be assigned by IANA. It is requested to use the values given below.

This requires three additional lines in the "Error Cause Codes" registry for SCTP:

##### Error Cause Codes

Value	Cause Code	Reference
176	VTag and Port Number Collision	[RFCXXXX]
177	Missing State	[RFCXXXX]
178	Port Number Collision	[RFCXXXX]

Table 4

### 10.3. Two New Chunk Parameter Types

Two chunk parameter types have to be assigned by IANA. IANA is requested to assign these values from the pool of parameters with the upper two bits set to '11' and to use the values given below.

This requires two additional lines in the "Chunk Parameter Types" registry for SCTP:

#### Chunk Parameter Types

ID Value	Chunk Parameter Type	Reference
49159	Disable Restart (0xC007)	[RFCXXXX]
49160	VTags (0xC008)	[RFCXXXX]

Table 5

### 10.4. One New URI

An URI in the "ns" subregistry within the "IETF XML" registry has to be assigned by IANA ([RFC3688]):

URI: urn:ietf:params:xml:ns:yang:ietf-nat-sctp  
 Registrant Contact: The IESG.  
 XML: N/A; the requested URI is an XML namespace.

### 10.5. One New YANG Module

An YANG module in the "YANG Module Names" subregistry within the "YANG Parameters" registry has to be assigned by IANA ([RFC6020]):

Name: ietf-nat-sctp  
 Namespace: urn:ietf:params:xml:ns:yang:ietf-nat-sctp  
 Maintained by IANA: N  
 Prefix: nat-sctp  
 Reference: RFCXXXX

## 11. Security Considerations

State maintenance within a NAT function is always a subject of possible Denial Of Service attacks. This document recommends that at a minimum a NAT function runs a timer on any SCTP state so that old association state can be cleaned up.

Generic issues related to address sharing are discussed in [RFC6269] and apply to SCTP as well.

For SCTP endpoints not disabling the restart procedure, this document does not add any additional security considerations to the ones given in [RFC4960], [RFC4895], and [RFC5061].

SCTP endpoints disabling the restart procedure, need to monitor the status of all associations to mitigate resource exhaustion attacks by establishing a lot of associations sharing the same IP addresses and port numbers.

In any case, SCTP is protected by the verification tags and the usage of [RFC4895] against off-path attackers.

For IP-level fragmentation and reassembly related issues see [RFC4963].

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

All data nodes defined in the YANG module that can be created, modified, and deleted (i.e., config true, which is the default) are considered sensitive. Write operations (e.g., edit-config) applied to these data nodes without proper protection can negatively affect network operations. An attacker who is able to access the SCTP NAT function can undertake various attacks, such as:

- \* Setting a low timeout for SCTP mapping entries to cause failures to deliver incoming SCTP packets.
- \* Instantiating mapping entries to cause NAT collision.

## 12. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4895] Tuexen, M., Stewart, R., Lei, P., and E. Rescorla, "Authenticated Chunks for the Stream Control Transmission Protocol (SCTP)", RFC 4895, DOI 10.17487/RFC4895, August 2007, <<https://www.rfc-editor.org/info/rfc4895>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<https://www.rfc-editor.org/info/rfc4960>>.
- [RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", RFC 5061, DOI 10.17487/RFC5061, September 2007, <<https://www.rfc-editor.org/info/rfc5061>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6096] Tuexen, M. and R. Stewart, "Stream Control Transmission Protocol (SCTP) Chunk Flags Registration", RFC 6096, DOI 10.17487/RFC6096, January 2011, <<https://www.rfc-editor.org/info/rfc6096>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8512] Boucadair, M., Ed., Sivakumar, S., Jacquenet, C., Vinapamula, S., and Q. Wu, "A YANG Module for Network Address Translation (NAT) and Network Prefix Translation (NPT)", RFC 8512, DOI 10.17487/RFC8512, January 2019, <<https://www.rfc-editor.org/info/rfc8512>>.

### 13. Informative References

- [DOI\_10.1145\_1496091.1496095]  
Hayes, D., But, J., and G. Armitage, "Issues with network address translation for SCTP", ACM SIGCOMM Computer Communication Review Vol. 39, pp. 23-33, DOI 10.1145/1496091.1496095, December 2008, <<https://doi.org/10.1145/1496091.1496095>>.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<https://www.rfc-editor.org/info/rfc3022>>.
- [RFC4787] Audet, F., Ed. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, DOI 10.17487/RFC4787, January 2007, <<https://www.rfc-editor.org/info/rfc4787>>.
- [RFC4963] Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly Errors at High Data Rates", RFC 4963, DOI 10.17487/RFC4963, July 2007, <<https://www.rfc-editor.org/info/rfc4963>>.
- [RFC5382] Guha, S., Ed., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, DOI 10.17487/RFC5382, October 2008, <<https://www.rfc-editor.org/info/rfc5382>>.
- [RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, DOI 10.17487/RFC5508, April 2009, <<https://www.rfc-editor.org/info/rfc5508>>.
- [RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, DOI 10.17487/RFC6056, January 2011, <<https://www.rfc-editor.org/info/rfc6056>>.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<https://www.rfc-editor.org/info/rfc6146>>.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <<https://www.rfc-editor.org/info/rfc6269>>.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <<https://www.rfc-editor.org/info/rfc6333>>.
- [RFC6458] Stewart, R., Tuexen, M., Poon, K., Lei, P., and V. Yasevich, "Sockets API Extensions for the Stream Control Transmission Protocol (SCTP)", RFC 6458, DOI 10.17487/RFC6458, December 2011, <<https://www.rfc-editor.org/info/rfc6458>>.
- [RFC6890] Cotton, M., Vegoda, L., Bonica, R., Ed., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, DOI 10.17487/RFC6890, April 2013, <<https://www.rfc-editor.org/info/rfc6890>>.
- [RFC6951] Tuexen, M. and R. Stewart, "UDP Encapsulation of Stream Control Transmission Protocol (SCTP) Packets for End-Host to End-Host Communication", RFC 6951, DOI 10.17487/RFC6951, May 2013, <<https://www.rfc-editor.org/info/rfc6951>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC7857] Penno, R., Perreault, S., Boucadair, M., Ed., Sivakumar, S., and K. Naito, "Updates to Network Address Translation (NAT) Behavioral Requirements", BCP 127, RFC 7857, DOI 10.17487/RFC7857, April 2016, <<https://www.rfc-editor.org/info/rfc7857>>.

- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <<https://www.rfc-editor.org/info/rfc8900>>.

#### Acknowledgments

The authors wish to thank Mohamed Boucadair, Gorrry Fairhurst, Bryan Ford, David Hayes, Alfred Hines, Karen E. E. Nielsen, Henning Peters, Maksim Proshin, Timo Völker, Dan Wing, and Qiaobing Xie for their invaluable comments.

In addition, the authors wish to thank David Hayes, Jason But, and Grenville Armitage, the authors of [DOI\_10.1145\_1496091.1496095], for their suggestions.

The authors also wish to thank Mohamed Boucadair for contributing the text related to the YANG module.

#### Authors' Addresses

Randall R. Stewart  
Netflix, Inc.  
Chapin, SC 29036  
United States of America

Email: [randall@lakerest.net](mailto:randall@lakerest.net)

Michael Tüxen  
Münster University of Applied Sciences  
Stegerwaldstrasse 39  
48565 Steinfurt  
Germany

Email: [tuexen@fh-muenster.de](mailto:tuexen@fh-muenster.de)

Irene Rüngeler  
Münster University of Applied Sciences  
Stegerwaldstrasse 39  
48565 Steinfurt  
Germany

Email: [i.ruengeler@fh-muenster.de](mailto:i.ruengeler@fh-muenster.de)



TSVWG  
Internet Draft  
Intended status: Best Current Practice  
Expires: October 2015

J. Touch  
USC/ISI  
April 24, 2015

Recommendations on Using Assigned Transport Port Numbers  
draft-ietf-tsvwg-port-use-11.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on October 24, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Abstract

This document provides recommendations to application and service protocol designers on how to use the assigned transport protocol port number space and when to request a port assignment from IANA. It provides designer guidelines on how to interact with the IANA processes defined in RFC6335, thus serving to complement (but not update) that document.

## Table of Contents

1. Introduction.....	2
2. Conventions used in this document.....	3
3. History.....	3
4. Current Port Number Use.....	5
5. What is a Port Number?.....	5
6. Conservation.....	7
6.1. Guiding Principles.....	7
6.2. Firewall and NAT Considerations.....	8
7. Considerations for Requesting Port Number Assignments.....	9
7.1. Is a port number assignment necessary?.....	9
7.2. How Many Assigned Port Numbers?.....	11
7.3. Picking an Assigned Port Number.....	12
7.4. Support for Security.....	13
7.5. Support for Future Versions.....	14
7.6. Transport Protocols.....	15
7.7. When to Request an Assignment.....	16
7.8. Squatting.....	17
7.9. Other Considerations.....	18
8. Security Considerations.....	18
9. IANA Considerations.....	19
10. References.....	19
10.1. Normative References.....	19
10.2. Informative References.....	20
11. Acknowledgments.....	22

## 1. Introduction

This document provides information and advice to application and service designers on the use of assigned transport port numbers. It provides a detailed historical background of the evolution of transport port numbers and their multiple meanings. It also provides specific recommendations to designers on how to use assigned port numbers. Note that this document provides information to potential port number applicants that complements the IANA process described in BCP165 [RFC6335], but it does not change any of the port number

assignment procedures described therein. This document is intended to address concerns typically raised during Expert Review of assigned port number applications, but it is not intended to bind those reviews. RFC 6335 also describes the interaction between port experts and port requests in IETF consensus document. Authors of IETF consensus documents should nevertheless follow the advice in this document and can expect comment on their port requests from the port experts during IETF last call or at other times when review is explicitly sought.

## 2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a statement using the key words listed above. This convention aids reviewers in quickly identifying or finding requirements for registration and recommendations for use of port numbers in this RFC.

## 3. History

The term 'port' was first used in [RFC33] to indicate a simplex communication path from an individual process and originally applied to only the Network Control Program (NCP) connection-oriented protocol. At a meeting described in [RFC37], an idea was presented to decouple connections between processes and links that they use as paths, and thus to include numeric source and destination socket identifiers in packets. [RFC38] provides further detail, describing how processes might have more than one of these paths and that more than one path may be active at a time. As a result, there was the need to add a process identifier to the header of each message so that incoming messages could be demultiplexed to the appropriate process. [RFC38] further suggested that 32 bit numbers would be used for these identifiers. [RFC48] discusses the current notion of listening on a specific port number, but does not discuss the issue of port number determination. [RFC61] notes that the challenge of knowing the appropriate port numbers is "left to the processes" in general, but introduces the concept of a "well-known" port number for common services.

[RFC76] proposed a "telephone book" by which an index would allow port numbers to be used by name, but still assumed that both source and destination port numbers are fixed by such a system. [RFC333] proposed that a port number pair, rather than an individual port number, would be used on both sides of the connection for demultiplexing messages. This is the final view in [RFC793] (and its predecessors, including [IEN112]), and brings us to their current meaning. [RFC739] introduced the notion of generic reserved port numbers for groups of protocols, such as "any private RJE server" [RFC739]. Although the overall range of such port numbers was (and remains) 16 bits, only the first 256 (high 8 bits cleared) in the range were considered assigned.

[RFC758] is the first to describe port numbers as being used for TCP (previous RFCs all refer to only NCP). It includes a list of such well-known port numbers, as well as describing ranges used for different purposes:

Decimal	Octal	
---------	-------	--

-----

0-63	0-77	Network Wide Standard Function
64-127	100-177	Hosts Specific Functions
128-223	200-337	Reserved for Future Use
224-255	340-377	Any Experimental Function

In [RFC820] those range meanings disappeared, and a single list of number assignments is presented. This is also the first time that port numbers are described as applying to a connectionless transport (UDP) rather than only connection-oriented transports.

By [RFC900] the ranges appeared as decimal numbers rather than the octal ranges used previously. [RFC1340] increased this range from 0..255 to 0..1023, and began to list TCP and UDP port number assignments individually (although the assumption was that once assigned a port number applies to all transport protocols, including TCP, UDP, recently SCTP and DCCP, as well as ISO-TP4 for a brief period in the early 1990s). [RFC1340] also established the Registered range of 1024-59151, though it notes that it is not controlled by the IANA at that point. The list provided by [RFC1700] in 1994 remained the standard until it was declared replaced by an on-line version, as of [RFC3232] in 2002.

#### 4. Current Port Number Use

RFC6335 indicates three ranges of port number assignments:

Binary	Hex	
-----		
0-1023	0x0000-0x03FF	System (also Well-Known)
1024-49151	0x0400-0xBFFF	User (also Registered)
49152-65535	0xC000-0xFFFF	Dynamic (also Private)

System (also Well-Known) encompasses the range 0..1023. On some systems, use of these port numbers requires privileged access, e.g., that the process run as 'root' (i.e., as a privileged user), which is why these are referred to as System port numbers. The port numbers from 1024..49151 denotes non-privileged services, known as User (also Registered), because these port numbers do not run with special privileges. Dynamic (also Private) port numbers are not assigned.

Both System and User port numbers are assigned through IANA, so both are sometimes called 'registered port numbers'. As a result, the term 'registered' is ambiguous, referring either to the entire range 0-49151 or to the User port numbers. Complicating matters further, System port numbers do not always require special (i.e., 'root') privilege. For clarity, the remainder of this document refers to the port number ranges as System, User, and Dynamic, to be consistent with IANA process [RFC6335].

#### 5. What is a Port Number?

A port number is a 16-bit number used for two distinct purposes:

- o Demultiplexing transport endpoint associations within an end host
- o Identifying a service

The first purpose requires that each transport endpoint association (e.g., TCP connection or UDP pairwise association) using a given transport between a given pair of IP addresses use a different pair of port numbers, but does not require either coordination or registration of port number use. It is the second purpose that drives the need for a common registry.

Consider a user wanting to run a web server. That service could run on any port number, provided that all clients knew what port number to use to access that service at that host. Such information can be explicitly distributed - for example, by putting it in the URI:

`http://www.example.com:51509/`

Ultimately, the correlation of a service with a port number is an agreement between just the two endpoints of the association. A web server can run on port number 53, which might appear as DNS traffic to others but will connect to browsers that know to use port number 53 rather than 80.

As a concept, a service is the combination of ISO Layers 5-7 that represents an application protocol capability. For example www (port number 80) is a service that uses HTTP as an application protocol and provides access to a web server [RFC7230]. However, it is possible to use HTTP for other purposes, such as command and control. This is why some current services (HTTP, e.g.) are a bit overloaded - they describe not only the application protocol, but a particular service.

IANA assigns port numbers so that Internet endpoints do not need pairwise, explicit coordination of the meaning of their port numbers. This is the primary reason for requesting port number assignment by IANA - to have a common agreement between all endpoints on the Internet as to the default meaning of a port number, which provides the endpoints with a default port number for a particular protocol or service.

Port numbers are sometimes used by intermediate devices on a network path, either to monitor available services, to monitor traffic (e.g., to indicate the data contents), or to intercept traffic (to block, proxy, relay, aggregate, or otherwise process it). In each case, the intermediate device interprets traffic based on the port number. It is important to recognize that any interpretation of port numbers - except at the endpoints - may be incorrect, because port numbers are meaningful only at the endpoints. Further, port numbers may not be visible to these intermediate devices, such as when the transport protocol is encrypted (as in network- or link-layer tunnels), or when a packet is fragmented (in which case only the first fragment has the port number information). Such port number invisibility may interfere with these in-network port number-based capabilities.

Port numbers can also be used for other purposes. Assigned port numbers can simplify end system configuration, so that individual

installations do not need to coordinate their use of arbitrary port numbers. Such assignments may also have the effect of simplifying firewall management, so that a single, fixed firewall configuration can either permit or deny a service that uses the assigned ports.

It is useful to differentiate a port number from a service name. The former is a numeric value that is used directly in transport protocol headers as a demultiplexing and service identifier. The latter is primarily a user convenience, where the default map between the two is considered static and resolved using a cached index. This document focuses on the former because it is the fundamental network resource. Dynamic maps between the two, i.e., using DNS SRV records, are discussed further in Section 7.1.

## 6. Conservation

Assigned port numbers are a limited resource that is globally shared by the entire Internet community. As of 2014, approximately 5850 TCP and 5570 UDP port numbers have been assigned out of a total range of 49151. As a result of past conservation, current assigned port use is small and the current rate of assignment avoids the need for transition to larger number spaces. This conservation also helps avoid the need for IANA to rely on assigned port number reclamation, which is practically impossible even though procedurally permitted [RFC6335].

IANA aims to assign only one port number per service, including variants [RFC6335], but there are other benefits to using fewer port numbers for a given service. Use of multiple assigned port numbers can make applications more fragile, especially when firewalls block a subset of those port numbers or use ports numbers to route or prioritize traffic differently. As a result:

>> Each assigned port requested MUST be justified by the applicant as an independently useful service.

### 6.1. Guiding Principles

This document provides recommendations for users that also help conserve assigned port number space. Again, this document does not update BCP165 [RFC6335], which describes the IANA procedures for managing assigned transport port numbers and services. Assigned port number conservation is based on a number of basic principles:

- o A single assigned port number can support different functions over separate endpoint associations, determined using in-band information. An FTP data connection can transfer binary or text files, the latter translating line-terminators, as indicated in-band over the control port number [RFC959].
- o A single assigned port number can indicate the Dynamic port number(s) on which different capabilities are supported, as with passive-mode FTP [RFC959].
- o Several existing services can indicate the Dynamic port number(s) on which other services are supported, such as with mDNS and portmapper [RFC1833] [RFC6762] [RFC6763].
- o Copies of some existing services can be differentiated using in-band information (e.g., URIs in HTTP Host field and TLS Server Name Indication extension) [RFC7230] [RFC6066].
- o Services requiring varying performance properties can already be supported using separate endpoint associations (connections or other associations), each configured to support the desired properties. E.g., a high-speed and low-speed variant can be determined within the service using the same assigned port.

Assigned port numbers are intended to differentiate services, not variations of performance, replicas, pairwise endpoint associations, or payload types. Assigned port numbers are also a small space compared to other Internet number spaces; it is never appropriate to consume assigned port numbers to conserve larger spaces such as IP addresses, especially where copies of a service represent different endpoints.

## 6.2. Firewall and NAT Considerations

Ultimately, port numbers indicate services only to the endpoints, and any intermediate device that assigns meaning to a value can be incorrect. End systems might agree to run web services (HTTP) over port number 53 (typically used for DNS) rather than port number 80, at which point a firewall that blocks port number 80 but permits port number 53 would not have the desired effect. Nonetheless, assigned port numbers are often used to help configure firewalls and other port-based systems for access control.

Using Dynamic port numbers, or explicitly-indicated port numbers indicated in-band over another service (such as with FTP) often complicates firewall and NAT interactions [RFC959]. FTP over firewalls often requires direct support for deep-packet inspection



(to snoop for the Dynamic port number for the NAT to correctly map) or passive-mode FTP (in which both connections are opened from the client side).

## 7. Considerations for Requesting Port Number Assignments

Port numbers are assigned by IANA by a set of documented procedures [RFC6335]. The following section describes the steps users can take to help assist with responsible use of assigned port numbers, and with preparing an application for a port number assignment.

### 7.1. Is a port number assignment necessary?

First, it is useful to consider whether a port number assignment is required. In many cases, a new number assignment may not be needed, for example:

- o Is this really a new service, or can an existing service suffice?
- o Is this an experimental service [RFC3692]? If so, consider using the current experimental ports [RFC2780].
- o Is this service independently useful? Some systems are composed from collections of different service capabilities, but not all component functions are useful as independent services. Port numbers are typically shared among the smallest independently-useful set of functions. Different service uses or properties can be supported in separate pairwise endpoint associations after an initial negotiation, e.g., to support software decomposition.
- o Can this service use a Dynamic port number that is coordinated out-of-band, e.g.:
  - o By explicit configuration of both endpoints.
  - o By internal mechanisms within the same host (e.g., a configuration file, indicated within a URI, or using interprocess communication).
- o Using information exchanged on a related service: FTP, SIP, etc. [RFC959] [RFC3261].
- o Using an existing port discovery service: portmapper, mDNS, etc. [RFC1833] [RFC6762] [RFC6763].

There are a few good examples of reasons that more directly suggest that not only is a port number assignment not necessary, but it is directly counter-indicated:

- o Assigned port numbers are not intended to differentiate performance variations within the same service, e.g., high-speed vs. ordinary speed. Performance variations can be supported within a single assigned port number in context of separate pairwise endpoint associations.
- o Additional assigned port numbers are not intended to replicate an existing service. For example, if a device is configured to use a typical web browser then it the port number used for that service is a copy of the http service that is already assigned to port number 80 and does not warrant a new assignment. However, an automated system that happens to use HTTP framing - but is not primarily accessed by a browser - might be a new service. A good way to tell is "can an unmodified client of the existing service interact with the proposed service"? If so, that service would be a copy of an existing service and would not merit a new assignment.
- o Assigned port numbers not intended for intra-machine communication. Such communication can already be supported by internal mechanisms (interprocess communication, shared memory, shared files, etc.). When Internet communication within a host is desired, the server can bind to a Dynamic port that is indicated to the client using these internal mechanisms.
- o Separate assigned port numbers are not intended for insecure versions of existing (or new) secure services. A service that already requires security would be made more vulnerable by having the same capability accessible without security.

Note that the converse is different, i.e., it can be useful to create a new, secure service that replicates an existing insecure service on a new port number assignment. This can be necessary when the existing service is not backward-compatible with security enhancements, such as the use of TLS [RFC5246] or DTLS [RFC6347].

- o Assigned port numbers are not intended for indicating different service versions. Version differentiation should be handled in-band, e.g., using a version number at the beginning of an association (e.g., connection or other transaction). This may not be possible with legacy assignments, but all new services should incorporate support for version indication.

Some services may not need assigned port numbers at all, e.g., SIP allows voice calls to use Dynamic ports [RFC3261]. Some systems can register services in the DNS, using SRV entries. These services can be discovered by a variety of means, including mDNS, or via direct query [RFC6762] [RFC6763]. In such cases, users can more easily request a SRV name, which are assigned first-come, first-served from a much larger namespace.

IANA assigns port numbers, but this assignment is typically used only for servers, i.e., the host that listens for incoming connections or other associations. Clients, i.e., hosts that initiate connections or other associations, typically refer to those assigned port numbers but do not need port number assignments for their endpoint.

Finally, an assigned port number is not a guarantee of exclusive use. Traffic for any service might appear on any port number, due to misconfiguration or deliberate misuse. Application and service designers are encouraged to validate traffic based on its content.

## 7.2. How Many Assigned Port Numbers?

As noted earlier, systems might require a single port number assignment, but rarely require multiple port numbers. There are a variety of known ways to reduce assigned port number consumption. Although some may be cumbersome or inefficient, they are nearly always preferable to consuming additional port number assignments.

Such techniques include:

- o Use of a discovery service, either a shared service (mDNS), or a discovery service for a given system [RFC6762] [RFC6763].
- o Multiplex packet types using in-band information, either on a per-message or per-connection basis. Such demultiplexing can even hand-off different messages and connections among different processes, such as is done with FTP [RFC959].

There are some cases where NAT and firewall traversal are significantly improved by having an assigned port number. Although

NAT traversal protocols supporting automatic configuration have been proposed and developed (e.g., STUN [RFC5389], TURN [RFC5766], and ICE [RFC5245]), not all application and service designers can rely on their presence as of yet.

In the past, some services were assigned multiple port numbers or sometimes fairly large port ranges (e.g., X11). This occurred for a variety of reasons: port number conservation was not as widely appreciated, assignments were not as ardently reviewed, etc. This no longer reflects current practice and such assignments are not considered to constitute a precedent for future assignments.

### 7.3. Picking an Assigned Port Number

Given a demonstrated need for a port number assignment, the next question is how to pick the desired port number. An application for a port number assignment does not need to include a desired port number; in that case, IANA will select from those currently available.

Users should consider whether the requested port number is important. For example, would an assignment be acceptable if IANA picked the port number value? Would a TCP (or other transport protocol) port number assignment be useful by itself? If so, a port number can be assigned to a service for one transport protocol where it is already (or can be subsequently) assigned to a different service for other transport protocols.

The most critical issue in picking a number is selecting the desired range, i.e., System vs. User port numbers. The distinction was intended to indicate a difference in privilege; originally, System port numbers required privileged ('root') access, while User port numbers did not. That distinction has since blurred because some current systems do not limit access control to System port numbers and because some System services have been replicated on User numbers (e.g., IRC). Even so, System port number assignments have continued at an average rate of 3-4 per year over the past 7 years (2007-2013), indicating that the desire to keep this distinction continues.

As a result, the difference between System and User port numbers needs to be treated with caution. Developers are advised to treat services as if they are always run without privilege.

Even when developers seek a System port number assignment, it may be very difficult to obtain. System port number assignment requires IETF Review or IESG Approval and justification that both User and

Dynamic port number ranges are insufficient [RFC6335]. Thus this document recommends both:

>> Developers SHOULD NOT apply for System port number assignments because the increased privilege they are intended to provide is not always enforced.

>> System implementers SHOULD enforce the need for privilege for processes to listen on System port numbers.

At some future date, it might be useful to deprecate the distinction between System and User port numbers altogether. Services typically require elevated ('root') privileges to bind to a System port number, but many such services go to great lengths to immediately drop those privileges just after connection or other association establishment to reduce the impact of an attack using their capabilities. Such services might be more securely operated on User port numbers than on System port numbers. Further, if System port numbers were no longer assigned, as of 2014 it would cost only 180 of the 1024 System values (17%), or 180 of the overall 49152 assigned (System and User) values (<0.04%).

#### 7.4. Support for Security

Just as a service is a way to obtain information or processing from a host over a network, a service can also be the opening through which to compromise that host. Protecting a service involves security, which includes integrity protection, source authentication, privacy, or any combination of these capabilities. Security can be provided in a number of ways, and thus:

>> New services SHOULD support security capabilities, either directly or via a content protection such as TLS [RFC5246] or DTLS [RFC6347] or transport protection such as TCP-AO [RFC5925]. Insecure versions of new or existing secure services SHOULD be avoided because of the new vulnerability they create.

Secure versions of legacy services that are not already security-capable via in-band negotiations can be very useful. However, there is no IETF consensus on when separate ports should be used for secure and insecure variants of the same service [RFC2595] [RFC2817] [RFC6335]. The overall preference is for use of a single port, as noted in Section 6 of this document and Section 7.2 of [RFC6335], but the appropriate approach depends on the specific characteristics of the service. As a result:

>> When requesting both secure and insecure port assignments for the same service, justification is expected for the utility and safety of each port as an independent service (Section 6). Precedent (e.g., citing other protocols that use a separate insecure port) is inadequate justification by itself.

It's also important to recognize that port number assignment is not itself a guarantee that traffic using that number provides the corresponding service, or that a given service is always offered only on its assigned port number. Port numbers are ultimately meaningful only between endpoints and any service can be run on any port. Thus:

>> Security SHOULD NOT rely on assigned port number distinctions alone; every service, whether secure or not, is likely to be attacked.

Applications for a new service that requires both a secure and insecure port may be found, on expert review, to be unacceptable, and may not be approved for allocation. Similarly, an application for a new port to support an insecure variant of an existing secure protocol may be found unacceptable. In both cases, the resulting security of the service in practice will be a significant consideration in the decision as to whether to assign an insecure port.

#### 7.5. Support for Future Versions

Requests for assigned port numbers are expected to support multiple versions on the same assigned port number [RFC6335]. Versions are typically indicated in-band, either at the beginning of a connection or other association, or in each protocol message.

>> Version support SHOULD be included in new services rather than relying on different port number assignments for different versions.

>> Version numbers SHOULD NOT be included in either the service name or service description, to avoid the need to make additional port number assignments for future variants of a service.

Again, the assigned port number space is far too limited to be used as an indicator of protocol version or message type. Although this has happened in the past (e.g., for NFS), it should be avoided in new requests.

## 7.6. Transport Protocols

IANA assigns port numbers specific to one or more transport protocols, typically UDP [RFC768] and TCP [RFC793], but also SCTP [RFC4960], DCCP [RFC4340], and any other standard transport protocol. Originally, IANA port number assignments were concurrent for both UDP and TCP, and other transports were not indicated. However, to conserve the assigned port number space and to reflect increasing use of other transports, assignments are now specific only to the transport being used.

In general, a service should request assignments for multiple transports using the same service name and description on the same port number only when they all reflect essentially the same service. Good examples of such use are DNS and NFS, where the difference between the UDP and TCP services are specific to supporting each transport. E.g., the UDP variant of a service might add sequence numbers and the TCP variant of the same service might add in-band message delimiters. This document does not describe the appropriate selection of a transport protocol for a service.

>> Service names and descriptions for multiple transport port number assignments SHOULD match only when they describe the same service, excepting only enhancements for each supported transport.

When the services differ, it may be acceptable or preferable to use the same port number, but the service names and descriptions should be different for each transport/service pair, reflecting the differences in the services. E.g., if TCP is used for the basic control protocol and UDP for an alarm protocol, then the services might be "name-ctl" and "name-alarm". A common example is when TCP is used for a service and UDP is used to determine whether that service is active (e.g., via a unicast, broadcast, or multicast test message) [RFC1122]. IANA has, for several years, used the suffix "-disc" in service names to distinguish discovery services, such as are used to identify endpoints capable of a given service:

>> Names of discovery services SHOULD use an identifiable suffix; the suggestion is "-disc".

Some services are used for discovery, either in conjunction with a TCP service or as a stand-alone capability. Such services will be more reliable when using multicast rather than broadcast (over IPv4) because IP routers do not forward "all nodes" broadcasts (all 1's, i.e., 255.255.255.255 for IPv4) and have not been required to support subnet-directed broadcasts since 1999 [RFC1812] [RFC2644].

This issue is relevant only for IPv4 because IPv6 does not support broadcast.

>> UDP over IPv4 multi-host services SHOULD use multicast rather than broadcast.

Designers should be very careful in creating services over transports that do not support congestion control or error recovery, notably UDP. There are several issues that should be considered in such cases, as summarized in Table 1 in [RFC5405]. In addition, the following recommendations apply to service design:

>> Services that use multipoint communication SHOULD be scalable, and SHOULD NOT rely solely on the efficiency of multicast transmission for scalability.

>> Services SHOULD NOT use UDP as a performance enhancement over TCP, e.g., to circumnavigate TCP's congestion control.

#### 7.7. When to Request an Assignment

Assignments are typically requested when a user has enough information to reasonably answer the questions in the IANA application. IANA applications typically take up to a few weeks to process, with some complex cases taking up to a month. The process typically involves a few exchanges between the IANA Ports Expert Review team and the applicant.

An application needs to include a description of the service, as well as to address key questions designed to help IANA determine whether the assignment is justified. The application should be complete and not refer solely to the Internet Draft, RFC, a website, or any other external documentation.

Services that are independently developed can be requested at any time, but are typically best requested in the last stages of design and initial experimentation, before any deployment has occurred that cannot easily be updated.

>> Users MUST NOT deploy implementations that use assigned port numbers prior their assignment by IANA.

>> Users MUST NOT deploy implementations that default to using the experimental System port numbers (1021 and 1022 [RFC4727]) outside a controlled environment where they can be updated with a subsequent assigned port [RFC3692].



Deployments that use unassigned port numbers before assignment complicate IANA management of the port number space. Keep in mind that this recommendation protects existing assignees, users of current services, and applicants for new assignments; it helps ensure that a desired number and service name are available when assigned. The list of currently unassigned numbers is just that - *\*currently\** unassigned. It does not reflect pending applications. Waiting for an official IANA assignment reduces the chance that an assignment request will conflict with another deployed service.

Applications made through Internet Draft / RFC publication (in any stream) typically use a placeholder ("PORTNUM") in the text, and implementations use an experimental port number until a final assignment has been made [RFC6335]. That assignment is initially indicated in the IANA Considerations section of the document, which is tracked by the RFC Editor. When a document has been approved for publication, that request is forwarded to IANA for handling. IANA will make the new assignment accordingly. At that time, IANA may also request that the applicant fill out the application form on their website, e.g., when the RFC does not directly address the information expected as per [RFC6335]. "Early" assignments can be made when justified, e.g., for early interoperability testing, according to existing process [RFC7120] [RFC6335].

>> Users writing specifications SHOULD use symbolic names for port numbers and service names until an IANA assignment has been completed. Implementations SHOULD use experimental port numbers during this time, but those numbers MUST NOT be cited in documentation except as interim.

## 7.8. Squatting

"Squatting" describes the use of a number from the assignable range in deployed software without IANA assignment for that use, regardless of whether the number has been assigned or remains available for assignment. It is hazardous because IANA cannot track such usage and thus cannot avoid making legitimate assignments that conflict with such unauthorized usage.

Such "squatted" port numbers remain unassigned, and IANA retains the right to assign them when requested by other applicants. Application and service designers are reminded that it is never appropriate to use port numbers that have not been directly assigned [RFC6335]. In particular, any unassigned code from the assigned ranges will be assigned by IANA, and any conflict will be easily resolved as the protocol designer's fault once that happens (because they would not be the assignee). This may reflect in the public's judgment on the

quality of their expertise and cooperation with the Internet community.

Regardless, there are numerous services that have squatted on such numbers that are in widespread use. Designers who are using such port numbers are encouraged to apply for an assignment. Note that even widespread de-facto use may not justify a later IANA assignment of that value, especially if either the value has already been assigned to a legitimate applicant or if the service would not qualify for an assignment of its own accord.

#### 7.9. Other Considerations

As noted earlier, System port numbers should be used sparingly, and it is better to avoid them altogether. This avoids the potentially incorrect assumption that the service on such port numbers run in a privileged mode.

Assigned port numbers are not intended to be changed; this includes the corresponding service name. Once deployed, it can be very difficult to recall every implementation, so the assignment should be retained. However, in cases where the current assignee of a name or number has reasonable knowledge of the impact on such uses, and is willing to accept that impact, the name or number of an assignment can be changed [RFC6335]

Aliases, or multiple service names for the same assigned port number, are no longer considered appropriate [RFC6335].

#### 8. Security Considerations

This document focuses on the issues arising when designing services that require new port assignments. Section 7.4 addresses the security and security-related issues of that interaction.

When designing a secure service, the use of TLS [RFC5246], DTLS [RFC6347], or TCP-AO [RFC5925] mechanisms that protect transport protocols or their contents is encouraged. It may not be possible to use IPsec [RFC4301] in similar ways because of the different relationship between IPsec and port numbers and because applications may not be aware of IPsec protections.

This document reminds application and service designers that port numbers do not protect against denial of service attack or guarantee that traffic should be trusted. Using assigned numbers for port filtering isn't a substitute for authentication, encryption, and integrity protection. The port number alone should not be used to

avoid denial of service attacks or to manage firewall traffic because the use of port numbers is not regulated or validated.

The use of assigned port numbers is the antithesis of privacy because they are intended to explicitly indicate the desired application or service. Strictly, port numbers are meaningful only at the endpoints, so any interpretation elsewhere in the network can be arbitrarily incorrect. However, those numbers can also expose information about available services on a given host. This information can be used by intermediate devices to monitor and intercept traffic as well as to potentially identify key endpoint software properties ("fingerprinting"), which can be used to direct other attacks.

## 9. IANA Considerations

The entirety of this document focuses on suggestions that help ensure the conservation of port numbers and provide useful hints for issuing informative requests thereof.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2780] Bradner, S., and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.
- [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3962, Jan. 2004.
- [RFC4727] Fenner, B., "Experimental Values in IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, November 2006.
- [RFC5246] Dierks, T., and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [RFC5405] Eggert, L., and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, Nov. 2008.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

- [RFC6335] Cotton, M., L. Eggert, J. Touch, M. Westerlund, and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, August 2011.
- [RFC6347] Rescorla, E., and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

## 10.2. Informative References

- [IEN112] Postel, J., "Transmission Control Protocol", IEN 112, August 1979.
- [RFC33] Crocker, S., "New Host-Host Protocol", RFC 33 February 1970.
- [RFC37] Crocker, S., "Network Meeting Epilogue", RFC 37, March 1970.
- [RFC38] Wolfe, S., "Comments on Network Protocol from NWG/RFC #36", RFC 38, March 1970.
- [RFC48] Postel, J., and S. Crocker, "Possible protocol plateau", RFC 48, April 1970.
- [RFC61] Walden, D., "Note on Interprocess Communication in a Resource Sharing Computer Network", RFC 61, July 1970.
- [RFC76] Bouknight, J., J. Madden, and G. Grossman, "Connection by name: User oriented protocol", RFC 76, October 1970.
- [RFC333] Bressler, R., D. Murphy, and D. Walden. "Proposed experiment with a Message Switching Protocol", RFC 333, May 1972.
- [RFC739] Postel, J., "Assigned numbers", RFC 739, November 1977.
- [RFC758] Postel, J., "Assigned numbers", RFC 758, August 1979.
- [RFC768] Postel, J., "User Datagram Protocol", RFC 768, August 1980.
- [RFC793] Postel, J., "Transmission Control Protocol" RFC 793, September 1981
- [RFC820] Postel, J., "Assigned numbers", RFC 820, August 1982.

- [RFC900] Reynolds, J., and J. Postel, "Assigned numbers", RFC 900, June 1984.
- [RFC959] Postel, J., and J. Reynolds, "FILE TRANSFER PROTOCOL (FTP)", RFC 959, October 1985.
- [RFC1122] Braden, B. (Ed.), "Requirements for Internet Hosts -- Communication Layers", RFC 1122, October 1989.
- [RFC1340] Reynolds, J., and J. Postel, "Assigned numbers", RFC 1340, July 1992.
- [RFC1700] Reynolds, J., and J. Postel, "Assigned numbers", RFC 1700, October 1994.
- [RFC1812] Baker, F. (Ed.), "Requirements for IP Version 4 Routers", RFC 1812, June 1995.
- [RFC1833] Srinivasan, R., "Binding Protocols for ONC RPC Version 2", RFC 1833, August 1995.
- [RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.
- [RFC2644] Senie, D., "Changing the Default for Directed Broadcasts in Routers", RFC 2644, August 1999.
- [RFC2817] Khare, R., and S. Lawrence, "Upgrading to TLS Within HTTP/1.1", RFC 2817, May 2000.
- [RFC3232] Reynolds, J. (Ed.), "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, January 2002.
- [RFC3261] Rosenberg, J., H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [RFC4301] Kent, S., and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC4340] Kohler, E., M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.
- [RFC4960] Stewart, R. (Ed.), "Stream Control Transmission Protocol", RFC 4960, September 2007.

- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.
- [RFC5389] Rosenberg, J., R. Mahy, P. Matthews, and D. Wing, "Session Traversal Utilities for NAT", RFC 5389, October 2008.
- [RFC5766] Mahy, R., P. Matthews, and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.
- [RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.
- [RFC6762] Cheshire, S., and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.
- [RFC6763] Cheshire, S., and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.
- [RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, January 2014.
- [RFC7230] Fielding, R., (Ed.), and J. Reshke, (Ed.), "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014.

## 11. Acknowledgments

This work benefitted from the feedback from David Black, Lars Eggert, Gorry Fairhurst, and Eliot Lear, as well as discussions of the IETF TSVWG WG.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Joe Touch  
USC/ISI  
4676 Admiralty Way  
Marina del Rey, CA 90292-6695  
U.S.A.

Phone: +1 (310) 448-9151  
EMail: touch@isi.edu





Internet Engineering Task Force  
Internet-Draft  
Intended status: Experimental  
Expires: April 6, 2015

Georgios Karagiannis  
Huawei Technologies  
Anurag Bhargava  
Cisco Systems, Inc.  
October 6, 2014

Generic Aggregation of Resource ReSerVation Protocol (RSVP)  
for IPv4 And IPv6 Reservations over PCN domains  
draft-ietf-tsvwg-rsvp-pcn-11

Abstract

This document specifies extensions to Generic Aggregated RSVP RFC 4860 for support of the PCN Controlled Load (CL) and Single Marking (SM) edge behaviors over a Diffserv cloud using Pre-Congestion Notification.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 6, 2015.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## Table of Contents

1. Introduction . . . . .	4
1.1. Objective . . . . .	4
1.2. Overview and Motivation . . . . .	5
1.3. Terminology . . . . .	7
1.4. Organization of This Document . . . . .	11
2. Overview of RSVP extensions and Operations . . . . .	11
2.1. Overview of RSVP Aggregation Procedures in PCN domains . . . . .	11
2.2. PCN Marking and encoding and transport of pre-congestion Information . . . . .	13
2.3. Traffic Classification Within The Aggregation Region . . . . .	13
2.4. Deaggregator (PCN-egress-node) Determination . . . . .	13
2.5. Mapping E2E Reservations Onto Aggregate Reservations . . . . .	13
2.6. Size of Aggregate Reservations . . . . .	14
2.7. E2E Path ADSPEC update . . . . .	14
2.8. Intra-domain Routes . . . . .	14
2.9. Inter-domain Routes . . . . .	15
2.10. Reservations for Multicast Sessions . . . . .	15
2.11. Multi-level Aggregation . . . . .	15
2.12. Reliability Issues . . . . .	15
3. Elements of Procedure . . . . .	15
3.1. Receipt of E2E Path Message by PCN-ingress-node (aggregating router) . . . . .	15
3.2. Handling Of E2E Path Message by Interior Routers . . . . .	16
3.3. Receipt of E2E Path Message by PCN-egress-node (deaggregating router) . . . . .	16
3.4. Initiation of new Aggregate Path Message By PCN-ingress-node (Aggregating Router) . . . . .	16
3.5. Handling Of new Aggregate Path Message by Interior Routers . . . . .	16
3.6. Handling Of Aggregate Path Message by Deaggregating Router . . . . .	16
3.7. Handling of E2E Resv Message by Deaggregating Router . . . . .	17
3.8. Handling Of E2E Resv Message by Interior Routers . . . . .	17

3.9. Initiation of New Aggregate Resv Message By Deaggregating Router	17
3.10. Handling of Aggregate Resv Message by Interior Routers . . . .	18
3.11. Handling of E2E Resv Message by Aggregating Router . . . . .	18
3.12. Handling of Aggregated Resv Message by Aggregating Router . .	18
3.13. Removal of E2E Reservation . . . . .	19
3.14. Removal of Aggregate Reservation . . . . .	19
3.15. Handling of Data On Reserved E2E Flow by Aggregating Router .	19
3.16. Procedures for Multicast Sessions . . . . .	19
3.17. Misconfiguration of PCN node . . . . .	19
3.18. PCN based Flow Termination . . . . .	19
4. Protocol Elements . . . . .	20
4.1 PCN object . . . . .	20
5. Security Considerations . . . . .	23
6. IANA Considerations . . . . .	24
7. Acknowledgments . . . . .	24
8. Normative References . . . . .	24
9. Informative References . . . . .	25
10. Appendix A: Example Signaling Flow . . . . .	26
11. Authors' Address . . . . .	29

## 1. Introduction

### 1.1 Objective

Pre-Congestion Notification (PCN) can support the quality of service (QoS) of inelastic flows within a Diffserv domain in a simple, scalable, and robust fashion. Two mechanisms are used: admission control and flow termination. Admission control is used to decide whether to admit or block a new flow request, while flow termination is used in abnormal circumstances to decide whether to terminate some of the existing flows. To support these two features, the overall rate of PCN-traffic is metered on every link in the domain, and PCN-packets are appropriately marked when certain configured rates are exceeded. These configured rates are below the rate of the link, thus providing notification to boundary nodes about overloads before any congestion occurs (hence "pre-congestion" notification). The PCN-egress-nodes measure the rates of differently marked PCN traffic in periodic intervals and report these rates to the Decision Points for admission control and flow termination; the Decision Points use these rates to make decisions. The Decision Points may be collocated with the PCN-ingress-nodes, or their function may be implemented in a another node. For more details see [RFC5559], [RFC6661], and [RFC6662].

The main objective of this document is to specify the signaling protocol that can be used within a Pre-Congestion Notification (PCN) domain to carry reports from a PCN-ingress-node to a PCN Decision point, considering that the PCN Decision Point and PCN-egress-node are collocated.

If the PCN Decision Point is not collocated with the PCN-egress-node then additional signaling procedures are required that are out of the scope of this document. Moreover, as mentioned above this architecture conforms with PBAC (Policy-Based Admission Control), when the Decision Point is located in a another node then the PCN-ingress-node [RFC2753].

Several signaling protocols can be used to carry information between PCN-boundary-nodes (PCN-ingress-node and PCN-egress-node). However, since (1) both PCN-egress-node and PCN-ingress-nodes are located on the data path and (2) the admission control procedure needs to be done at PCN-egress-node, a signaling protocol that follows the same path as the data path, like RSVP (Resource Reservation Protocol), is more suited for this purpose. In particular, this document specifies extensions to Generic Aggregated RSVP [RFC4860] for support of the PCN Controlled Load (CL) and Single Marking (SM) edge behaviors over a Diffserv cloud using Pre-Congestion Notification.

This draft is intended to be published as Experimental in order to:

- o) validate industry interest by allowing implementation and deployment
- o) gather operational experience, in particular around dynamic interactions of RSVP signaling and PCN notification and

corresponding levels of performance.

Support for the techniques specified in this document involves RSVP functionality in boundary nodes of a PCN domain whose interior nodes forward RSVP traffic without performing RSVP functionality.

## 1.2 Overview and Motivation

Two main Quality of Service (QoS) architectures have been specified by the IETF. These are the Integrated Services (Intserv) [RFC1633] architecture and the Differentiated Services (DiffServ) architecture ([RFC2475]).

Intserv provides methods for the delivery of end-to-end Quality of Service (QoS) to applications over heterogeneous networks. One of the QoS signaling protocols used by the Intserv architecture is the Resource reSerVation Protocol (RSVP) [RFC2205], which can be used by applications to request per-flow resources from the network. These RSVP requests can be admitted or rejected by the network. Applications can express their quantifiable resource requirements using Intserv parameters as defined in [RFC2211] and [RFC2212]. The Controlled Load (CL) service [RFC2211] is a quality of service (QoS) closely approximating the QoS that the same flow would receive from a lightly loaded network element. The CL service is useful for inelastic flows such as those used for real-time media.

The DiffServ architecture can support the differentiated treatment of packets in very large scale environments. While Intserv and RSVP classify packets per-flow, Diffserv networks classify packets into one of a small number of aggregated flows or "classes", based on the Diffserv codepoint (DSCP) in the packet IP header. At each Diffserv router, packets are subjected to a "per-hop behavior" (PHB), which is invoked by the DSCP. The primary benefit of Diffserv is its scalability, since the need for per-flow state and per-flow processing, is eliminated.

However, DiffServ does not include any mechanism for communication between applications and the network. Several solutions have been specified to solve this issue. One of these solutions is Intserv over Diffserv [RFC2998] including resource-based admission control (RBAC), PBAC, assistance in traffic identification/classification, and traffic conditioning. Intserv over Diffserv can operate over a statically provisioned or a RSVP aware Diffserv region. When it is RSVP aware, several mechanisms may be used to support dynamic provisioning and topology-aware admission control, including aggregate RSVP reservations, per-flow RSVP, or a bandwidth broker. [RFC3175] specifies aggregation of Resource ReSerVation Protocol (RSVP) end-to-end reservations over aggregate RSVP reservations. In [RFC3175] the RSVP generic aggregated reservation is characterized by a RSVP SESSION object using the 3-tuple <source IP address, destination IP address, Diffserv Code Point>.

Several scenarios require the use of multiple generic aggregate reservations that are established for a given PHB from a given source

IP address to a given destination IP address, see [SIG-NESTED], [RFC4860]. For example, multiple generic aggregate reservations can be applied in the situation that multiple E2E reservations using different preemption priorities need to be aggregated through a PCN-domain using the same PHB. By using multiple aggregate reservations for the same PHB, it allows enforcement of the different preemption priorities within the aggregation region. This allows more efficient management of the Diffserv resources, and in periods of resource shortage, this allows sustainment of a larger number of E2E reservations with higher preemption priorities. In particular, [SIG-NESTED] discusses in detail how end-to-end RSVP reservations can be established in a nested VPN environment through RSVP aggregation.

[RFC4860] provides generic aggregate reservations by extending [RFC3175] to support multiple aggregate reservations for the same source IP address, destination IP address, and PHB (or set of PHBs). In particular, multiple such generic aggregate reservations can be established for a given PHB from a given source IP address to a given destination IP address. This is achieved by adding the concept of a Virtual Destination Port and of an Extended Virtual Destination Port in the RSVP SESSION object. In addition to this, the RSVP SESSION object for generic aggregate reservations uses the PHB Identification Code (PHB-ID) defined in [RFC3140], instead of using the Diffserv Code Point (DSCP) used in [RFC3175]. The PHB-ID is used to identify the PHB, or set of PHBs, from which the Diffserv resources are to be reserved.

The RSVP like signaling protocol required to carry (1) requests from a PCN-egress-node to a PCN-ingress-node and (2) reports from a PCN-ingress-node to a PCN-egress-node needs to follow the PCN signaling requirements defined in [RFC6663]. In addition to that the signaling protocol functionality supported by the PCN-ingress-nodes and PCN-egress-nodes needs to maintain logical aggregate constructs (i.e. ingress-egress-aggregate state) and be able to map E2E reservations to these aggregate constructs. Moreover, no actual reservation state is needed to be maintained inside the PCN domain, i.e., the PCN-interior-nodes are not maintaining any reservation state.

This can be accomplished by two possible approaches:

Approach (1):

- o) adapting the RFC 4860 aggregation procedures to fit the PCN requirements with as little change as possible over the RFC 4860 functionality
- o) hence performing aggregate RSVP signaling (even if it is to be ignored by PCN interior nodes)
- o) using this aggregate RSVP signaling procedures to carry PCN information between the PCN-boundary-nodes (PCN-ingress-node and PCN-egress-node).

## Approach (2):

- o) adapting the RFC 4860 aggregation procedures to fit the PCN requirements with more significant changes over RFC4860 (i.e. the aspect of the procedures that have to do with maintaining aggregate states and to do with mapping the E2E reservations to aggregate constructs are kept, but the procedures that have to do with the aggregate RSVP signaling and aggregate reservation establishment/maintenance are dropped).
- o) hence not performing aggregate RSVP signaling
- o) piggy-backing of the PCN information inside the E2E RSVP signaling.

Both approaches are probably viable, however, since the RFC 4860 operations have been thoroughly studied and implemented, it can be considered that the RFC 4860 solution can better deal with the more challenging situations (rerouting in the PCN domain, failure of an PCN-ingress-node, failure of an PCN-egress-node, rerouting towards a different edge, etc.). This is the reason for choosing Approach (1) for the specification of the signaling protocol used to carry PCN information between the PCN-boundary-nodes (PCN-ingress-node and PCN-egress-node).

In particular, this document specifies extensions to Generic Aggregated RSVP [RFC4860] for support of the PCN Controlled Load (CL) and Single Marking (SM) edge behaviors over a Diffserv cloud using Pre-Congestion Notification.

This document follows the PCN signaling requirements defined in [RFC6663] and specifies extensions to Generic Aggregated RSVP [RFC4860] for support of PCN edge behaviors as specified in [RFC6661] and [RFC6662]. Moreover, this document specifies how RSVP aggregation can be used to setup and maintain: (1) Ingress Egress Aggregate (IEA) states at Ingress and Egress nodes and (2) generic aggregation of RSVP end-to-end RSVP reservations over PCN (Congestion and Pre-Congestion Notification) domains.

To comply with this specification, PCN-nodes MUST be able to support the functionality specified in [RFC5670], [RFC5559], [RFC6660], [RFC6661], [RFC6662]. Furthermore, the PCN-boundary-nodes MUST support the RSVP generic aggregated reservation procedures specified in [RFC4860] which are augmented with procedures specified in this document.

### 1.3. Terminology

This document uses terms defined in [RFC4860], [RFC3175], [RFC5559], [RFC5670], [RFC6661], [RFC6662].

For readability, a number of definitions from [RFC3175] as well as definitions for terms used in [RFC5559], [RFC6661], and [RFC6662] are provided here, where some of them are augmented with new meanings:

Aggregator	This is the process in (or associated with) the router at the ingress edge of the aggregation region (with respect to the end-to-end RSVP reservation) and behaving in accordance with [RFC4860]. In this document, it is also the PCN-ingress-node. It is important to notice that in the context of this document the Aggregator must be able to determine the Deaggregator using the procedures specified in Section 4 of [RFC4860] and in Section 1.4.2 of [RFC3175].
Congestion level estimate (CLE):	<p>The ratio of PCN-marked to total PCN-traffic (measured in octets) received for a given ingress-egress-aggregate during a given measurement period. The CLE is used to derive the PCN-admission-state and is also used by the report suppression procedure if report suppression is activated.</p>
Deaggregator	This is the process in (or associated with) the router at the egress edge of the aggregation region (with respect to the end-to-end RSVP reservation) and behaving in accordance with [RFC4860]. In this document, it is also the PCN-egress-node and Decision Point.
E2E	end to end
E2E Reservation	<p>This is an RSVP reservation such that:</p> <ul style="list-style-type: none"><li>(i) corresponding RSVP Path messages are initiated upstream of the Aggregator and terminated downstream of the Deaggregator, and</li><li>(ii) corresponding RSVP Resv messages are initiated downstream of the Deaggregator and terminated upstream of the Aggregator, and</li><li>(iii) this RSVP reservation is aggregated over an Ingress Egress Aggregate (IEA) between the Aggregator and Deaggregator.</li></ul> <p>An E2E RSVP reservation may be a per-flow reservation, which in this document is only maintained at the PCN-ingress-node and PCN-egress-node. Alternatively, the E2E reservation may itself be an aggregate reservation of various types (e.g., Aggregate IP reservation, Aggregate IPsec reservation, see [RFC4860]). As per regular RSVP operations, E2E RSVP reservations are unidirectional.</p>
E2E microflow	a microflow where its associated packets are being forwarded on an E2E path.



## Extended vDstPort (Extended Virtual Destination Port)

An identifier used in the SESSION that remains constant over the life of the generic aggregate reservation. The length of this identifier is 32-bits when IPv4 addresses are used and 128 bits when IPv6 addresses are used.

A sender(or Aggregator) that wishes to narrow the scope of a SESSION to the sender-receiver pair (or Aggregator-Deaggregator pair) should place its IPv4 or IPv6 address here as a network unique identifier. A sender (or Aggregator) that wishes to use a common session with other senders (or Aggregators) in order to use a shared reservation across senders (or Aggregators) must set this field to all zeros. In this document, the Extended vDstPort should contain the IPv4 or IPv6 address of the Aggregator.

## ETM-rate

The rate of excess-traffic-marked PCN-traffic received at a PCN-egress-node for a given ingress-egress-aggregate in octets per second.

## Ingress-egress-aggregate (IEA):

The collection of PCN-packets from all PCN-flows that travel in one direction between a specific pair of PCN-boundary-nodes. In this document one RSVP generic aggregated reservation is mapped to only one ingress-egress-aggregate, while one ingress-egress-aggregate is mapped to either one or to more than one RSVP generic aggregated reservations. PCN-flows and their PCN-traffic that are mapped into a specific RSVP generic aggregated reservation can also easily be mapped into their corresponding ingress-egress-aggregate.

Microflow:  
(from [RFC2474])

a single instance of an application-to-application flow of packets which is identified by source address, destination address, protocol id, and source port, destination port (where applicable).

## PCN-domain:

a PCN-capable domain; a contiguous set of PCN-enabled nodes that perform Diffserv scheduling [RFC2474]; the complete set of PCN-nodes that in principle can, through PCN-marking packets, influence decisions about flow admission and termination within the domain; includes the PCN-egress-nodes, which measure these PCN-marks, and the PCN-ingress-nodes.

PCN-boundary-node: a PCN-node that connects one PCN-domain to a node either in another PCN-domain or in a non-PCN-domain.

- PCN-interior-node: a node in a PCN-domain that is not a PCN-boundary-node.
- PCN-node: a PCN-boundary-node or a PCN-interior-node.
- PCN-egress-node: a PCN-boundary-node in its role in handling traffic as it leaves a PCN-domain. In this document the PCN-egress-node operates also as a Decision Point and Deaggregator.
- PCN-ingress-node: a PCN-boundary-node in its role in handling traffic as it enters a PCN-domain. In this document the PCN-ingress-node operates also as a Aggregator.
- PCN-traffic,  
PCN-packets,  
PCN-BA: a PCN-domain carries traffic of different Diffserv behavior aggregates (BAs) [RFC2474]. The PCN-BA uses the PCN mechanisms to carry PCN-traffic, and the corresponding packets are PCN-packets. The same network will carry traffic of other Diffserv BAs. The PCN-BA is distinguished by a combination of the Diffserv codepoint (DSCP) and ECN fields.
- PCN-flow: the unit of PCN-traffic that the PCN-boundary-node admits (or terminates); the unit could be a single E2E microflow (as defined in [RFC2474]) or some identifiable collection of microflows.
- PCN-admission-state: The state ("admit" or "block") derived by the Decision Point for a given ingress-egress-aggregate based on statistics about PCN-packet marking. The Decision Point decides to admit or block new flows offered to the aggregate based on the current value of the PCN-admission-state.
- PCN-sent-rate The rate of PCN-traffic received at a PCN-ingress-node and destined for a given ingress-egress-aggregate in octets per second.
- PHB-ID (Per Hop Behavior Identification Code)  
A 16-bit field containing the Per Hop Behavior Identification Code of the PHB, or of the set of PHBs, from which Diffserv resources are to be reserved. This field must be encoded as specified in Section 2 of [RFC3140].
- RSVP generic aggregated reservation: an RSVP reservation that is identified by using the RSVP SESSION object for generic RSVP aggregated reservation. This RSVP

SESSION object is based on the RSVP SESSION object specified in [RFC4860] augmented with the following information:

- o) the IPv4 DestAddress, IPv6 DestAddress should be set to the IPv4 or IPv6 destination addresses, respectively, of the Deaggregator (PCN-egress-node)
- o) PHB-ID (Per Hop Behavior Identification Code) should be set equal to PCN-compatible Diffserv codepoint(s).
- o) Extended vDstPort should be set to the IPv4 or IPv6 destination addresses, of the Aggregator (PCN-ingress-node)

VDstPort (Virtual Destination Port)

A 16-bit identifier used in the SESSION that remains constant over the life of the generic aggregate reservation.

#### 1.4. Organization of This Document

This document is organized as follows. Section 2 gives an overview of RSVP extensions and operations. The elements of the used procedures are specified in Section 3. Section 4 describes the protocol elements. The security considerations are given in section 5 and the IANA considerations are provided in Section 6.

## 2. Overview of RSVP extensions and Operations

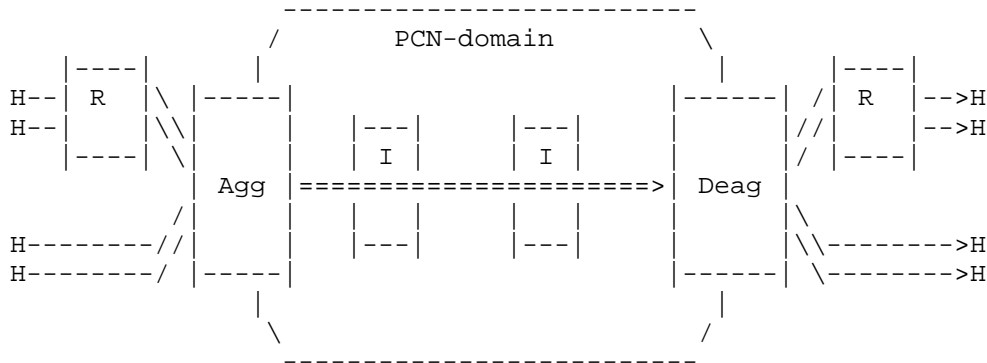
### 2.1 Overview of RSVP Aggregation Procedures in PCN domains

The PCN-boundary-nodes, see Figure 1, can support RSVP SESSIONS for generic aggregated reservations [RFC4860], which are depending on ingress-egress-aggregates. In particular, one RSVP generic aggregated reservation matches to only one ingress-egress-aggregate.

However, one ingress-egress-aggregate matches to either one, or more than one, RSVP generic aggregated reservations. In addition, to comply with this specification, the PCN-boundary nodes need to distinguish and process (1) RSVP SESSIONS for generic aggregated sessions and their messages according to [RFC4860], (2) E2E RSVP sessions and messages according to [RFC2205].

This document locates all RSVP processing for a PCN domain at PCN-Boundary nodes. PCN-interior-nodes do not perform any RSVP functionality or maintain RSVP-related state information. Rather, PCN-interior nodes forward all RSVP messages (for both generic aggregated reservations[RFC4860] and end to end reservations [RFC2205]) as if they were ordinary network traffic.

Moreover, each Aggregator and Deaggregator (i.e., PCN-boundary-nodes) need to support policies to initiate and maintain for each pair of PCN-boundary-nodes of the same PCN-domain one ingress-egress-aggregate.



H = Host requesting end-to-end RSVP reservations  
 R = RSVP router  
 Agg = Aggregator (PCN-ingress-node)  
 Deag = Deaggregator (PCN-egress-node)  
 I = Interior Router (PCN-interior-node)  
 --> = E2E RSVP reservation  
 ==> = Aggregate RSVP reservation

Figure 1 : Aggregation of E2E Reservations  
 over Generic Aggregate RSVP Reservations  
 in PCN domains, based on [RFC4860]

Both the Aggregator and Deaggregator can maintain one or more RSVP generic aggregated Reservations, but the Deaggregator is the entity that initiates these RSVP generic aggregated reservations. Note that one RSVP generic aggregated reservation matches to only one ingress-egress-aggregate, while one ingress-egress-aggregate matches to either one or to more than one RSVP generic aggregated reservations. This can be accomplished by using for the different RSVP generic aggregated reservations the same combinations of ingress and egress identifiers, but with a different PHB-ID value (see [RFC4860]). The procedures for aggregation of E2E reservations over generic aggregate RSVP reservations are the same as the procedures specified in Section 4 of [RFC4860], augmented with the ones specified in Section 2.5.

One significant difference between this document and [RFC4860] is the fact that in this document the admission control of E2E RSVP reservations over the PCN core is performed according to the PCN procedures, while in [RFC4860] this is achieved via first admitting aggregate RSVP reservations over the aggregation region and then admitting the E2E reservations over the aggregate RSVP reservations. Therefore, in this document, the RSVP generic aggregate RSVP reservations are not subject to admission control in the PCN-core, and the E2E RSVP reservations are not subject to admission control

over the aggregate reservations. In turn, this means that several procedures of [RFC4860] are significantly simplified in this document:

- o) unlike [RFC4860], the generic aggregate RSVP reservations need not be admitted in the PCN core.
- o) unlike [RFC4860], the RSVP aggregated traffic does not need to be tunneled between Aggregator and Deaggregator, see Section 2.3.
- o) unlike [RFC4860], the Deaggregator need not perform admission control of E2E reservations over the aggregate RSVP reservations.
- o) unlike [RFC4860], there is no need for dynamic adjustment of the RSVP generic aggregated reservation size, see Section 2.6.

## 2.2 PCN Marking and encoding and transport of pre-congestion information

The method of PCN marking within the PCN domain is specified in [RFC5670]. In addition, the method of encoding and transport of pre-congestion information is specified in [RFC6660]. The PHB-ID (Per Hop Behavior Identification Code) used SHOULD be set equal to PCN-compatible Diffserv codepoint(s).

## 2.3. Traffic Classification Within The Aggregation Region

The PCN-ingress marks a PCN-BA using PCN-marking (i.e., combination of the DSCP and ECN fields), which interior nodes use to classify PCN-traffic. The PCN-traffic (e.g., E2E microflows) belonging to a RSVP generic aggregated reservation can be classified only at the PCN-boundary-nodes (i.e., Aggregator and Deaggregator) by using the RSVP SESSION object for RSVP generic aggregated reservations, see Section 2.1 of [RFC4860]. Note that the DSCP value included in the SESSION object, SHOULD be set equal to a PCN-compatible Diffserv codepoint. Since no admission control procedures over the RSVP generic aggregated reservations in the PCN-core are required, unlike [RFC4860], the RSVP aggregated traffic need not to be tunneled between Aggregator and Deaggregator. In this document one RSVP generic aggregated reservation is mapped to only one ingress-egress-aggregate, while one ingress-egress-aggregate is mapped to either one or to more than one RSVP generic aggregated reservations. PCN-flows and their PCN-traffic that are mapped into a specific RSVP generic aggregated reservation can also easily be classified into their corresponding ingress-egress-aggregate. The method of traffic conditioning of PCN-traffic and non-PCN traffic and PHB configuration is described in [RFC6661] and [RFC6662].

## 2.4. Deaggregator Determination

The present document assumes the same dynamic Deaggregator determination method as used in [RFC4860].

## 2.5. Mapping E2E Reservations Onto Aggregate Reservations

To comply with this specification for the mapping of E2E reservations

onto aggregate reservations, the same methods MUST be used as the ones described in Section 4 of [RFC4860], augmented by the following rules:

- o) An Aggregator (also PCN-ingress-node in this document) or Deaggregator (also PCN-egress-node and Decision Point in this document) MUST use one or more policies to determine whether a RSVP generic aggregated reservation can be mapped into an ingress-Egress-aggregate. This can be accomplished by using for the different RSVP generic aggregated reservations the same combinations of ingress and egress identifiers, but with a different PHB-ID value (see [RFC4860]) corresponding to the PCN specifications. In particular, the RSVP SESSION object specified in [RFC4860] augmented with the following information:
  - o) the IPv4 DestAddress, IPv6 DestAddress MUST be set to the IPv4 or IPv6 destination addresses, respectively, of the Deaggregator (PCN-egress-node), see [RFC4860]. Note that the PCN-domain is considered as being only one RSVP hop (for Generic aggregated RSVP or E2E RSVP). This means that the next RSVP hop for the Aggregator in the downstream direction is the Deaggregator and the next RSVP hop for the Deaggregator in the upstream direction is the Aggregator.
  - o) PHB-ID (Per Hop Behavior Identification Code) SHOULD be set equal to PCN-compatible Diffserv codepoint(s).
  - o) Extended vDstPort SHOULD be set to the IPv4 or IPv6 destination addresses, of the Aggregator (PCN-ingress-node), see [RFC4860].

## 2.6. Size of Aggregate Reservations

Since:(i) no admission control of E2 reservations over the RSVP aggregated reservations is required, and (ii) no admission control of the RSVP aggregated reservation over the PCN core is required, the size of the generic aggregate reservation is irrelevant and can be set to any arbitrary value by the Deaggregator. The Deaggregator SHOULD set the value of a generic aggregate reservation to a null bandwidth. We also observe that there is no need for dynamic adjustment of the RSVP aggregated reservation size.

## 2.7. E2E Path ADSPEC update

To comply with this specification, for the update of the E2E Path ADSPEC, the same methods can be used as the ones described in [RFC4860].

## 2.8. Intra-domain Routes

The PCN-interior-nodes are neither maintaining E2E RSVP nor RSVP generic aggregation states and reservations. Therefore, intra-domain route changes will not affect intra-domain reservations since such reservations are not maintained by the PCN-interior-nodes.

Furthermore, it is considered that by configuration, the PCN-interior-nodes are not able to distinguish neither RSVP generic aggregated sessions and their associated messages [RFC4860], nor E2E RSVP sessions and their associated messages [RFC2205].

## 2.9. Inter-domain Routes

The PCN-charter scope precludes inter-domain considerations. However, for solving inter-domain routes changes associated with the operation of the RSVP messages, the same methods SHOULD be used as the ones described in [RFC4860] and in Section 1.4.7 of [RFC3175].

## 2.10. Reservations for Multicast Sessions

PCN does not consider reservations for multicast sessions.

## 2.11. Multi-level Aggregation

PCN does not consider multi-level aggregations within the PCN domain. Therefore, the PCN-interior-nodes are not supporting multi-level aggregation procedures. However, the Aggregator and Deaggregator SHOULD support the multi-level aggregation procedures specified in [RFC4860] and in Section 1.4.9 of [RFC3175].

## 2.12. Reliability Issues

To comply with this specification, for solving possible reliability issues, the same methods MUST be used as the ones described in Section 4 of [RFC4860].

## 3. Elements of Procedure

This section describes the procedures used to implement the aggregated RSVP procedure over PCN. It is considered that the procedures for aggregation of E2E reservations over generic aggregate RSVP reservations are same as the procedures specified in Section 4 of [RFC4860] except where a departure from these procedures is explicitly described in the present section. Please refer to [RFC4860] for all the below error cases:

- o) Incomplete message
- o) Unexpected objects

### 3.1. Receipt of E2E Path Message by Aggregating router

When the E2E Path message arrives at the exterior interface of the Aggregator, (also PCN-ingress-node in this document), then standard RSVP generic aggregation [RFC4860] procedures are used.

### 3.2. Handling Of E2E Path Message by Interior Routers

The E2E Path messages traverse zero or more PCN-interior-nodes. The PCN-interior-nodes receive the E2E Path message on an interior interface and forward it on another interior interface. It is considered that, by configuration, the PCN-interior-nodes ignore the E2E RSVP signaling messages [RFC2205]. Therefore, the E2E Path messages are simply forwarded as normal IP datagrams.

### 3.3. Receipt of E2E Path Message by Deaggregating router

When receiving the E2E Path message the Deaggregator (also PCN-egress-node and Decision Point in this document) performs the regular [RFC4860] procedures, augmented with the following rules:

- o) The Deaggregator MUST NOT perform the RSVP-TTL vs IP TTL-check and MUST NOT update the ADspec Break bit. This is because the whole PCN-domain is effectively handled by E2E RSVP as a virtual link on which integrated service is indeed supported (and admission control performed) so that the Break bit MUST NOT be set, see also [draft-lefaucheur-rsvp-ecn-01].

The Deaggregator forwards the E2E Path message towards the receiver.

### 3.4. Initiation of new Aggregate Path Message by Aggregating Router

To comply with this specification, for the initiation of the new RSVP generic aggregated Path message by the Aggregator (also PCN-ingress-node in this document), the same methods MUST be used as the ones described in [RFC4860].

### 3.5. Handling Of Aggregate Path Message By Interior Routers

The Aggregate Path messages traverse zero or more PCN-interior-nodes. The PCN-interior-nodes receive the Aggregated Path message on an interior interface and forward it on another interior interface. It is considered that, by configuration, the PCN-interior-nodes ignore the Aggregated Path signaling messages. Therefore, the Aggregated Path messages are simply forwarded as normal IP datagrams.

### 3.6. Handling Of Aggregate Path Message By Deaggregating Router

When receiving the Aggregated Path message, the Deaggregator (also PCN-egress-node and Decision Point in this document) performs the regular [RFC4860] procedures, augmented with the following rules:

- o) When the received Aggregated Path message by the Deaggregator contains the RSVP-AGGREGATE-IPv4-PCN-response or RSVP-AGGREGATE-IPv6-PCN-response PCN objects, which carry the PCN-sent-rate, then the procedures specified in Section 3.18 of this document MUST be followed.



### 3.7. Handling of E2E Resv Message by Deaggregating Router

When the E2E Resv message arrives at the exterior interface of the Deaggregator, (also PCN-egress-node and Decision Point in this document) then standard RSVP aggregation [RFC4860] procedures are used, augmented with the following rules:

- o) The E2E RSVP session associated with an E2E Resv message that arrives at the external interface of the Deaggregator is mapped/matched with an RSVP generic aggregate and with a PCN ingress-egress-aggregate.
- o) Depending on the type of the PCN edge behavior supported by the Deaggregator, the PCN admission control procedures specified in Section 3.3.1 of [RFC6661] or [RFC6662] MUST be followed. Since no admission control procedures over the RSVP aggregated reservations in the PCN-core are required, unlike [RFC4860], the Deaggregator does not perform any admission control of the E2E Reservation over the mapped generic aggregate RSVP reservation. If the PCN based admission control procedure is successful then the Deaggregator MUST allow the new flow to be admitted onto the associated RSVP generic aggregation reservation and onto the PCN ingress-egress-aggregate, see [RFC6661] and [RFC6662]. If the PCN based admission control procedure is not successful, then the E2E Resv MUST NOT be admitted onto the associated RSVP generic aggregate reservation and onto the PCN ingress-egress-aggregation. The E2E Resv message is further processed according to [RFC4860].

The way of how the PCN-admission-state is maintained is specified in [RFC6661] and [RFC6662].

### 3.8. Handling Of E2E Resv Message By Interior Routers

The E2E Resv messages traversing the PCN core are IP addressed to the Aggregating router and are not marked with Router Alert, therefore the E2E Resv messages are simply forwarded as normal IP datagrams.

### 3.9. Initiation of New Aggregate Resv Message By Deaggregating Router

To comply with this specification, for the initiation of the new RSVP generic aggregated Resv message by the Deaggregator (also PCN-egress-node and Decision Point in this document), the same methods MUST be used as the ones described in Section 4 of [RFC4860] augmented with the following rules:

- o) The size of the generic aggregate reservation is irrelevant, see Section 2.6, and can be set to any arbitrary value by the PCN-egress node. The Deaggregator SHOULD set the value of a RSVP generic aggregate reservation to a null bandwidth. We also observe that there is no need for dynamic adjustment of the RSVP generic aggregated reservation size.

- o) When [RFC6661] is used and the ETM-rate measured by the Deaggregator contains a non-zero value for some ingress-egress-aggregate, see [RFC6661] and [RFC6662], the Deaggregator MUST request the PCN-ingress-node to provide an estimate of the rate (PCN-sent-rate) at which the Aggregator (also PCN-ingress-node in this document) is receiving PCN-traffic that is destined for the given ingress-egress-aggregate.
- o) When [RFC6662] is used and the PCN-admission-state computed by the Deaggregator, on the basis of the CLE is "block" for the given ingress-egress-aggregate, the Deaggregator MUST request the PCN-ingress-node to provide an estimate of the rate (PCN-sent-rate) at which the Aggregator is receiving PCN-traffic that is destined for the given ingress-egress-aggregate.
- o) In the above two cases and when the PCN-sent-rate needs to be requested from the Aggregator, the Deaggregator MUST generate and send an (refresh) Aggregated Resv message to the Aggregator that MUST carry one of the following PCN objects, see Section 4.1, depending on whether IPv4 or IPv6 is supported:
  - o) RSVP-AGGREGATE-IPv4-PCN-request
  - o) RSVP-AGGREGATE-IPv6-PCN-request.

### 3.10. Handling of Aggregate Resv Message by Interior Routers

The Aggregated Resv messages traversing the PCN core are IP addressed to the Aggregating router and are not marked with Router Alert, therefore the Aggregated Resv messages are simply forwarded as normal IP datagrams.

### 3.11. Handling of E2E Resv Message by Aggregating Router

When the E2E Resv message arrives at the interior interface of the Aggregator (also PCN-ingress-node in this document), then standard RSVP aggregation [RFC4860] procedures are used.

### 3.12. Handling of Aggregated Resv Message by Aggregating Router

When the Aggregated Resv message arrives at the interior interface of the Aggregator, (also PCN-ingress-node in this document), then standard RSVP aggregation [RFC4860] procedures are used, augmented with the following rules:

- o) the Aggregator SHOULD use the information carried by the PCN objects, see Section 4, and follow the steps specified in [RFC6661], [RFC6662]. If the "R" flag carried by the RSVP-AGGREGATE-IPv4-PCN-request or RSVP-AGGREGATE-IPv6-PCN-request PCN objects is set to ON, see Section 4.1, then the Aggregator follows the steps described in Section 3.4 of [RFC6661] and [RFC6662] on calculating the PCN-sent-rate. In particular, the Aggregator MUST provide the estimated current rate of PCN-traffic received at that node and destined for a given ingress-egress-aggregate in octets per second (the PCN-sent-rate). The way this rate estimate is derived is a matter of implementation, see [RFC6661] or [RFC6662].

- o) the Aggregator initiates an Aggregated Path message. In particular, when the Aggregator receives an Aggregated Resv message which carries one of the following PCN objects: RSVP-AGGREGATE-IPv4-PCN-request or RSVP-AGGREGATE-IPv6-PCN-request, with the flag "R" set to ON, see Section 4.1, the Aggregator initiates an Aggregated Path message, and includes the calculated PCN-sent-rate into the RSVP-AGGREGATE-IPv4-PCN-response or RSVP-AGGREGATE-IPv6-PCN-response PCN objects, see Section 4.1, which that MUST be carried by the Aggregated Path message. This Aggregated Path message is sent towards the Deaggregator (also PCN-egress-node and Decision Point in this document) that requested the calculation of the PCN-sent-rate.

### 3.13. Removal of E2E Reservation

To comply with this specification, for the removal of E2E reservations, the same methods MUST be used as the ones described in Section 4 of [RFC4860] and [RFC4495].

### 3.14. Removal of Aggregate Reservation

To comply with this specification, for the removal of RSVP generic aggregated reservations, the same methods MUST be used as the ones described in Section 4 of [RFC4860] and Section 2.10 of [RFC3175]. In particular, should an aggregate reservation go away (presumably due to a configuration change, route change, or policy event), the E2E reservations it supports are no longer active. They MUST be treated accordingly.

### 3.15. Handling of Data On Reserved E2E Flow by Aggregating Router

The handling of data on the reserved E2E flow by Aggregator (also PCN-ingress-node in this document) uses the procedures described in [RFC4860] augmented with:

- o) Regarding, PCN marking and traffic classification the procedures defined in Section 2.2 and 2.3 of this document are used.

### 3.16. Procedures for Multicast Sessions

In this document no multicast sessions are considered.

### 3.17. Misconfiguration of PCN-node

In an event where a PCN-node is misconfigured within a PCN-domain, the desired behavior is same as described in Section 3.10.

### 3.18 PCN based Flow Termination

When the Deaggregator (also PCN-egress-node and Decision Point in this document) needs to terminate an amount of traffic associated with one ingress-egress-aggregate (see Section 3.3.2 of [RFC6661] and [RFC6662]), then several procedures of terminating E2E microflows can be deployed. The default procedure of terminating E2E microflows (i.e., PCN-flows) is as follows, see i.e., [RFC6661] and [RFC6662].

For the same ingress-egress-aggregate, select a number of E2E microflows to be terminated in order to decrease the total incoming amount of bandwidth associated with one ingress-egress-aggregate by the amount of traffic to be terminated, see above. In this situation the same mechanisms for terminating an E2E microflow can be followed as specified in [RFC2205]. However, based on a local policy, the Deaggregator could use other ways of selecting which microflows should be terminated. For example, for the same ingress-egress-aggregate, select a number of E2E microflows to be terminated or to reduce their reserved bandwidth in order to decrease the total incoming amount of bandwidth associated with one ingress-egress-aggregate by the amount of traffic to be terminated. In this situation the same mechanisms for terminating an E2E microflow or reducing bandwidth associated with an E2E microflow can be followed as specified in [RFC4495].

#### 4. Protocol Elements

The protocol elements in this document are using the ones defined in Section 4 of [RFC4860] and Section 3 of [RFC3175] augmented with the following rules:

- o) the DSCP value included in the SESSION object, SHOULD be set equal to a PCN-compatible Diffserv codepoint.
- o) Extended vDstPort SHOULD be set to the IPv4 or IPv6 destination addresses, of the Aggregator (also PCN-ingress-node in this document), see [RFC4860].
- o) When the Deaggregator (also PCN-egress-node and Decision Point in this document) needs to request the PCN-sent-rate from the PCN-ingress-node, see Section 3.9 of this document, the Deaggregator MUST generate and send an (refresh) Aggregate Resv message to the Aggregator that MUST carry one of the following PCN objects, see Section 4.1, depending on whether IPv4 or IPv6 is supported:
  - o) RSVP-AGGREGATE-IPv4-PCN-request
  - o) RSVP-AGGREGATE-IPv6-PCN-request.
- o) When the Aggregator receives an Aggregate Resv message which carries one of the following PCN objects:  
RSVP-AGGREGATE-IPv4-PCN-request or  
RSVP-AGGREGATE-IPv6-PCN-request, with the flag "R" set to ON, see Section 4.1, then the Aggregator MUST generate and send to the Deaggregator an Aggregated Path message which carries one of the following PCN objects, see Section 4.1, depending on whether IPv4 or IPv6 is supported:
  - o) RSVP-AGGREGATE-IPv4-PCN-response,
  - o) RSVP-AGGREGATE-IPv6-PCN-response.

##### 4.1 PCN objects

This section describes four types of PCN objects that can be carried by the (refresh) Aggregate Path or the (refresh) Aggregate Resv messages specified in [RFC4860].

These objects are:

- o RSVP-AGGREGATE-IPv4-PCN-request,
- o RSVP-AGGREGATE-IPv6-PCN-request,
- o RSVP-AGGREGATE-IPv4-PCN-response,
- o RSVP-AGGREGATE-IPv6-PCN-response.

- o) RSVP-AGGREGATE-IPv4-PCN-request: PCN request object, when IPv4 addresses are used:

Class = 248 (PCN)

C-Type = 1 (RSVP-AGGREGATE-IPv4-PCN-request)

+-----+-----+-----+-----+	
	IPv4 PCN-ingress-node Address (4 bytes)
+-----+-----+-----+-----+	
	IPv4 PCN-egress-node Address (4 bytes)
+-----+-----+-----+-----+	
	IPv4 Decision Point Address (4 bytes)
+-----+-----+-----+-----+	
R	Reserved
+-----+-----+-----+-----+	

- o) RSVP-AGGREGATE-IPv6-PCN-request: PCN object, when IPv6 addresses are used:

Class = 248 (PCN)

C-Type = 2 (RSVP-AGGREGATE-IPv6-PCN-request)

+-----+-----+-----+-----+	
	IPv6 PCN-ingress-node Address (16 bytes)
+	
+	
+-----+-----+-----+-----+	
	IPv6 PCN-egress-node Address (16 bytes)
+	
+	
+-----+-----+-----+-----+	
	Decision Point Address (16 bytes)
+	
+	
+-----+-----+-----+-----+	
R	Reserved
+-----+-----+-----+-----+	

- o) RSVP-AGGREGATE-IPv4-PCN-response: PCN object, IPv4 addresses are used:  
 Class = 248 (PCN)  
 C-Type = 3 (RSVP-AGGREGATE-IPv4-PCN-response)

```

+-----+-----+-----+-----+
| IPv4 PCN-ingress-node Address (4 bytes) |
+-----+-----+-----+-----+
| IPv4 PCN-egress-node Address (4 bytes) |
+-----+-----+-----+-----+
| IPv4 Decision Point Address (4 bytes) |
+-----+-----+-----+-----+
| PCN-sent-rate |
+-----+-----+-----+-----+

```

- o) RSVP-AGGREGATE-IPv6-PCN-response: PCN object, IPv6 addresses are used:  
 Class = 248 (PCN)  
 C-Type = 4 (RSVP-AGGREGATE-IPv6-PCN-response)

```

+-----+-----+-----+-----+
|                                     |
+                                     +
| IPv6 PCN-ingress-node Address (16 bytes) |
+                                     +
|                                     |
+-----+-----+-----+-----+
|                                     |
+                                     +
| IPv6 PCN-egress-node Address (16 bytes) |
+                                     +
|                                     |
+-----+-----+-----+-----+
|                                     |
+                                     +
| Decision Point Address (16 bytes) |
+                                     +
|                                     |
+-----+-----+-----+-----+
| PCN-sent-rate |
+-----+-----+-----+-----+

```

The fields carried by the PCN object are specified in [RFC6663], [RFC6661] and [RFC6662]:

- o the IPv4 or IPv6 address of the PCN-ingress-node (Aggregator) and the IPv4 or IPv6 address of the PCN-egress-node (Deaggregator); together they specify the ingress-egress-aggregate to which the report refers. According to [RFC6663] the report should carry the identifier of the PCN-ingress-node (Aggregator) and the identifier of the PCN-egress-node (Deaggregator) (typically their IP addresses);
- o Decision Point address specify the IPv4 or IPv6 address of the Decision Point. In this document this field MUST contain the IP address of the Deaggregator.
- o "R": 1 bit flag that when set to ON, signifies, according to [RFC6661] and [RFC6662], that the PCN-ingress-node (Aggregator) MUST provide an estimate of the rate (PCN-sent-rate) at which the PCN-ingress-node (Aggregator) is receiving PCN-traffic that is destined for the given ingress-egress-aggregate.
- o "Reserved": 31 bits that are currently not used by this document and are reserved. These SHALL be set to 0 and SHALL be ignored on reception.
- o PCN-sent-rate: the PCN-sent-rate for the given ingress-egress-aggregate. It is expressed in octets/second; its format is a 32-bit IEEE floating point number; The PCN-sent-rate is specified in [RFC6661] and [RFC6662] and it represents the estimate of the rate at which the PCN-ingress-node (Aggregator) is receiving PCN-traffic that is destined for the given ingress-egress-aggregate.

## 5. Security Considerations

The security considerations specified in [RFC2205], [RFC4860] and [RFC5559] apply to this document. In addition, [RFC4230] and [RFC6411] provide useful guidance on RSVP security mechanisms.

Security within a PCN domain is fundamentally based on the controlled environment trust assumption stated in Section 6.3.1 of [RFC5559], in particular that all PCN-nodes are PCN-enabled and are trusted to perform accurate PCN-metering and PCN-marking.

In the PCN domain environments addressed by this document, Generic Aggregate Resource ReSerVation Protocol (RSVP) messages specified in [RFC4860] are used for support of the PCN Controlled Load (CL) and Single Marking (SM) edge behaviors over a Diffserv cloud using Pre-Congestion Notification. Hence the security mechanisms discussed in [RFC4860] are applicable. Specifically, the INTEGRITY object [RFC2747][RFC3097] can be used to provide hop-by-hop RSVP message integrity, node authentication and replay protection, thereby protecting against corruption and spoofing of RSVP messages and PCN feedback conveyed by RSVP messages.

For these reasons, this document does not introduce significant additional security considerations beyond those discussed in

[RFC5559] and [RFC4860].

## 6. IANA Considerations

IANA has modified the RSVP parameters registry, 'Class Names, Class Numbers, and Class Types' subregistry, to add a new Class Number and assign 4 new C-Types under this new Class Number, as described below, see Section 4.1:

Class Number	Class Name	Reference
-----	-----	-----
248	PCN	this document
Class Types or C-Types:		
1	RSVP-AGGREGATE-IPv4-PCN-request	this document
2	RSVP-AGGREGATE-IPv6-PCN-request	this document
3	RSVP-AGGREGATE-IPv4-PCN-response	this document
4	RSVP-AGGREGATE-IPv6-PCN-response	this document

When this draft is published as an RFC, IANA should update the reference for the above 5 items to that published RFC (and the RFC Editor should remove this sentence).

## 7. Acknowledgments

We would like to thank the authors of [draft-lefaucheur-rsvp-ecn-01.txt], since some ideas used in this document are based on the work initiated in [draft-lefaucheur-rsvp-ecn-01.txt]. Moreover, we would like to thank Bob Briscoe, David Black, Ken Carlberg, Tom Taylor, Philip Eardley, Michael Menth, Toby Moncaster, James Polk, Scott Bradner, Lixia Zhang and Robert Sparks for the provided comments. In particular, we would like to thank Francois Le Faucheur for contributing in addition to comments also to a significant amount of text.

## 8. Normative References

- [RFC6661] T. Taylor, A. Charny, F. Huang, G. Karagiannis, M. Menth, "PCN Boundary Node Behaviour for the Controlled Load (CL) Mode of Operation", July 2012.
- [RFC6662] A. Charny, J. Zhang, G. Karagiannis, M. Menth, T. Taylor, "PCN Boundary Node Behaviour for the Single Marking (SM) Mode of Operation", July 2012.
- [RFC6663] G. Karagiannis, T. Taylor, K. Chan, M. Menth, P. Eardley, " Requirements for Signaling of (Pre-) Congestion Information in a DiffServ Domain", July 2012.



- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2205] Braden, R., ed., et al., "Resource ReSerVation Protocol (RSVP)- Functional Specification", RFC 2205, September 1997.
- [RFC3140] Black, D., Brim, S., Carpenter, B., and F. Le Faucheur, "Per Hop Behavior Identification Codes", RFC 3140, June 2001.
- [RFC3175] Baker, F., Iturralde, C., Le Faucheur, F., and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", RFC 3175, September 2001.
- [RFC4495] Polk, J. and S. Dhesikan, "A Resource Reservation Protocol (RSVP) Extension for the Reduction of Bandwidth of a Reservation Flow", RFC 4495, May 2006.
- [RFC4860] F. Le Faucheur, B. Davie, P. Bose, C. Christou, M. Davenport, "Generic Aggregate Resource ReSerVation Protocol (RSVP) Reservations", RFC4860, May 2007.
- [RFC5670] Eardley, P., "Metering and Marking Behaviour of PCN-Nodes", RFC 5670, November 2009.
- [RFC6660] Moncaster, T., Briscoe, B., and M. Menth, "Baseline Encoding and Transport of Pre-Congestion Information", RFC 6660, July 2012.

## 9. Informative References

- [draft-lefaucheur-rsvp-ecn-01.txt] Le Faucheur, F., Charny, A., Briscoe, B., Eardley, P., Chan, K., and J. Babiarz, "RSVP Extensions for Admission Control over Diffserv using Pre-congestion Notification (PCN) (Work in progress)", June 2006.
- [RFC1633] Braden, R., Clark, D., and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994.
- [RFC2211] J. Wroclawski, Specification of the Controlled-Load Network Element Service, September 1997
- [RFC2212] S. Shenker et al., Specification of Guaranteed Quality of Service, September 1997
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "A framework for Differentiated Services", RFC 2475, December 1998.

[RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, January 2000.

[RFC2753] Yavatkar, R., D. Pendarakis and R. Guerin, "A Framework for Policy-based Admission Control", January 2000.

[RFC2998] Bernet, Y., Yavatkar, R., Ford, P., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J. and E. Felstaine, "A Framework for Integrated Services Operation Over DiffServ Networks", RFC 2998, November 2000.

[RFC3097] Braden, R. and L. Zhang, "RSVP Cryptographic Authentication -- Updated Message Type Value", RFC 3097, April 2001.

[RFC4230] H. Tschofenig, R. Graveman, "RSVP Security Properties", RFC 4230, December 2005.

[RFC5559] Eardley, P., "Pre-Congestion Notification (PCN) Architecture", RFC 5559, June 2009.

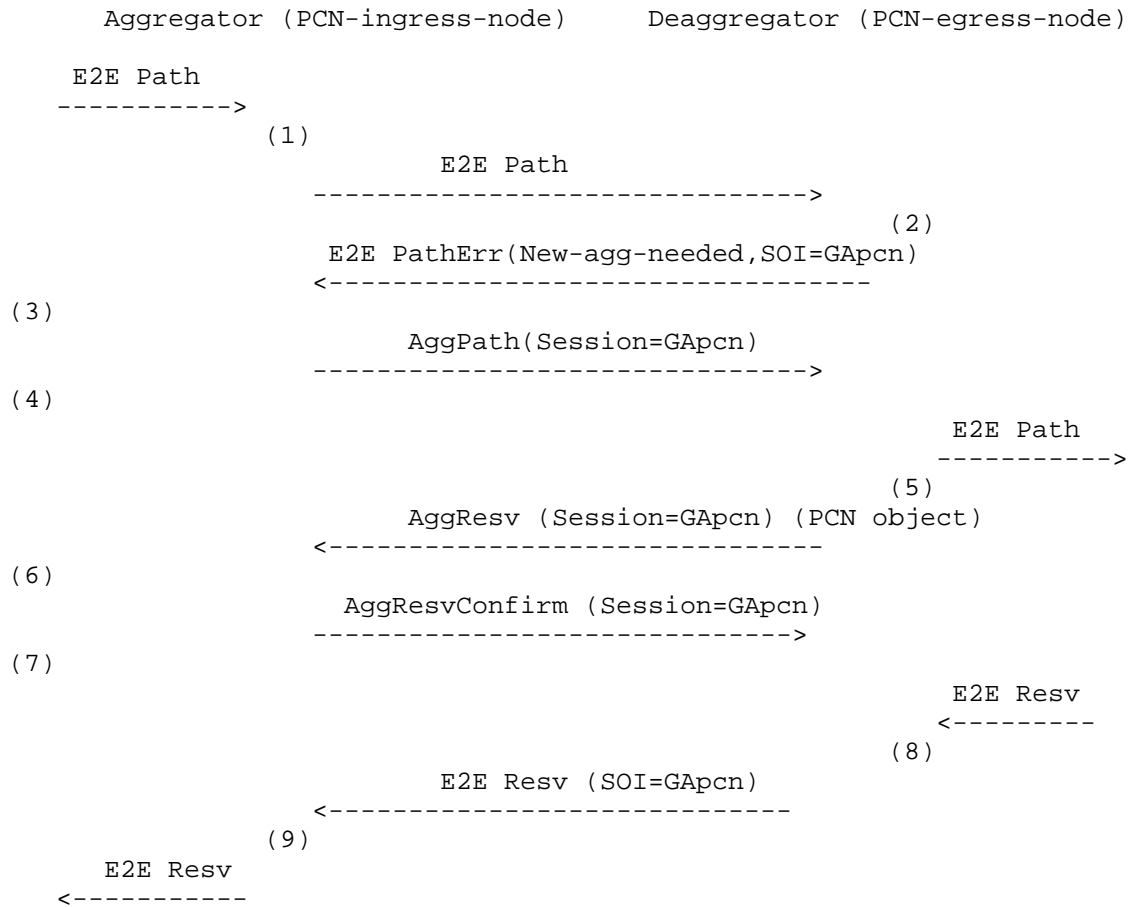
[RFC6411] M. Behringer, F. Le Faucheur, B. Weis, "Applicability of Keying Methods for RSVP Security", RFC 6411, October 2011.

[SIG-NESTED] Baker, F. and P. Bose, "QoS Signaling in a Nested Virtual Private Network", Work in Progress, July 2007.

## 10. Appendix A: Example Signaling Flow

This appendix is based on the appendix provided in [RFC4860]. In particular, it provides an example signaling flow of the specification detailed in Section 3 and 4.

This signaling flow assumes an environment where E2E reservations are aggregated over generic aggregate RSVP reservations and applied over a PCN domain. In particular the Aggregator (PCN-ingress-node) and Deaggregator (PCN-egress-node) are located at the boundaries of the PCN domain. The PCN-interior-nodes are located within the PCN-domain, between the PCN-boundary nodes, but are not shown in this Figure. It illustrates a possible RSVP message flow that could take place in the successful establishment of a unicast E2E reservation that is the first between a given pair of Aggregator/Deaggregator.



(1) The Aggregator forwards E2E Path into the aggregation region after modifying its IP protocol number to RSVP-E2E-IGNORE

(2) Let's assume no Aggregate Path exists. To be able to accurately update the ADSPEC of the E2E Path, the Deaggregator needs the ADSPEC of Aggregate Path. In this example, the Deaggregator elects to instruct the Aggregator to set up an Aggregate Path state for the PCN PHB-ID. To do that, the Deaggregator sends an E2E PathErr message with a New-Agg-Needed PathErr code.

The PathErr message also contains a SESSION-OF-INTEREST (SOI) object. The SOI contains a GENERIC-AGGREGATE SESSION (GApcn) whose PHB-ID is set to the PCN PHB-ID. The GENERIC-AGGREGATE SESSION contains an interface-independent Deaggregator address inside the DestAddress and appropriate values inside the vDstPort and Extended vDstPort fields. In this document, the Extended vDstPort SHOULD contain the IPv4 or IPv6 address of the Aggregator.

(3) The Aggregator follows the request from the Deaggregator and

signals an Aggregate Path for the GENERIC-AGGREGATE Session (GApn).

- (4) The Deaggregator takes into account the information contained in the ADSPEC from both Aggregate Paths and updates the E2E Path ADSPEC accordingly. The PCN-egress-node MUST NOT perform the RSVP-TTL vs IP TTL-check and MUST NOT update the ADSpec Break bit. This is because the whole PCN-domain is effectively handled by E2E RSVP as a virtual link on which integrated service is indeed supported (and admission control performed) so that the Break bit MUST NOT be set, see also [draft-lefaucheur-rsvp-ecn-01]. The Deaggregator also modifies the E2E Path IP protocol number to RSVP before forwarding it.
- (5) In this example, the Deaggregator elects to immediately proceed with establishment of the generic aggregate reservation. In effect, the Deaggregator can be seen as anticipating the actual demand of E2E reservations so that the generic aggregate reservation is in place when the E2E Resv request arrives, in order to speed up establishment of E2E reservations. Here it is also assumed that the Deaggregator includes the optional Resv Confirm Request in the Aggregate Resv message.
- (6) The Aggregator merely complies with the received ResvConfirm Request and returns the corresponding Aggregate ResvConfirm.
- (7) The Deaggregator has explicit confirmation that the generic aggregate reservation is established.
- (8) On receipt of the E2E Resv, the Deaggregator applies the mapping policy defined by the network administrator to map the E2E Resv onto a generic aggregate reservation. Let's assume that this policy is such that the E2E reservation is to be mapped onto the generic aggregate reservation with the PCN PHB-ID=x. The Deaggregator knows that a generic aggregate reservation (GApn) is in place for the corresponding PHB-ID since (7). At this step the Deaggregator maps the generic aggregated reservation onto one ingress-egress-aggregate maintained by the Deaggregator (as a PCN-egress-node), see Section 3.7. The Deaggregator performs admission control of the E2E Resv onto the generic Aggregate reservation for the PCN PHB-ID (GApn). The Deaggregator takes also into account the PCN admission control procedure as specified in [RFC6661] and [RFC6662], see Section 3.7. If one or both the admission control procedures (PCN based admission control procedure and admission control procedure specified in [RFC4860]) are not successful, then the E2E Resv is not admitted onto the associated RSVP generic aggregate reservation for the PCN PHB-ID (GApn). Otherwise, assuming that the generic aggregate reservation for the PCN (GApn) had been established with sufficient bandwidth to support the E2E Resv, the Deaggregator adjusts its counter, tracking the unused bandwidth on the generic aggregate reservation. Then it forwards the E2E Resv to the Aggregator including a SESSION-OF-INTEREST

object conveying the selected mapping onto GApcn (and hence onto the PCN PHB-ID).

- (9) The Aggregator records the mapping of the E2E Resv onto GApcn (and onto the PCN PHB-ID). The Aggregator removes the SOI object and forwards the E2E Resv towards the sender.

## 11. Authors' Address

Georgios Karagiannis  
Huawei Technologies  
Hansaallee 205,  
40549 Dusseldorf,  
Germany  
Email: Georgios.Karagiannis@huawei.com

Anurag Bhargava  
Cisco Systems, Inc.  
7100-9 Kit Creek Road  
PO Box 14987  
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709-4987  
USA  
Email: anuragb@cisco.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 20, 2016

Y. Nishida  
GE Global Research  
P. Natarajan  
Cisco Systems  
A. Caro  
BBN Technologies  
P. Amer  
University of Delaware  
K. Nielsen  
Ericsson  
February 17, 2016

SCTP-PF: Quick Failover Algorithm in SCTP  
draft-ietf-tsvwg-sctp-failover-16.txt

Abstract

SCTP supports multi-homing. However, when the failover operation specified in RFC4960 is followed, there can be significant delay and performance degradation in the data transfer path failover. To overcome this problem this document specifies a quick failover algorithm (SCTP-PF) based on the introduction of a Potentially Failed (PF) state in SCTP Path Management.

The document also specifies a dormant state operation of SCTP. This dormant state operation is required to be followed by an SCTP-PF implementation, but it may equally well be applied by a standard RFC4960 SCTP implementation.

Additionally, the document introduces an alternative switchback operation mode called Primary Path Switchover that will be beneficial in certain situations. This mode of operation applies to both a standard RFC4960 SCTP implementation as well as to a SCTP-PF implementation.

The procedures defined in the document require only minimal modifications to the RFC4960 specification. The procedures are sender-side only and do not impact the SCTP receiver.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 20, 2016.

#### Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	3
2. Conventions and Terminology . . . . .	4
3. SCTP with Potentially Failed Destination State (SCTP-PF) . .	4
3.1. Overview . . . . .	4
3.2. Specification of the SCTP-PF Procedures . . . . .	5
4. Dormant State Operation . . . . .	9
4.1. SCTP Dormant State Procedure . . . . .	10
5. Primary Path Switchover . . . . .	11
6. Suggested SCTP Protocol Parameter Values . . . . .	12
7. Socket API Considerations . . . . .	12
7.1. Support for the Potentially Failed Path State . . . . .	13
7.2. Peer Address Thresholds (SCTP_PEER_ADDR_THLDS) Socket Option . . . . .	14
7.3. Exposing the Potentially Failed Path State (SCTP_EXPOSE_POTENTIALLY_FAILED_STATE) Socket Option . .	15
8. Security Considerations . . . . .	15
9. MIB Considerations . . . . .	16
10. IANA Considerations . . . . .	16
11. Acknowledgements . . . . .	16
12. Proposed Change of Status (to be Deleted before Publication)	17
13. References . . . . .	17

13.1. Normative References . . . . .	17
13.2. Informative References . . . . .	17
Appendix A. Discussions of Alternative Approaches . . . . .	18
A.1. Reduce Path.Max.Retrans (PMR) . . . . .	18
A.2. Adjust RTO related parameters . . . . .	19
Appendix B. Discussions for Path Bouncing Effect . . . . .	20
Appendix C. SCTP-PF for SCTP Single-homed Operation . . . . .	20
Authors' Addresses . . . . .	21

## 1. Introduction

The Stream Control Transmission Protocol (SCTP) specified in [RFC4960] supports multi-homing at the transport layer. SCTP's multi-homing features include failure detection and failover procedures to provide network interface redundancy and improved end-to-end fault tolerance. In SCTP's current failure detection procedure, the sender must experience Path.Max.Retrans (PMR) number of consecutive failed timer-based retransmissions on a destination address before detecting a path failure. Until detecting the path failure, the sender continues to transmit data on the failed path. The prolonged time in which [RFC4960] SCTP continues to use a failed path severely degrades the performance of the protocol. To address this problem, this document specifies a quick failover algorithm (SCTP-PF) based on the introduction of a new Potentially Failed (PF) path state in SCTP path management. The performance deficiencies of the [RFC4960] failover operation, and the improvements obtainable from the introduction of a Potentially Failed state in SCTP, were proposed and documented in [NATARAJAN09] for Concurrent Multipath Transfer SCTP [IYENGAR06].

While SCTP-PF can accelerate failover process and improve performance, the risks that an SCTP endpoint enters the dormant state where all destination addresses are inactive can be increased. [RFC4960] leaves the protocol operation during dormant state to implementations and encourages to avoid entering the state as much as possible by careful tuning of the Path.Max.Retrans (PMR) and Association.Max.Retrans (AMR) parameters. We specify a dormant state operation for SCTP-PF which makes SCTP-PF provide the same disruption tolerance as [RFC4960] despite that the dormant state may be entered more quickly. The dormant state operation may equally well be applied by an [RFC4960] implementation and will here serve to provide added fault tolerance for situations where the tuning of the Path.Max.Retrans (PMR) and Association.Max.Retrans (AMR) parameters fail to provide adequate prevention of the entering of the dormant state.

The operation after the recovery of a failed path also impacts the performance of the protocol. With the procedures specified in



[RFC4960] SCTP will, after a failover from the primary path, switch back to use the primary path for data transfer as soon as this path becomes available again. From a performance perspective such a forced switchback of the data transmission path can be suboptimal as the CWND towards the original primary destination address has to be rebuilt once data transfer resumes, [CARO02]. As an optional alternative to the switchback operation of [RFC4960], this document specifies an alternative Primary Path Switchover procedure which avoid such forced switchbacks of the data transfer path. The Primary Path Switchover operation was originally proposed in [CARO02].

While SCTP-PF primarily is motivated by a desire to improve the multi-homed operation, the feature applies also to SCTP single-homed operation. Here the algorithm serves to provide increased failure detection on idle associations, whereas the failover or switchback aspects of the algorithm will not be activated. This is discussed in more detail in Appendix C.

A brief description of the motivation for the introduction of the Potentially Failed state including a discussion of alternative approaches to mitigate the deficiencies of the [RFC4960] failover operation are given in the Appendices. Discussion of path bouncing effects that might be caused by frequent switchovers, are also provided there.

## 2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. SCTP with Potentially Failed Destination State (SCTP-PF)

### 3.1. Overview

To minimize the performance impact during failover, the sender should avoid transmitting data to a failed destination address as early as possible. In the [RFC4960] SCTP path management scheme, the sender stops transmitting data to a destination address only after the destination address is marked inactive. This process takes a significant amount of time as it requires the error counter of the destination address to exceed the Path.Max.Retrans (PMR) threshold. The issue cannot simply be mitigated by lowering of the PMR threshold because this may result in spurious failure detection and unnecessary prevention of the usage of a preferred primary path. Also due to the coupled tuning of the Path.Max.Retrans (PMR) and the Association.Max.Retrans (AMR) parameter values in [RFC4960], lowering

of the PMR threshold may result in lowering of the AMR threshold, which would result in decrease of the fault tolerance of SCTP.

The solution provided in this document is to extend the SCTP path management scheme of [RFC4960] by the addition of the Potentially Failed (PF) state as an intermediate state in between the active and inactive state of a destination address in the [RFC4960] path management scheme, and let the failover of data transfer away from a destination address be driven by the entering of the PF state instead of by the entering of the inactive state. Thereby SCTP may perform quick failover without negatively impacting the overall fault tolerance of [RFC4960] SCTP. At the same time, RTO-based HEARTBEAT probing is initiated towards a destination address once it enters PF state. Thereby SCTP may quickly ascertain whether network connectivity towards the destination address is broken or whether the failover was spurious. In the case where the failover was spurious data transfer may quickly resume towards the original destination address.

The new failure detection algorithm assumes that loss detected by a timeout implies either severe congestion or network connectivity failure. It recommends that by default a destination address is classified as PF at the occurrence of the first timeout.

### 3.2. Specification of the SCTP-PF Procedures

The SCTP-PF operation is specified as follows:

1. The sender maintains a new tunable SCTP Protocol Parameter called PotentiallyFailed.Max.Retrans (PFMR). The PFMR defines the new intermediate PF threshold on the destination address error counter. When this threshold is exceeded the destination address is classified as PF. The RECOMMENDED value of PFMR is 0. If PFMR is set to be greater than or equal to Path.Max.Retrans (PMR), the resulting PF threshold will be so high that the destination address will reach the inactive state before it can be classified as PF.
2. The error counter of an active destination address is incremented or cleared as specified in [RFC4960]. This means that the error counter of the destination address in active state will be incremented each time the T3-rtx timer expires, or each time a HEARTBEAT chunk is sent when idle and not acknowledged within an RTO. When the value in the destination address error counter exceeds PFMR, the endpoint MUST mark the destination address as in the PF state.

3. A SCTP-PF sender SHOULD NOT send data to destination addresses in PF state when alternative destination addresses in active state are available. Specifically this means that:
  - i When there is outbound data to send and the destination address presently used for data transmission is in PF state, the sender SHOULD choose a destination address in active state, if one exists, and use this destination address for data transmission.
  - ii As specified in [RFC4960] section 6.4.1, when the sender retransmits data that has timed out, it should attempt to pick a new destination address for data retransmission. In this case, the sender SHOULD choose an alternate destination transport address in active state if one exists.
  - iii When there is outbound data to send and the SCTP user explicitly requests to send data to a destination address in PF state, the sender SHOULD send the data to an alternate destination address in active state if one exists.

When choosing among multiple destination addresses in active state an SCTP sender will follow the guiding principles of section 6.4.1 of [RFC4960] of choosing most divergent source-destination pairs compared with, for i.: the destination address in PF state that it performs a failover from, and for ii.: the destination address towards which the data timed out. Rules for picking the most divergent source-destination pair are an implementation decision and are not specified within this document.

In all cases, the sender MUST NOT change the state of chosen destination address, whether this state be active or PF, and it MUST NOT clear the error counter of the destination address as a result of choosing the destination address for data transmission.

4. When the destination addresses are all in PF state or some in PF state and some in inactive state, the sender MUST choose one destination address in PF state and SHOULD transmit or retransmit data to this destination address using the following rules:
  - A. The sender SHOULD choose the destination in PF state with the lowest error count (fewest consecutive timeouts) for data transmission and transmit or retransmit data to this destination.

- B. When there are multiple destination addresses in PF state with same error count, the sender should let the choice among the multiple destination addresses in PF state with equal error count be based on the [RFC4960], section 6.4.1, principles of choosing most divergent source-destination pairs when executing (potentially consecutive) retransmission. Rules for picking the most divergent source-destination pair are an implementation decision and are not specified within this document.

The sender MUST NOT change the state and the error counter of any destination addresses as the result of the selection.

5. The HB.interval of the Path Heartbeat function of [RFC4960] MUST be ignored for destination addresses in PF state. Instead HEARTBEAT chunks are sent to destination addresses in PF state once per RTO. HEARTBEAT chunks SHOULD be sent to destination addresses in PF state, but the sending of HEARTBEATS MUST honor whether the Path Heartbeat function (Section 8.3 of [RFC4960]) is enabled for the destination address or not. I.e., if the Path Heartbeat function is disabled for the destination address in question, HEARTBEATS MUST NOT be sent. Note that when Heartbeat function is disabled, it may take longer to transition a destination address in PF state back to active state.
6. HEARTBEATS are sent when a destination address reaches the PF state. When a HEARTBEAT chunk is not acknowledged within the RTO, the sender increments the error counter and exponentially backs off the RTO value. If the error counter is less than PMR, the sender transmits another packet containing the HEARTBEAT chunk immediately after timeout expiration on the previous HEARTBEAT. When data is being transmitted to a destination address in the PF state, the transmission of a HEARTBEAT chunk MAY be omitted in case where the receipt of a SACK of the data or a T3-rtx timer expiration on the data can provide equivalent information, such as the case where the data chunk has been transmitted to a single destination address only. Likewise, the timeout of a HEARTBEAT chunk MAY be ignored if data is outstanding towards the destination address.
7. When the sender receives a HEARTBEAT ACK from a HEARTBEAT sent to a destination address in PF state, the sender SHOULD clear the error counter of the destination address and transition the destination address back to active state. However, there may be a situation where HEARTBEAT chunks can go through while DATA chunks cannot. Hence, in a situation where a HEARTBEAT ACK arrives while there is data outstanding towards the destination address to which the HEARTBEAT was sent, then an implementation

MAY choose to not have the HEARTBEAT ACK reset the error counter, but have the error counter reset await the fate of the outstanding data transmission. This situation can happen when data is sent to a destination address in PF state. When the sender resumes data transmission on a destination address after a transition of the destination address from PF to active state, it MUST do this following the prescriptions of Section 7.2 of [RFC4960].

8. Additional (PMR - PFMR) consecutive timeouts on a destination address in PF state confirm the path failure, upon which the destination address transitions to the inactive state. As described in [RFC4960], the sender (i) SHOULD notify the ULP about this state transition, and (ii) transmit HEARTBEAT chunks to the inactive destination address at a lower HB.interval frequency as described in Section 8.3 of [RFC4960] (when the Path Heartbeat function is enabled for the destination address).
9. Acknowledgments for chunks that have been transmitted to multiple destinations (i.e., a chunk which has been retransmitted to a different destination address than the destination address to which the chunk was first transmitted) SHOULD NOT clear the error count for an inactive destination address and SHOULD NOT move a destination address in PF state back to active state, since a sender cannot disambiguate whether the ACK was for the original transmission or the retransmission(s). A SCTP sender MAY clear the error counter and move a destination address back to active state by information other than acknowledgments, when it can uniquely determine which destination, among multiple destination addresses, the chunk reached. This document makes no reference to what such information could consist of, nor how such information could be obtained.
10. Acknowledgments for data chunks that has been transmitted to one destination address only MUST clear the error counter for the destination address and MUST transition a destination address in PF state back to active state. This situation can happen when new data is sent to a destination address in the PF state. It can also happen in situations where the destination address is in the PF state due to the occurrence of a spurious T3-rtx timer and acknowledgments start to arrive for data sent prior to occurrence of the spurious T3-rtx and data has not yet been retransmitted towards other destinations. This document does not specify special handling for detection of or reaction to spurious T3-rtx timeouts, e.g., for special operation vis-a-vis the congestion control handling or data retransmission operation towards a destination address which undergoes a transition from

active to PF to active state due to a spurious T3-rtx timeout. But it is noted that this is an area which would benefit from additional attention, experimentation and specification for single-homed SCTP as well as for multi-homed SCTP protocol operation.

11. When all destination addresses are in inactive state, and SCTP protocol operation thus is said to be in dormant state, the prescriptions given in Section 4 shall be followed.
12. The SCTP stack SHOULD expose the PF state of its destination addresses to the ULP as well as provide the means to notify the ULP of state transitions of its destination addresses from active to PF, and vice-versa. However it is recommended that an SCTP stack implementing SCTP-PF also allows for that the ULP is kept ignorant of the PF state of its destinations and the associated state transitions, thus allowing for retain of the simpler state transition model of RFC4960 in the ULP. For this reason it is recommended that an SCTP stack implementing SCTP-PF also provides the ULP with the means to suppress exposure of the PF state and the associated state transitions.

#### 4. Dormant State Operation

In a situation with complete disruption of the communication in between the SCTP Endpoints, the aggressive HEARTBEAT transmissions of SCTP-PF on destination addresses in PF state may make the association enter dormant state faster than a standard [RFC4960] SCTP implementation given the same setting of Path.Max.Retrans (PMR) and Association.Max.Retrans (AMR). For example, an SCTP association with two destination addresses typically would reach dormant state in half the time of an [RFC4960] SCTP implementation in such situations. This is because a SCTP PF sender will send HEARTBEATS and data retransmissions in parallel with RTO intervals when there are multiple destinations addresses in PF state. This argument presumes that  $RTO \ll HB.interval$  of [RFC4960]. With the design goal that SCTP-PF shall provide the same level of disruption tolerance as an [RFC4960] SCTP implementation with the same Path.Max.Retrans (PMR) and Association.Max.Retrans (AMR) setting, we prescribe for that an SCTP-PF implementation SHOULD operate as described below in Section 4.1 during dormant state.

An SCTP-PF implementation MAY choose a different dormant state operation than the one described below in Section 4.1 provided that the solution chosen does not decrease the fault tolerance of the SCTP-PF operation.

The below prescription for SCTP-PF dormant state handling MUST NOT be coupled to the value of the PFMR, but solely to the activation of SCTP-PF logic in an SCTP implementation.

It is noted that the below dormant state operation is considered to provide added disruption tolerance also for an [RFC4960] SCTP implementation, and that it can be sensible for an [RFC4960] SCTP implementation to follow this mode of operation. For an [RFC4960] SCTP implementation the continuation of data transmission during dormant state makes the fault tolerance of SCTP be more robust towards situations where some, or all, alternative paths of an SCTP association approach, or reach, inactive state before the primary path used for data transmission observes trouble.

#### 4.1. SCTP Dormant State Procedure

- a. When the destination addresses are all in inactive state and data is available for transfer, the sender MUST choose one destination and transmit data to this destination address.
- b. The sender MUST NOT change the state of the chosen destination address (it remains in inactive state) and it MUST NOT clear the error counter of the destination address as a result of choosing the destination address for data transmission.
- c. The sender SHOULD choose the destination in inactive state with the lowest error count (fewest consecutive timeouts) for data transmission. When there are multiple destinations with same error count in inactive state, the sender SHOULD attempt to pick the most divergent source - destination pair from the last source - destination pair where failure was observed. Rules for picking the most divergent source-destination pair are an implementation decision and are not specified within this document. To support differentiation of inactive destination addresses based on their error count SCTP will need to allow for increment of the destination address error counters up to some reasonable limit above PMR+1, thus changing the prescriptions of [RFC4960], section 8.3, in this respect. The exact limit to apply is not specified in this document but it is considered reasonable to require for the limit to be an order of magnitude higher than the PMR value. A sender MAY choose to deploy other strategies than the strategy defined here. The strategy to prioritize the last active destination address, i.e., the destination address with the fewest error counts is optimal when some paths are permanently inactive, but suboptimal when a path instability is transient.

## 5. Primary Path Switchover

The objective of the Primary Path Switchover operation is to allow the SCTP sender to continue data transmission on a new working path even when the old primary destination address becomes active again. This is achieved by having SCTP perform a switchover of the primary path to the new working path if the error counter of the primary path exceeds a certain threshold. This mode of operation can be applied not only to SCTP-PF implementations, but also to [RFC4960] implementations.

The Primary Path Switchover operation requires only sender side changes. The details are:

1. The sender maintains a new tunable parameter, called Primary.Switchover.Max.Retrans (PSMR). For SCTP-PF implementations, the PSMR MUST be set greater or equal to the PFMR value. For [RFC4960] implementations the PSMR MUST be set greater or equal to the PMR value. Implementations MUST reject any other values of PSMR.
2. When the path error counter on a set primary path exceeds PSMR, the SCTP implementation MUST autonomously select and set a new primary path.
3. The primary path selected by the SCTP implementation MUST be the path which at the given time would be chosen for data transfer. A previously failed primary path can be used as data transfer path as per normal path selection when the present data transfer path fails.
4. For SCTP-PF, the recommended value of PSMR is PFMR when Primary Path Switchover operation mode is used. This means that no forced switchback to a previously failed primary path is performed. An SCTP-PF implementation of Primary Path Switchover MUST support the setting of PSMR = PFMR. A SCTP-PF implementation of Primary Path Switchover MAY support setting of PSMR > PFMR.
5. For [RFC4960] SCTP, the recommended value of PSMR is PMR when Primary Path Switchover is used. This means that no forced switchback to a previously failed primary path is performed. A [RFC4960] SCTP implementation of Primary Path Switchover MUST support the setting of PSMR = PMR. An [RFC4960] SCTP implementation of Primary Path Switchover MAY support larger settings of PSMR > PMR.



6. It MUST be possible to disable the Primary Path Switchover operation and obtain the standard switchback operation of [RFC4960].

The manner of switchover operation that is most optimal in a given scenario depends on the relative quality of a set primary path versus the quality of alternative paths available as well as on the extent to which it is desired for the mode of operation to enforce traffic distribution over a number of network paths. I.e., load distribution of traffic from multiple SCTP associations may be sought to be enforced by distribution of the set primary paths with [RFC4960] switchback operation. However as [RFC4960] switchback behavior is suboptimal in certain situations, especially in scenarios where a number of equally good paths are available, an SCTP implementation MAY support also, as alternative behavior, the Primary Path Switchover mode of operation and MAY enable it based on applications' requests.

For an SCTP implementation that implements the Primary Path Switchover operation, this specification RECOMMENDS that the standard RFC4960 switchback operation is retained as the default operation.

## 6. Suggested SCTP Protocol Parameter Values

This document does not alter the [RFC4960] value recommendation for the SCTP Protocol Parameters defined in [RFC4960].

The following protocol parameter is RECOMMENDED:

PotentiallyFailed.Max.Retrans (PFMR) - 0

## 7. Socket API Considerations

This section describes how the socket API defined in [RFC6458] is extended to provide a way for the application to control and observe the SCTP-PF behavior as well as the Primary Path Switchover function.

Please note that this section is informational only.

A socket API implementation based on [RFC6458] is, by means of the existing SCTP\_PEER\_ADDR\_CHANGE event, extended to provide the event notification when a peer address enters or leaves the potentially failed state as well as the socket API implementation is extended to expose the potentially failed state of a peer address in the existing SCTP\_GET\_PEER\_ADDR\_INFO structure.

Furthermore, two new read/write socket options for the level IPPROTO\_SCTP and the name SCTP\_PEER\_ADDR\_THLDS and

SCTP\_EXPOSE\_POTENTIALLY\_FAILED\_STATE are defined as described below. The first socket option is used to control the values of the PFMR and PSMP parameters described in Section 3 and in Section 5. The second one controls the exposition of the potentially failed path state.

Support for the SCTP\_PEER\_ADDR\_THLDS and SCTP\_EXPOSE\_POTENTIALLY\_FAILED\_STATE socket options need also to be added to the function sctp\_opt\_info().

#### 7.1. Support for the Potentially Failed Path State

As defined in [RFC6458], the SCTP\_PEER\_ADDR\_CHANGE event is provided if the status of a peer address changes. In addition to the state changes described in [RFC6458], this event is also provided, if a peer address enters or leaves the potentially failed state. The notification as defined in [RFC6458] uses the following structure:

```
struct sctp_paddr_change {
    uint16_t spc_type;
    uint16_t spc_flags;
    uint32_t spc_length;
    struct sockaddr_storage spc_aaddr;
    uint32_t spc_state;
    uint32_t spc_error;
    sctp_assoc_t spc_assoc_id;
}
```

[RFC6458] defines the constants SCTP\_ADDR\_AVAILABLE, SCTP\_ADDR\_UNREACHABLE, SCTP\_ADDR\_REMOVED, SCTP\_ADDR\_ADDED, and SCTP\_ADDR\_MADE\_PRIM to be provided in the spc\_state field. This document defines in addition to that the new constant SCTP\_ADDR\_POTENTIALLY\_FAILED, which is reported if the affected address becomes potentially failed.

The SCTP\_GET\_PEER\_ADDR\_INFO socket option defined in [RFC6458] can be used to query the state of a peer address. It uses the following structure:

```
struct sctp_paddrinfo {
    sctp_assoc_t spinfo_assoc_id;
    struct sockaddr_storage spinfo_address;
    int32_t spinfo_state;
    uint32_t spinfo_cwnd;
    uint32_t spinfo_srtt;
    uint32_t spinfo_rto;
    uint32_t spinfo_mtu;
};
```

[RFC6458] defines the constants `SCTP_UNCONFIRMED`, `SCTP_ACTIVE`, and `SCTP_INACTIVE` to be provided in the `spinfo_state` field. This document defines in addition to that the new constant `SCTP_POTENTIALLY_FAILED`, which is reported if the peer address is potentially failed.

## 7.2. Peer Address Thresholds (`SCTP_PEER_ADDR_THLDS`) Socket Option

Applications can control the SCTP-PF behavior by getting or setting the number of consecutive timeouts before a peer address is considered potentially failed or unreachable. The same socket option is used by applications to set and get the number of timeouts before the primary path is changed automatically by the Primary Path Switchover function. This socket option uses the level `IPPROTO_SCTP` and the name `SCTP_PEER_ADDR_THLDS`.

The following structure is used to access and modify the thresholds:

```
struct sctp_paddrthlds {
    sctp_assoc_t spt_assoc_id;
    struct sockaddr_storage spt_address;
    uint16_t spt_pathmaxrxt;
    uint16_t spt_pathpfthld;
    uint16_t spt_pathcpthld;
};
```

`spt_assoc_id`: This parameter is ignored for one-to-one style sockets. For one-to-many style sockets the application may fill in an association identifier or `SCTP_FUTURE_ASSOC`. It is an error to use `SCTP_{CURRENT|ALL}_ASSOC` in `spt_assoc_id`.

`spt_address`: This specifies which peer address is of interest. If a wild card address is provided, this socket option applies to all current and future peer addresses.

`spt_pathmaxrxt`: Each peer address of interest is considered unreachable, if its path error counter exceeds `spt_pathmaxrxt`.

`spt_pathpfthld`: Each peer address of interest is considered Potentially Failed, if its path error counter exceeds `spt_pathpfthld`.

`spt_pathcpthld`: Each peer address of interest is not considered the primary remote address anymore, if its path error counter exceeds `spt_pathcpthld`. Using a value of `0xffff` disables the selection of a new primary peer address. If an implementation does not support the automatically selection of a new primary address, it should indicate an error with `errno` set to `EINVAL` if a value different

from 0xffff is used in `spt_pathcpthld`. For SCTP-PF, the setting of `spt_pathcpthld < spt_pathpfthld` should be rejected with `errno` set to `EINVAL`. For [RFC4960] SCTP, the setting of `spt_pathcpthld < spt_pathmaxrxt` should be rejected with `errno` set to `EINVAL`. A SCTP-PF implementation may support only setting of `spt_pathcpthld = spt_pathpfthld` and `spt_pathcpthld = 0xffff` and a [RFC4960] SCTP implementation may support only setting of `spt_pathcpthld = spt_pathmaxrxt` and `spt_pathcpthld = 0xffff`. In these cases SCTP shall reject setting of other values with `errno` set to `EINVAL`.

### 7.3. Exposing the Potentially Failed Path State (`SCTP_EXPOSE_POTENTIALLY_FAILED_STATE`) Socket Option

Applications can control the exposure of the potentially failed path state in the `SCTP_PEER_ADDR_CHANGE` event and the `SCTP_GET_PEER_ADDR_INFO` as described in Section 7.1. The default value is implementation specific.

This socket option uses the level `IPPROTO_SCTP` and the name `SCTP_EXPOSE_POTENTIALLY_FAILED_STATE`.

The following structure is used to control the exposition of the potentially failed path state:

```
struct sctp_assoc_value {
    sctp_assoc_t assoc_id;
    uint32_t assoc_value;
};
```

`assoc_id`: This parameter is ignored for one-to-one style sockets. For one-to-many style sockets the application may fill in an association identifier or `SCTP_FUTURE_ASSOC`. It is an error to use `SCTP_{CURRENT|ALL}_ASSOC` in `assoc_id`.

`assoc_value`: The potentially failed path state is exposed if and only if this parameter is non-zero.

## 8. Security Considerations

Security considerations for the use of SCTP and its APIs are discussed in [RFC4960] and [RFC6458].

The logic introduced by this document does not impact existing SCTP messages on the wire. Also, this document does not introduce any new SCTP messages on the wire that require new security considerations.

SCTP-PF makes SCTP not only more robust during primary path failure/congestion but also more vulnerable to network connectivity/

congestion attacks on the primary path. SCTP-PF makes it easier for an attacker to trick SCTP to change data transfer path, since the duration of time that an attacker needs to negatively influence the network connectivity is much shorter than [RFC4960]. However, SCTP-PF does not constitute a significant change in the duration of time and effort an attacker needs to keep SCTP away from the primary path. With the standard switchback operation [RFC4960] SCTP resumes data transfer on its primary path as soon as the next HEARTBEAT succeeds.

On the other hand, usage of the Primary Path Switchover mechanism, does change the threat analysis. This is because on-path attackers can force a permanent change of the data transfer path by blocking the primary path until the switchover of the primary path is triggered by the Primary Path Switchover algorithm. This especially will be the case when the Primary Path Switchover is used together with SCTP-PF with the particular setting of PSMR = PFMR = 0, as Primary Path Switchover here happens already at the first RTO timeout experienced. Users of the Primary Path Switchover mechanism should be aware of this fact.

The event notification of path state transfer from active to potentially failed state and vice versa gives attackers an increased possibility to generate more local events. However, it is assumed that event notifications are rate-limited in the implementation to address this threat.

#### 9. MIB Considerations

SCTP-PF introduces new SCTP algorithms for failover and switchback with associated new state parameters. It is recommended that the SCTP-MIB defined in [RFC3873] is updated to support the management of the SCTP-PF implementation. This can be done by extending the sctpAssocRemAddrActive field of the SCTPAssocRemAddrTable to include information of the PF state of the destination address and by adding new fields to the SCTPAssocRemAddrTable supporting PotentiallyFailed.Max.Retrans (PFMR) and Primary.Switchover.Max.Retrans (PSMR) parameters.

#### 10. IANA Considerations

This document does not create any new registries or modify the rules for any existing registries managed by IANA.

#### 11. Acknowledgements

The authors wish to thank Michael Tuexen for his many invaluable comments and for his very substantial support with the making of this document.

## 12. Proposed Change of Status (to be Deleted before Publication)

Initially this work looked to entail some changes of the Congestion Control (CC) operation of SCTP and for this reason the work was proposed as Experimental. These intended changes of the CC operation have since been judged to be irrelevant and are no longer part of the specification. As the specification entails no other potential harmful features, consensus exists in the WG to bring the work forward as PS.

Initially concerns have been expressed about the possibility for the mechanism to introduce path bouncing with potential harmful network impacts. These concerns are believed to be unfounded. This issue is addressed in Appendix B.

It is noted that the feature specified by this document is implemented by multiple SCTP SW implementations and furthermore that various variants of the solution have been deployed in telephony signaling environments for several years with good results.

## 13. References

### 13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

### 13.2. Informative References

- [CARO02] Caro Jr., A., Iyengar, J., Amer, P., Heinz, G., and R. Stewart, "A Two-level Threshold Recovery Mechanism for SCTP", Tech report, CIS Dept, University of Delaware , 7 2002.
- [CARO04] Caro Jr., A., Amer, P., and R. Stewart, "End-to-End Failover Thresholds for Transport Layer Multi homing", MILCOM 2004 , 11 2004.
- [CARO05] Caro Jr., A., "End-to-End Fault Tolerance using Transport Layer Multi homing", Ph.D Thesis, University of Delaware , 1 2005.

- [FALLON08] Fallon, S., Jacob, P., Qiao, Y., Murphy, L., Fallon, E., and A. Hanley, "SCTP Switchover Performance Issues in WLAN Environments", IEEE CCNC 2008, 1 2008.
- [GRINNEMO04] Grinnemo, K-J. and A. Brunstrom, "Performance of SCTP-controlled failovers in M3UA-based SIGTRAN networks", Advanced Simulation Technologies Conference , 4 2004.
- [IYENGAR06] Iyengar, J., Amer, P., and R. Stewart, "Concurrent Multipath Transfer using SCTP Multihoming over Independent End-to-end Paths.", IEEE/ACM Trans on Networking 14(5), 10 2006.
- [JUNGMAIER02] Jungmaier, A., Rathgeb, E., and M. Tuexen, "On the use of SCTP in failover scenarios", World Multiconference on Systemics, Cybernetics and Informatics , 7 2002.
- [NATARAJAN09] Natarajan, P., Ekiz, N., Amer, P., and R. Stewart, "Concurrent Multipath Transfer during Path Failure", Computer Communications , 5 2009.
- [RFC3873] Pastor, J. and M. Belinchon, "Stream Control Transmission Protocol (SCTP) Management Information Base (MIB)", RFC 3873, DOI 10.17487/RFC3873, September 2004, <<http://www.rfc-editor.org/info/rfc3873>>.
- [RFC6458] Stewart, R., Tuexen, M., Poon, K., Lei, P., and V. Yasevich, "Sockets API Extensions for the Stream Control Transmission Protocol (SCTP)", RFC 6458, December 2011.

## Appendix A. Discussions of Alternative Approaches

This section lists alternative approaches for the issues described in this document. Although these approaches do not require to update RFC4960, we do not recommend them from the reasons described below.

### A.1. Reduce Path.Max.Retrans (PMR)

Smaller values for Path.Max.Retrans shorten the failover duration and in fact this is recommended in some research results [JUNGMAIER02] [GRINNEMO04] [FALLON08]. However to significantly reduce the failover time it is required to go down (as with PFMR) to Path.Max.Retrans=0 and with this setting SCTP switches to another

destination address already on a single timeout which may result in spurious failover. Spurious failover is a problem in [RFC4960] SCTP as the transmission of HEARTBEATS on the left primary path, unlike in SCTP-PF, is governed by 'HB.interval' also during the failover process. 'HB.interval' is usually set in the order of seconds (recommended value is 30 seconds) and when the primary path becomes inactive, the next HEARTBEAT may be transmitted only many seconds later. Indeed as recommended, only 30 secs later. Meanwhile, the primary path may since long have recovered, if it needed recovery at all (indeed the failover could be truly spurious). In such situations, post failover, an endpoint is forced to wait in the order of many seconds before the endpoint can resume transmission on the primary path and furthermore once it returns on the primary path the CWND needs to be rebuild anew - a process which the throughput already have had to suffer from on the alternate path. Using a smaller value for 'HB.interval' might help this situation, but it would result in a general waste of bandwidth as such more frequent HEARTBEATING would take place also when there are no observed troubles. The bandwidth overhead may be diminished by having the ULP use a smaller 'HB.interval' only on the path which at any given time is set to be the primary path, but this adds complication in the ULP.

In addition, smaller Path.Max.Retrans values also affect the 'Association.Max.Retrans' value. When the SCTP association's error count exceeds Association.Max.Retrans threshold, the SCTP sender considers the peer endpoint unreachable and terminates the association. Section 8.2 in [RFC4960] recommends that Association.Max.Retrans value should not be larger than the summation of the Path.Max.Retrans of each of the destination addresses. Else the SCTP sender considers its peer reachable even when all destinations are INACTIVE and to avoid this dormant state operation, [RFC4960] SCTP implementation SHOULD reduce Association.Max.Retrans accordingly whenever it reduces Path.Max.Retrans. However, smaller Association.Max.Retrans value decreases the fault tolerance of SCTP as it increases the chances of association termination during minor congestion events.

#### A.2. Adjust RTO related parameters

As several research results indicate, we can also shorten the duration of failover process by adjusting RTO related parameters [JUNGMAIER02] [FALLON08]. During failover process, RTO keeps being doubled. However, if we can choose smaller value for RTO.max, we can stop the exponential growth of RTO at some point. Also, choosing smaller values for RTO.initial or RTO.min can contribute to keep the RTO value small.



Similar to reducing Path.Max.Retrans, the advantage of this approach is that it requires no modification to the current specification, although it needs to ignore several recommendations described in the Section 15 of [RFC4960]. However, this approach requires to have enough knowledge about the network characteristics between end points. Otherwise, it can introduce adverse side-effects such as spurious timeouts.

The significant issue with this approach, however, is that even if the RTO.max is lowered to an optimal low value, then as long as the Path.Max.Retrans is kept at the [RFC4960] recommended value, the reduction of the RTO.max doesn't reduce the failover time sufficiently enough to prevent severe performance degradation during failover.

#### Appendix B. Discussions for Path Bouncing Effect

The methods described in the document can accelerate the failover process. Hence, they might introduce the path bouncing effect where the sender keeps changing the data transmission path frequently. This sounds harmful to the data transfer, however several research results indicate that there is no serious problem with SCTP in terms of path bouncing effect [CARO04] [CARO05].

There are two main reasons for this. First, SCTP is basically designed for multipath communication, which means SCTP maintains all path related parameters (CWND, ssthresh, RTT, error count, etc) per each destination address. These parameters cannot be affected by path bouncing. In addition, when SCTP migrates the data transfer to another path, it starts with the minimal or the initial CWND. Hence, there is little chance for packet reordering or duplicating.

Second, even if all communication paths between the end-nodes share the same bottleneck, the SCTP-PF results in a behavior already allowed by [RFC4960].

#### Appendix C. SCTP-PF for SCTP Single-homed Operation

For a single-homed SCTP association the only tangible effect of the activation of SCTP-PF operation is enhanced failure detection in terms of potential notification of the PF state of the sole destination address as well as, for idle associations, more rapid entering, and notification, of inactive state of the destination address and more rapid end-point failure detection. It is believed that neither of these effects are harmful, provided adequate dormant state operation is implemented, and furthermore that they may be particularly useful for applications that deploys multiple SCTP associations for load balancing purposes. The early notification of

the PF state may be used for preventive measures as the entering of the PF state can be used as a warning of potential congestion. Depending on the PMR value, the aggressive HEARTBEAT transmission in PF state may speed up the end-point failure detection (exceed of AMR threshold on the sole path error counter) on idle associations in case where relatively large HB.interval value compared to RTO (e.g. 30secs) is used.

#### Authors' Addresses

Yoshifumi Nishida  
GE Global Research  
2623 Camino Ramon  
San Ramon, CA 94583  
USA

Email: nishida@wide.ad.jp

Preethi Natarajan  
Cisco Systems  
510 McCarthy Blvd  
Milpitas, CA 95035  
USA

Email: prenatar@cisco.com

Armando Caro  
BBN Technologies  
10 Moulton St.  
Cambridge, MA 02138  
USA

Email: acaro@bbn.com

Paul D. Amer  
University of Delaware  
Computer Science Department - 434 Smith Hall  
Newark, DE 19716-2586  
USA

Email: amer@udel.edu

Karen E. E. Nielsen  
Ericsson  
Kistavaegen 25  
Stockholm 164 80  
Sweden

Email: karen.nielsen@tieto.com

Network WG  
Internet-Draft  
Intended status: Proposed Standard  
Expires: January 4, 2015  
Updates: RFC 2205 & 4495 (if published as an RFC)

James Polk  
Subha Dhesikan  
Cisco  
July 4, 2014

Resource Reservation Protocol Multiple Instance Object  
draft-polk-rsvp-multi-instance-object-02.txt

Abstract

This document creates the framework for a new Resource Reservation Protocol version 1 (RSVP) object for instances in which there are multiple occurrences of existing RSVP objects is to be included within the same RSVP message. This document offers two instances for multiple versions of the same object will be valid in RSVP messages, for more than one traffic specification object (TSPEC), and more than one TSPEC priority object (PREEMPTION\_PRI).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 4, 2015.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	5
3. Framework for the MULTI_INSTANCE Object . . . . .	6
3.1 The MULTI_INSTANCE Object Format . . . . .	9
3.2 Rules for building a MULTI_INSTANCE Object . . . . .	9
4. Multiple TSPEC Objects in the MULTI_INSTANCE Object . . . . .	10
5. Multiple PREEMPTION_PRI Elements in the MULTI_INSTANCE Object	12
6. IANA Considerations . . . . .	15
7. Security Considerations . . . . .	16
8. Contributing Authors . . . . .	16
9. Acknowledgements . . . . .	17
10. References . . . . .	17
10.1 Normative References . . . . .	17
10.2 Informative References . . . . .	18
Author's Addresses . . . . .	18
Appendix . . . . .	18

## 1. Introduction

This document creates the framework for a new Resource Reservation Protocol version 1 (RSVP) [RFC2205] object for instances in which there is a need to carry multiple occurrences of included RSVP object within the same RSVP message. The need for multiple versions of existing objects is for environments in which the information conveyed within these objects may or may not be grantable by the network. To optimization this operation, if a different version of the same object, with different information or demands, can be included without the need for that rejecting entire RSVP message. For example, the initial RSVP PATH message contains a request for a 12Mbps bandwidth reservation, but that amount is not grantable by one or more network nodes. If a reduced amount of bandwidth can still be granted, and is acceptable to the network as well as both endpoints, allowing that PATH message to contain a backup bandwidth request for, say 4Mbps, saves the time of completely rejecting the initial PATH and having the sender generate a new PATH. A complete rejection to this scenario is how RSVP operates today.

This is a general purpose optimization for RSVP, and will allow any RSVP object to have multiple versions of an existing object, provided that existing object is specified to do so. It is important to understand that RSVP operates normally, with all Objects and elements in their native locations. This document offers two instances for multiple versions of the same object will be valid in RSVP messages, for more than one traffic specification object (TSPEC) [RFC2210], and more than one priority element (PREEMPTION\_PRI) [RFC3181]. This extension will bring RSVP more in line with existing application layer protocols that offer multiple choices for the specifics within a call or session. At the same

time, all extra versions of Objects or elements are contained in a single location that is ignored if not understood. Thus, backwards compatibility is assured.

Realtime session set-up protocols such as SIP [RFC3261] carry a Session Description Protocol (SDP) [RFC4566] payload which establishes the parameters for rich media calls (i.e., voice, video) between two or more endpoints. Since the late 1990s, SDP has had the capability to offer more than one codec per application type (i.e., more than one audio payload type and/or more than one video payload type), which can be carried in the same session set-up message. This means a calling endpoint can give the called party a list of codecs to choose from for that call, as well as multiple applications for that call.

With this RSVP extension, for example, a SIP voice and/or video call can have a reservation adapt to whichever codec(s) are picked for that call, without wasting unnecessary bandwidth that will not be utilized.

Visually, Figure 1. is a normal RSVP reservation set-up exchange that is accepted by all RSVP enabled nodes.

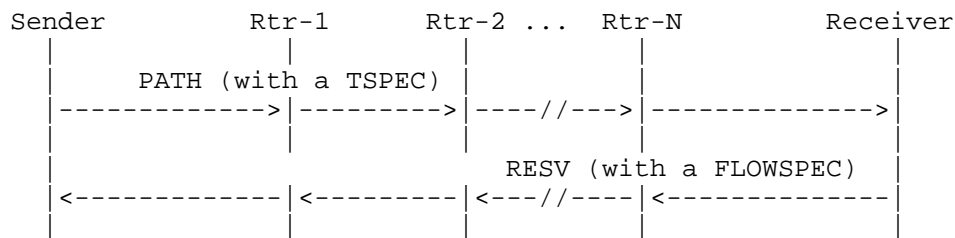


Figure 1. Concept of RSVP in a Single Direction

However, Figure 2. is a normal RSVP reservation set-up exchange that is rejected as the reservation is partially established. We will use bandwidth as the reason for the rejection because it is probably the easiest thing to understand about RSVP, that it creates reservation of fixed bandwidth.

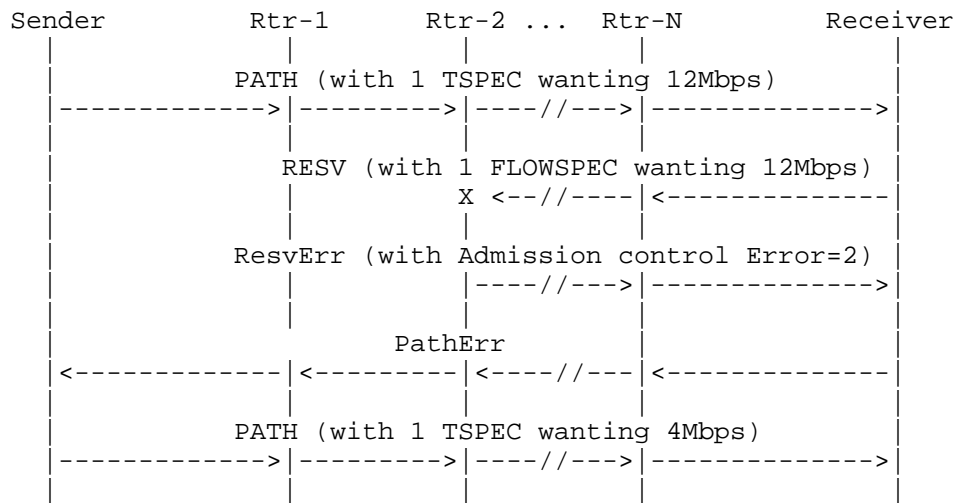


Figure 2. Concept of RSVP Rejection due to Limited Bandwidth

Rtr-2 in the above example reservation attempt rejects the bandwidth requested for this reservation. Once the Sender receives the PathErr message indicating why the rejection occurred, it can attempt a new reservation requesting less bandwidth. Regrettably, this is a bit of a guessing game put on the Sender to figure out how much bandwidth to request next. When this scenario is complicated when the reservation request is initiated because of a layer 7 signaling protocol, such as SIP, to establish a call between two endpoints, as defined in [RFC3312]. All the users experience is further delay as RSVP attempts to successfully establish the reservation before "the phone can ring".

Presently, translating a (SIP) layer 7 operation into RSVP at layer 4, only a single reservation can be established per application (i.e., one for voice, one for video) at a time (without creating chaos). This one reservation request would most probably its bandwidth request using the codec with the largest bandwidth requirements. Bandwidth parameters are conveyed within a traffic specification (TSPEC), as defined in [RFC2210]. Once one considers the bandwidth needs of present day video codecs, always initially setting up the maximum bandwidth reservation is less than optimal (some might argue criminal).

If, on the other hand, the initial RSVP PATH message could contain more than one version of a TSPEC, say one per codec. Then the reservation would be established with the greatest amount of bandwidth the network could grant at its most congested node in the signaling path, which would in turn choose for the endpoints which codec within SDP is selected for this call.

Thus, this extension would solve the problem in Figure 2. in this way,

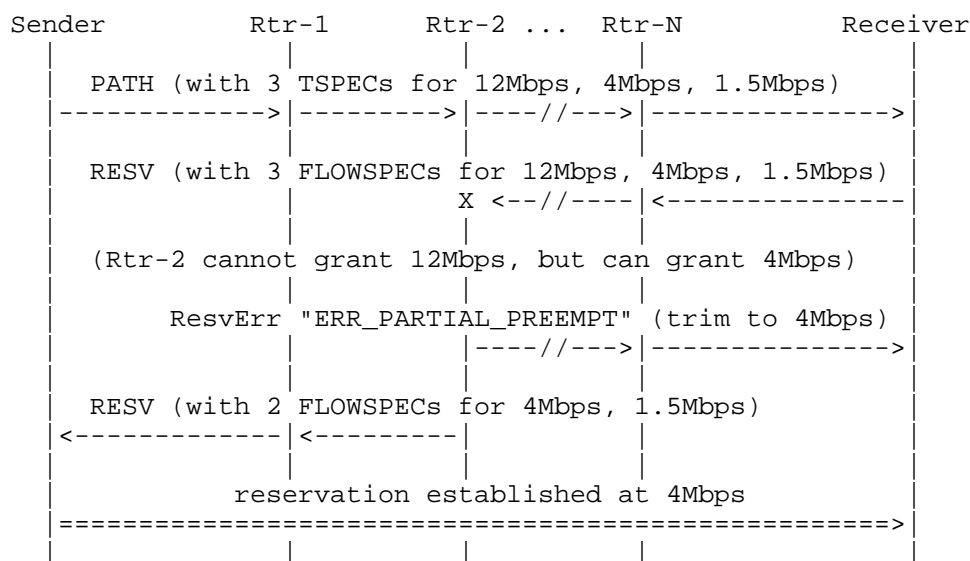


Figure 3. Concept of RSVP Rejection due to Limited Bandwidth

Figure 3. shows a RSVP PATH message containing 3 TSPECs (12Mbps, 4Mbps, and 1.5Mbps), placed in that order in the PATH. Router 2 (Rtr-2) cannot grant the RESV message at 12Mbps, but can grant the 4Mbps bandwidth request. Rtr-2 trims the bandwidth upstream with a slight modification to the procedures defined in RFC 4495, and transmits the RESV downstream without the 12Mbps bandwidth request, which was removed from the RESV. The Sender, in this example, receives the RESV with 4Mbps and the reservation is established.

In Section 3., we will create the framework and format for the MULTI\_INSTANCE Object. In Section 4., we will show how to include multiple TSPEC Objects within this MULTI\_INSTANCE Object, as well as stipulate the rules for TSPEC usage. In Section 5., we will show how to include multiple PREEMPTION\_PRI Objects within this MULTI\_INSTANCE Object, as well as stipulate the rules for PREEMPTION\_PRI usage. Section 6. will have the IANA registry considerations, and Section 7. will have the Security considerations.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.



### 3. Framework for the MULTI\_INSTANCE Object

The format of all RSVP Objects is based on a series of 32-bit words. This is true with the MULTI\_INSTANCE Object as well. Normally, RSVP and IntServ documents specify Objects in a 32-bit wide, top down format, where the most significant bit is the top left bit, and the least significant bit on the right. This is loosely shown in Figure 4. below.

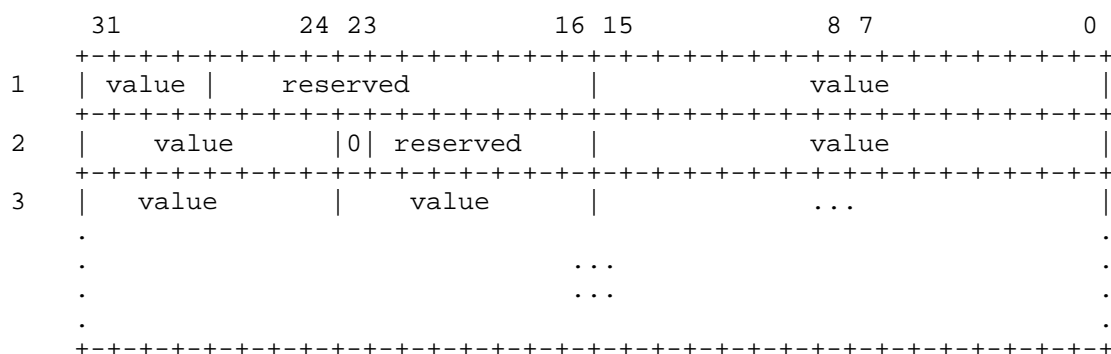


Figure 4. Generic RSVP Format for Illustration Purposes

The individual field value lengths within each RSVP Object depend on the Object, thus Figure 4. is merely an example (which happens to be the first 3 words format of a TSPEC).

However, RSVP messages can be quite long in this format, so what one usually sees in documents are each individual Object and no overall RSVP message format. Each Object has an field identifier indicating which RSVP message this Object is within (e.g., PATH, RESV, REFRESH), as well as a 'Parameter ID' indicating which Object within this (e.g., TSPEC, Rspec, Policy\_Data).

Looking at RSVP another way, to illustrate the point about where certain parts can be within an overall RSVP message, Figure 5. Shows an example RSVP message on its side, where the top of the message is on the left and the bottom of the message is on the right. With that in mind, the most significant bit of the top 32-bit word is on the lower left of Figure 5., and the least significant bit is on the upper left. The length of this message can vary and does not represent anything other than this message has some size to it; i.e., it has a number of Objects within this message, including a sender\_descriptor where the 'primary' TSPEC Object resides, and Policy\_Data Object where the PREEMPTION\_PRI Object resides. Additionally in the RSVP message is the proposed MULTI\_INSTANCE Object, which is neither in the sender\_descriptor or Policy\_Data Object.

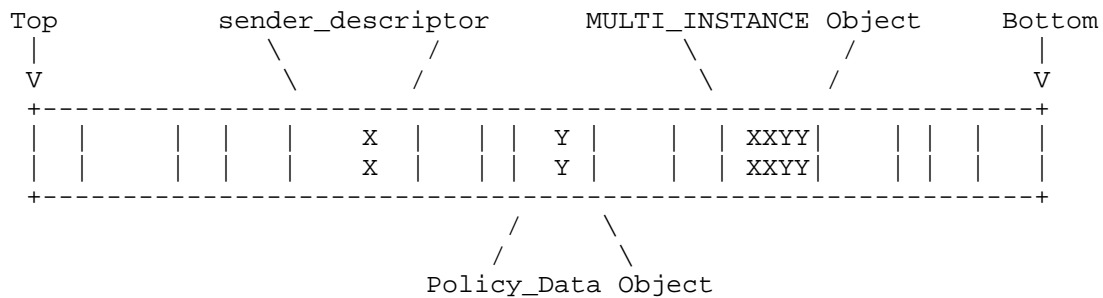


Figure 5. Generic RSVP Format Shown Sideways

It is important to remember that RSVP enabled nodes will always ignore Objects that are not understood. This allows the protocol to be extended before all the RSVP nodes are upgraded to understand new functions and capabilities. In other words, no one expects a single 'flag day' upgrade to occur in all routers at the same time in the same network, which could be disruptive if not performed correctly.

An important aspect of this new Object is that the initial copy or instance of the Object, however many Objects have multiple instances in this RSVP message, MUST remain in its original place within the message. We will refer to this original version of an Object or element to be the 'primary' version or copy. Its placement allows RSVP to operate normally. The MULTI\_INSTANCE Object only carries a second, third, etc. versions of Objects. Once the RSVP node determines that it cannot grant what is asked for in an existing Object, it will look to the MULTI\_INSTANCE Object for the next instance of that Object to replace the original with. Failing this, the RSVP message will mostly likely be rejected through the normal procedures already defined in RSVP documentation.

To give a practical example of this, we will use the message flow from Figure 3. In it, we have a PATH message carrying not one, but three TSPECs for 12Mbps, 4Mbps and 1.5Mbps. Once Rtr-2 cannot grant the primary TSPEC asking for 12Mbps, that router discards that TSPEC, from the RSVP message. It knows to look into the MULTI\_INSTANCE Object for a second version of the TSPEC. Finding two (4Mbps and 1.5Mbps), Rtr-2 moves the 4Mbps TSPEC completely into the sender\_descriptor as the new 'primary' TSPEC and attempts to establish the reservation at 4Mbps. In this case, 4Mbps is granted and transmits the RESV upstream towards Rtr-1 with only one remaining TSPEC in the MULTI\_INSTANCE Object.

[EDITOR'S NOTE: It is important to state that, so far, we have not defined where this MULTI\_INSTANCE Object goes within the RSVP message.]

The MULTI\_INSTANCE Object can have a number of versions of the same Object that is within the RSVP message, as well as can have more than one different type of Object. To this end, here is the proposed

generic format for the MULTI\_INSTANCE Object that is only carrying a single other Object

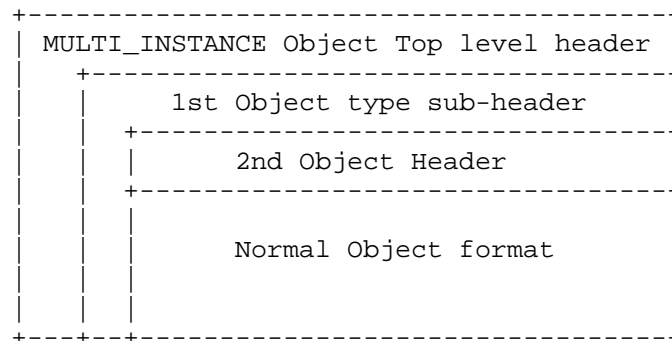


Figure 6. MULTI\_INSTANCE with 1 Object

Figure 6. shows a complete second version of an existing RSVP Object, which can be removed and copied bit-for-bit into the normal placement of this Object within the RSVP message. It is important to note that this MUST be a complete new copy of a valid Object.

Figure 7. shows a MULTI\_INSTANCE Object with a second and third version of the same RSVP Object

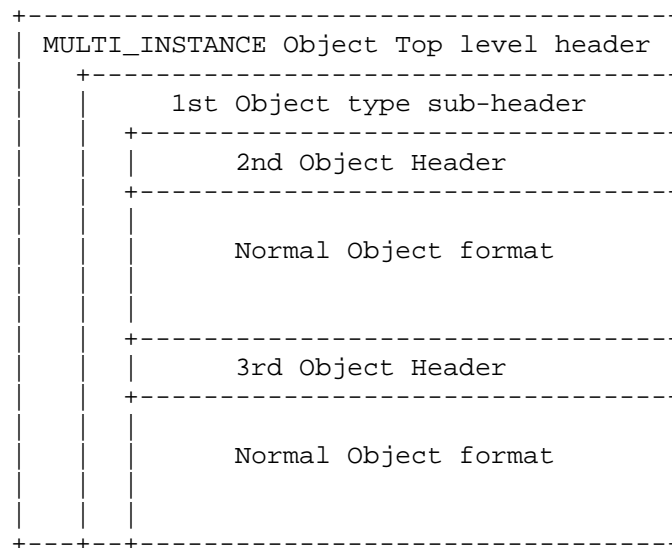


Figure 7. MULTI\_INSTANCE with 2 Objects

Again, version 2 and 3 are completely valid versions of the RSVP Object they are meant to replace, with no change in any value allowed.

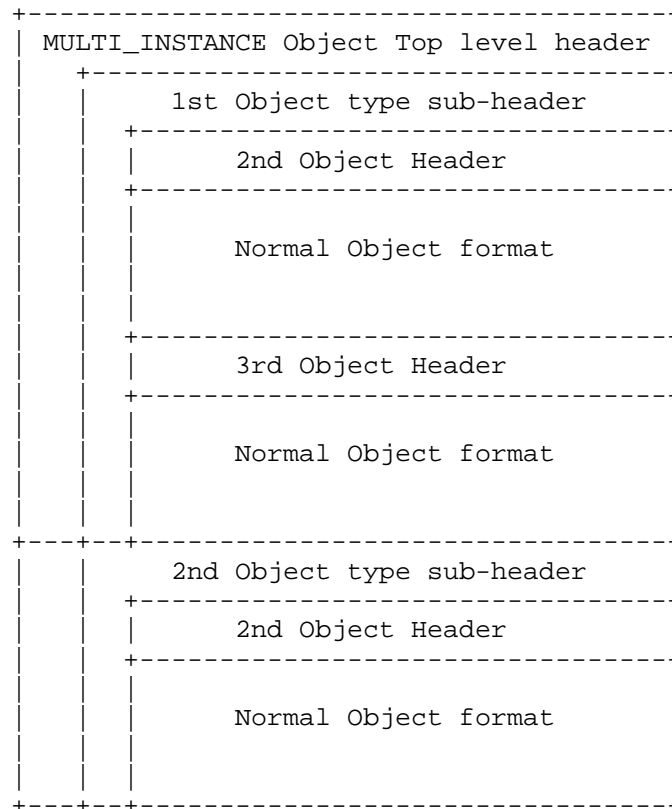


Figure 8. MULTI\_INSTANCE with 2 Objects

Figure 8. adds a second type of Object to the MULTI\_INSTANCE Object that is shown in Figure 7. To be able to add another type of Object, and not a second copy of the same Object, a new Object Type header is REQUIRED to preface the first 32-bit word of the new Object. These two different Objects carried within the MULTI\_INSTANCE Object can be related, or they might not have anything to do with each other.

### 3.1 The MULTI\_INSTANCE Object Format

The multi-32-bit word format of the MULTI\_INSTANCE Object is TBD in a subsequent revision of this document.

### 3.2 Rules for building a MULTI\_INSTANCE Object

The following are the rules for implementations of the MULTI\_INSTANCE object:

- #1 - having only 1 \*SPEC or Object is allowed in the MULTI\_INSTANCE

Object (i.e., a grouping can have a single entry)

- #2 - more than one \*SPEC or Object is allowed in the MULTI\_INSTANCE Object (i.e., separate groups can have a single entry each)
- #3 - more than one \*SPEC or Object is allowed in the MULTI\_INSTANCE Object (i.e., separate groups can have a multiple entries each)
- #4 - some groupings within MULTI\_INSTANCE MUST be paired in whenever a single instance occurs in any group.

In other words, based on rule #3, if a TSPEC is in each group, so MUST there be an RSPEC if any RSPEC is within this MULTI\_INSTANCE Object. An RSPEC is an example of a \*SPEC that MUST NOT be alone without its TSPEC.

#### 4. Multiple TSPEC Objects in the MULTI\_INSTANCE Object

This document defines the framework for the MULTI\_INSTANCE Object, as well as for two Objects to be available for inclusion within this new Object: the TSPEC Object and the PREEMPTION\_PRI Object (detailed in Section 5.). This section deals with how to include one or more TSPEC Objects within the MULTI\_INSTANCE Object.

This document specifies if the reservation is to be Controlled Load [RFC2211], the entire TSPEC, including the two 32-bit word headers (totaling eight 32-bit words), are included in the MULTI\_INSTANCE object. An example of a TSPEC from RFC 2210 is here in Figure 9.:

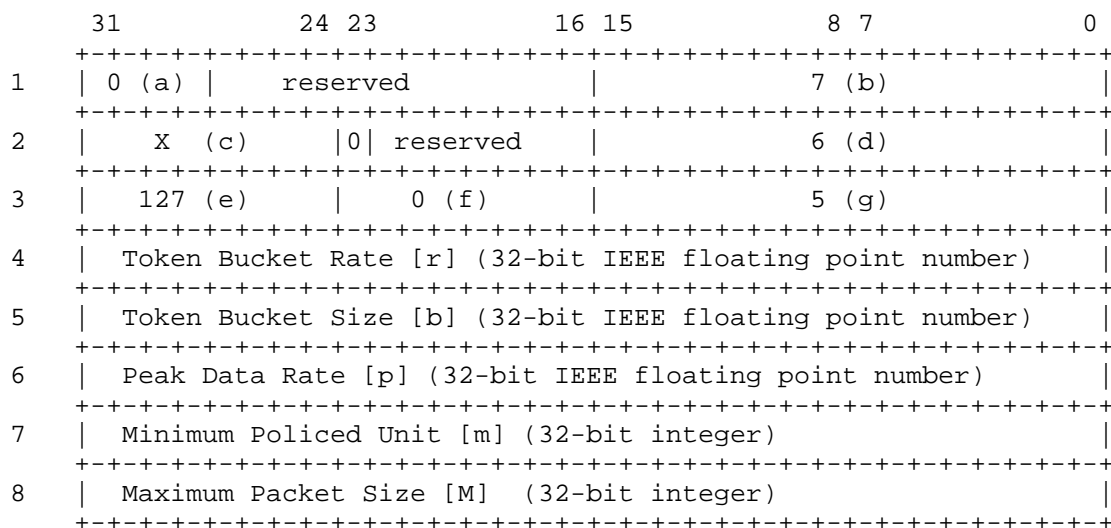


Figure 9. Controlled Load SENDER\_TSPEC in a PATH

- (a) - Message format version number (0)
- (b) - Overall length (7 words not including header)
- (c) - Service header, service number
  - '1' (Generic information) if in a PATH message;
- (d) - Length of service data, 6 words not including per-service header
- (e) - Parameter ID, parameter 127 (Token Bucket TSPEC)
- (f) - Parameter 127 flags (none set)
- (g) - Parameter 127 length, 5 words not including per-service header

This document specifies if the reservation is to be Guaranteed Service, the entire TSPEC and RSPEC, including the two 32-bit word headers (totaling eleven 32-bit words), are included in the MULTI\_INSTANCE object as a single consecutive chunk.

A request guaranteed service reservation contains a TSPEC and RSPEC [RFC2215], as shown in Figure 10.:

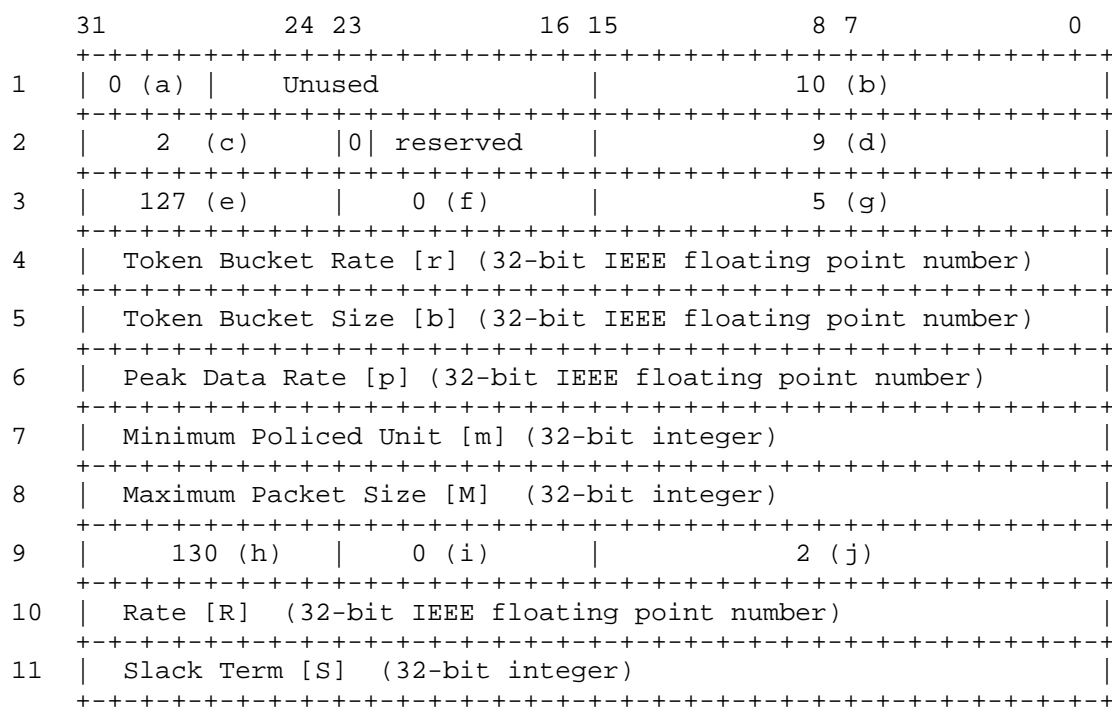


Figure 10. Guaranteed Service SENDER\_TSPEC in a PATH

- (a) - Message format version number (0)
- (b) - Overall length (9 words not including header)
- (c) - Service header, service number 2 (Guaranteed)
- (d) - Length of per-service data, 9 words not including per-service header

- (e) - Parameter ID, parameter 127 (Token Bucket TSpec)
- (f) - Parameter 127 flags (none set)
- (g) - Parameter 127 length, 5 words not including parameter header
- (h) - Parameter ID, parameter 130 (Guaranteed Service RSpec)
- (i) - Parameter xxx flags (none set)
- (j) - Parameter xxx length, 2 words not including parameter header

The difference in structure between the Controlled-Load FLOWSPEC and Guaranteed FLOWSPEC is the RSPEC, defined in [RFC2212]. The difference with respect to the MULTI\_INSTANCE Object is found in the first 32-bit word, value 'b' above - the TSpec Object overall length. This will tell a node whether it is a Controlled Load or Guaranteed Service TSpec.

As a reminder, TSPECs contained in the MULTI\_INSTANCE Object MUST NOT be altered when moved from the MULTI\_INSTANCE Object to the sender\_descriptor or FLOWSPEC. Generically, this needs to be a simple cut and paste operation.

If there are multiple TSPECs in the MULTI\_INSTANCE Object, each MUST be the same type of TSpec. In other words, there MUST NOT be a mix of Controlled Load with Guaranteed Service TSPECs in the same MULTI\_INSTANCE Object.

RFC 4495 defines how existing reservations can partially preempt (trim) the agreed upon bandwidth assigned to an existing reservation. This specification extends RFC 4495 by allowing that trimming of bandwidth assigned to a reservation to occur during reservation establishment downstream. This occurs when a node upstream cannot grant the bandwidth already granted downstream, but that upstream node can grant a reduced amount of bandwidth from another TSpec within FLOWSPEC, from within the MULTI\_INSTANCE Object. This operation is shown in Figure 3.

## 5. Multiple PREEMPTION\_PRI Elements in the MULTI\_INSTANCE Object

The order of the TSPECs within the MULTI\_INSTANCE Object is one way to determine which is the next TSpec to be processed by a router. Another way of determining which TSpec is the next one to be processed is by allowing the dynamic bandwidth selection to reflect a different reservation priority for each of the multiple "bandwidth" associated with a reservation.

[RFC2750] presents a set of extensions for supporting generic policy based admission control in RSVP. These extensions include the standard format of POLICY\_DATA objects, and a description of RSVP's handling of policy events. These extensions are consistent with the framework for policy-based admission control presented in [RFC2753]. POLICY\_DATA objects are carried by RSVP messages and contain policy information. The exchange of POLICY\_DATA objects between policy-capable nodes along the data path, supports the generation

and enforcement of consistent end-to-end admission control policies.

POLICY\_DATA objects contain a list of Policy Elements that each contain a single unit of information necessary for the evaluation of policy rules. Multiple policy elements are already specified. For example, [RFC2872] specifies the Application and Sub Application Identity policy element for use with RSVP.

[RFC3181] specifies another policy element, the Preemption Priority Policy Element, that can be signaled in RSVP so that network node may take into account this policy element in order to preempt some previously admitted low priority sessions in order to make room for a newer, higher priority session. The Preemption Priority Policy Element (PREEMPTION\_PRI) contains:

- o one Preemption Priority specifying the priority of the new flow compared with the defending priority of previously admitted flows.
- o one Defending Priority that is used once this reservation is established to compare with the preemption priority of new flows.

The format of preemption priority policy element (copied from RFC 3181) is as follows:

+-----+-----+-----+-----+			
Length (12)		P-Type = PREEMPTION_PRI	
+-----+-----+-----+-----+			
Flags	M. Strategy	Error Code	Reserved(0)
+-----+-----+-----+-----+			
Preemption Priority		Defending Priority	
+-----+-----+-----+-----+			

Figure 11. Preemption Priority Policy Element Format

Length: 16 bits

Always 12. The overall length of the policy element, in bytes.

P-Type: 16 bits

PREEMPTION\_PRI = 1

This value is registered with IANA, see Section 7.

Flags: 8 bits

Reserved (always 0).

Merge Strategy: 8 bit

- 1 Take priority of highest QoS: recommended
- 2 Take highest priority: aggressive
- 3 Force Error on heterogeneous merge



Reserved: 8 bits

Error code: 8 bits

- |   |               |  |
|---|---------------|--|
| 0 | NO_ERROR      | Value used for regular PREEMPTION_PRI elements |
| 1 | PREEMPTION    | This previously admitted flow was preempted    |
| 2 | HETEROGENEOUS | This element encountered heterogeneous merge   |

Reserved: 8 bits

Always 0.

Preemption Priority: 16 bit (unsigned)

The priority of the new flow compared with the defending priority of previously admitted flows. Higher values represent higher Priority.

Defending Priority: 16 bits (unsigned)

Once a flow was admitted, the preemption priority becomes irrelevant. Instead, its defending priority is used to compare with the preemption priority of new flows.

For any specific flow, its preemption priority MUST always be less than or equal to the defending priority.

The preemption priority and defending priority of the Preemption Priority Policy Element carried in a RESV message MUST be associated with the flow specification carried in the FLOWSPEC object. There MUST be either no Preemption Priority Policy Element carried in the MULTI\_INSTANCE Object, or there needs to be the exact same number of Preemption Priority Policy Element as there are TSPEC Objects. This MUST be a one-to-one mapping of numbers. For example, the preemption priority and defending priority of the first (respectively second) sub-element (when present) of the MULTI\_INSTANCE Object is to be associated with the first (respectively second) flow specification (when present) in the MULTI\_INSTANCE Object.

If a RESV message contains a dissimilar number of TSPECs than Preemption Priority Policy Elements in the MULTI\_INSTANCE object, but contains a Preemption Priority Policy Element in the POLICY\_DATA object, then the Preemption Priority Policy Element in the MULTI\_INSTANCE object MUST be ignored, and all TSPECs retain the priority properties of the Preemption Priority Policy Element in the POLICY\_DATA object.

An example MULTI\_INSTANCE Object with 2 TSPEC Objects and 2 Preemption Priority Policy Element is showing generically in Figure 12.

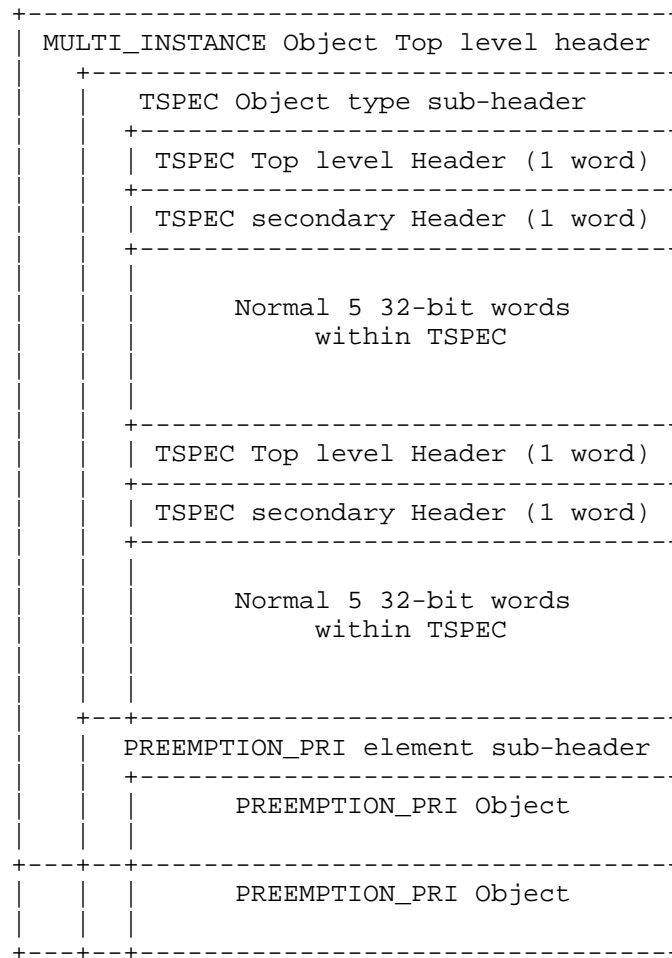


Figure 12. MULTI\_INSTANCE with 2 TSPECs and 2 PREEMPTION\_PRIs

## 6. IANA Considerations

This document IANA registers the following new parameter name in the rsvp-parameters assignments at [IANA]:

Registry Name: Parameter Names

Registry:

Value	Description	Reference
125	Multiple_Instance_object	[RFCXXXX]

Where RFCXXXX is replaced with the RFC number assigned to this Document.

This document IANA registers the following new error subcode in the

Error code section, under the Admission Control Failure (error=1), of the rsvp-parameters assignments at [IANA]:

Registry Name: Error Codes and Globally-Defined Error Value  
Sub-Codes

Registry:

"Admission Control  
Failure"

Error Subcode	meaning	Reference
6	= MULTI_INSTANCE bandwidth unavailable	[RFCXXXX]

## 7. Security Considerations

The security considerations for this document do not exceed what is already in RFC 2205 (RSVP), as nothing in either of those documents prevent a node from requesting a lot of bandwidth in a single TSPEC, or what priority values are given in a Preemption Priority Policy Element. This document merely reduces the signaling traffic load on the network by allowing many requests that fall under the same policy controls to be included in a single round-trip message exchange.

Further, this document does not increase the security risk(s) to that defined in RFC 4495, where this document creates additional meaning to the RFC 4495 created error code 102.

A misbehaving Sender can include too many TSPECs in the MULTI\_INSTANCE object, which can lead to an amplification attack. That said, a bad implementation can create a reservation for each TSPEC received from within the RESV message. The number of TSPECs in the new MULTI\_INSTANCE object is limited, and the spec clearly states that only a single reservation is to be set up per RESV message.

To ensure the integrity of RSVP, the RSVP Authentication mechanisms defined in [RFC2747] and [RFC3097] SHOULD be used. Those protect RSVP message integrity hop-by-hop and provide node authentication as well as replay protection, thereby protecting against corruption and spoofing of RSVP messages.

## 8. Contributing Authors

The authors here would like to thank the authors of draft-lefaucheur-tsvwg-rsvp-multiple-preemption-02 for allowing that draft to be merged with this draft, specifically for the Preemption Priority Policy Element discussion in Section 5. They are:

Francois Le Faucheur  
Arun Kudur and  
Ashok Narayanan

## 9. Acknowledgements

The authors wish to thank Fred Baker, Joe Touch, Bruce Davie, Dave Oran, Ashok Narayanan, Lou Berger, Lars Eggert, Arun Kudur, Janet Gunn and Ken Carlberg for their helpful comments and guidance in this effort.

## 10. References

### 10.1 Normative References

- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997
- [RFC2205] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997
- [RFC2210] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997
- [RFC2211] J. Wroclawski, "Specification of the Controlled-Load Network Element Service ", RFC 2211, September 1997
- [RFC2212] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997
- [RFC2215] S. Shenker, J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", RFC 2212, September 1997
- [RFC2747] F. Baker, B. Lindell, M. Talwar, " RSVP Cryptographic Authentication", RFC2747, January 2000
- [RFC2750] S. Herzog, "RSVP Extensions for Policy Control", RFC 2750, January 2000
- [RFC2753] R. Yavatkar, D. Pendarakis, R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000
- [RFC2872] Y. Bernet, R. Pabbati, "Application and Sub Application Identity Policy Element for Use with RSVP", RFC 2872, June 2000
- [RFC3097] R. Braden, L. Zhang, "RSVP Cryptographic Authentication -- Updated Message Type Value", RFC 3097, April 2001
- [RFC3181] S. Herzog, "Signaled Preemption Priority Policy Element", RFC 3181, October 2001

- [RFC3261] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, May 2002.
- [RFC3312] G. Camarillo, Ed., W. Marshall, Ed., J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)", RFC 3312 Preconditions, October 2002
- [RFC4495] J. Polk, S. Dhesikan, "A Resource Reservation Protocol (RSVP) Extension for the Reduction of Bandwidth of a Reservation Flow", RFC 4495, May 2006
- [RFC4566] M. Handley, V. Jacobson, C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006

## 10.2 Informative References

draft-lefaucheur-tsvwg-rsvp-multiple-preemption

draft-ietf-tsvwg-intserv-multiple-tspec

## Author's Addresses

James Polk  
3913 Treemont Circle  
Colleyville, Texas, USA  
+1.817.271.3552

mailto: jmpolk@cisco.com

Subha Dhesikan  
Cisco Systems  
170 W. Tasman Drive  
San Jose, CA 95134 USA

mailto: sdhesika@cisco.com

## Appendix A - History

This history of how we got to this phase of the development in this document can be traced to the work and choices articulated in the appendix within

draft-ietf-tsvwg-intserv-multiple-tspec

From there, another team developed

draft-lefaucheur-tsvwg-rsvp-multiple-preemption

Then Lou Berger (yeah, blame him!) came up with the bright idea of

combining the two efforts in such a way that one can take a complete Object or element, and replace the primary instance of that within an RSVP message for whatever reason (perhaps the message would be rejected if that piece was not replaced with more reasonable demands on the network; who knows). Anyway, that's how we got here. That's the story and I'm sticking to it... :-p

Network WG  
Internet-Draft  
Intended status: Informational  
Expires: January 9, 2012

James Polk  
Cisco  
July 9, 2012

The Problem Statement for the Standard  
Configuration of DiffServ Service Classes  
draft-polk-tsvwg-diffserv-stds-problem-statement-00.txt

## Abstract

This document describes the problem statement on two recently proposed expansions to DiffServ. The first of these expansions proposes updating the informational RFC 4594 document to standards track status, while making the necessary changes to make it current; for example, creating more granular traffic treatments, some with new Per Hop Behaviors (PHB). The second proposal defines 6 new DiffServ Codepoints necessary from these new PHBs in the proposal within the first draft.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2012.

## Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Brief Overview of RFC 4594 and RFC 5127 . . . . .	3
2.1 Brief Overview of RFC 4594 . . . . .	3
2.2 Brief Overview of RFC 5127 . . . . .	4
3. Brief Discussion of the RFC 4594 Update Draft . . . . .	5
4. Conclusion and What's Next . . . . .	7
5. Acknowledgements . . . . .	7
6. IANA Considerations . . . . .	7
7. Security Considerations . . . . .	8
8. References . . . . .	8
8.1 Normative References . . . . .	8
8.2 Informative References . . . . .	8
Author's Address . . . . .	9

## 1. Introduction

Differentiated Services (DiffServ) [RFC2474] creates an IP header marking or indicator with which intermediate nodes (i.e., routers and switches) can make policy decisions. These 6-bit values are called Differentiated Services Codepoint Point (DSCP) values. DSCP values are used to differentiate packet treatment within an intermediate node, not across a network, as the conditions affecting that marking are different within each node. This is called Per Hop Behavior (PHB). In other words, even though a packet has the same DSCP from source to destination, it can and often does experience different treatment depending on the conditions of the nodes it traverses on its journey.

The DiffServ architecture allows for DSCP values within a packet to be changed, or remarked, any number of times. In other words, a packet can have its DSCP remarked at every layer-3 hop throughout the life of that packet. This practice actually occurs infrequently, but it is allowed.

At issue is a combination of the number of networks or endpoints that are choosing to use DiffServ markings, and the number of administrative domains (called "networks" in this document) a packet traverses with different policies for how packet flows of a similar type (e.g., a voice flow, or an email flow, etc.) are to be marked.

The community presently has RFC 4594 [RFC4594], which is an informational guideline on how networks can or should mark certain packet flows with differing traffic characteristics using DiffServ. There are several reasons why this informational RFC lacks the necessary clarity and strength to reach widespread adoption:

- o confusion between RFC 4594 and RFC 5127 [RFC5127], the latter of which is for aggregating many 6-bit DSCP values into a 3-bit (8



value) field used specifically by service provider (SP) networks.

- o some believe both RFCs are for SPs, while others ignore RFC 5127 and use RFC 4594 as if it were standards track or BCP.
- o some believe RFC 5127 is for SPs only, and want RFC 4594 to reduce the number of DSCPs within its guidelines to recommend using only 3 or 4 DSCPs. This seems to stem from a manageability and operational perspective.
- o some know RFC 4594 is informational and do not follow its guidelines specifically because it is informational.
- o some use DSCP values that are not defined within RFC 4594, making mapping between different networks using similar or identical application flows difficult.
- o some believe enterprise networks should not use either RFC except at the edge of their networks, where they directly connect to SP networks.
- o some argue that the services classes guidance per class is too broad and are therefore not sure in which service class a particular application is to reside.

This document is not intended to reach RFC status. Rather, it is to stimulate discussion on both RFC 4594 and 5127 to lessen existing confusion within the community. It should be noted that RFC 4594 has an offered update within TSVWG [ID-4594-UPDATE]. This draft has created some heated discussions within that WG before and during the Paris IETF meeting.

First, we'll discuss briefly RFCs 4594 and 5127 in Section 2. Then we will discuss what the update to RFC 4594 proposes differently and what we expect to happen to RFC 5127 in Section 3.

## 2. Brief Overview of RFC 4594 and RFC 5127

### 2.1 Brief Overview of RFC 4594

Essentially, RFC 4594 is a guideline for how to choose which DSCP to use based on the traffic characteristics an application flow needs to experience within a network for optimal performance. RFC 4594 specifically points to several existing standards-track DiffServ RFCs to augment the text in each of those RFCs, without violating any of the rules within each of those documents. RFC 4594:

- o painstakingly lays out definitions and guidelines for each service class.
- o clearly indicates each service class's tolerance to delay, jitter

and packet loss.

- o details the conditioning treatments at the Differentiated Services (DS) edge.
- o categorizes traffic characteristics into 12 service classes utilizing one or more DSCPs:

Network Control	Broadcast Video
Telephony	Low-Latency Data
Signaling	OAM
Multimedia Conferencing	High-throughput Data
Realtime Interactive	Standard
Multimedia Streaming	Low-priority Data

## 2.2 Brief Overview of RFC 5127

At its barest, RFC 5127 recommends that, of the many service classes described within RFC 4594, each having different traffic characteristics, similar service classes be grouped or aggregated into 3, 4, or 5 markings for SP traversal. This limitation of the number of individual service classes is partly to reduce the number of separate distinctions traversing over their network because SPs have difficulty managing what is deemed 'too many' different classes. Another part for this reduction is customer expectations of meeting contractual Service Level Agreements (SLAs).

To this end, and perhaps because of it, MPLS was designed with only 8 values of priority differentiation, i.e., the 3 EXP bits. To be fair, LAN based IEEE has only a 3-bit priority field as well within its specifications, known as the Priority Code Point (PCP), as part of the 802.1Q header spec. IEEE 802.1e, which defines QoS over Wi-Fi, also only defines 8 levels (called User Priority or UP codes).

The result is to have the IETF within RFC 5127 recommend the following (which is Figure 2 within that RFC):

Treatment Aggregate Behavior	Treatment Aggregate Behavior	DSCP
Network Control	CS (RFC 2474)	CS6
Real-Time	EF (RFC 3246)	EF, CS5, AF41, AF42, AF43, CS4, CS3
Assured Elastic	AF (RFC 2597)	CS2, AF31, AF21, AF11
		AF32, AF22, AF12
		AF33, AF23, AF13
Elastic	Default (RFC 2474)	Default, (CS0)
		CS1

Figure 1: Treatment Aggregate Behavior

RFC 5127 goes on to recommend the marking and treatments on either side of the provider edge remain the same. In other words, the DSCP values remain the same and are used to determine which queue to place the packets into within the aggregates, where the packets are treated the same within that tunnel until the egress provider edge.

Many within enterprise networks do not pay attention to what RFC 5127 says because they are sufficiently removed from dealing with the constraints of very few DSCP values or the need to aggregate DSCP values into groups.

### 3. Brief Discussion of the RFC 4594 Update Draft

The RFC 4594 update draft [ID-4594-UPDATE] proposes to update what has occurred since RFC 4594 was written (i.e., 2006), in which more granular service classes can be differentiated by application requirements. For example, Figure 2 within RFC 4594 identifies "Telephony" as having 'Fixed-size small packets'. That is not true for today's video flow, therefore it needs to be modified. The update draft currently breaks out audio and video separately to reflect this different, as well as the ability to treat each traffic type differently within a network. Another example is gaming and TCP. The two were believed by most, and it is still believed by many that gaming requires a UDP delivery due to the requirements for timely delivery of packets and that retransmissions would cause delays and bad things to happen to gaming applications. This was proved false within [ID-TCMTF], in which the author of that document

had a presentation showing TCP was used and viable.

[RFC5865] created a new Expedited Forwarding (EF) DSCP value called VOICE-ADMIT, the second time an application is identified within the DiffServ realm. The first was the service class Broadcast Video, which is poorly used within RFC 4594 because other types of flows can be 'broadcast' other than video, such as audio. From this, [ID-4594-UPDATE] moved in two directions:

- o it called out two service classes (audio and video), even though audio and video packets are not the only types of packets within each traffic characteristic.
- o it removed "Video" from the Broadcast service class name.

From the resistance to this proposal within [ID-4594-UPDATE], perhaps other service class label names should be used.

The draft also recognizes the differences in video traffic, even though it is always carried over RTP [RFC3550]. Aside from silence suppression, video traffic varies far more than audio traffic. For example, video is

- o far more variable in bandwidth utilization within the same flow.
- o far more variable in packet size.
- o at different business priorities in some networks based on a configuration. For example, desktop video often is of less important than Telepresence video on the same network. Lacking congestion, the two are treated the same. When congestion exists, one is given priority over the other.

Consequently any service class that contains video needs to account for larger packet size variation than audio, which was equally true in 2006, but not contained in RFC 4594.

Further, with the publication of RFC 5865, the concept of 'capacity admitted' traffic flows have been defined within DiffServ, and are being expanded with the proposal within this new draft [ID-NEW-DSCPS]. There are differing opinions as to whether the realtime Treatment Aggregate in Figure 1 above should also contain these capacity admitted flows, or if 'capacity admitted' traffic flows should have their own Treatment Aggregate containing all realtime capacity admitted traffic. Mixing capacity admitted traffic with unbounded realtime traffic seems to be trouble from a predictability point of view within routers believing they individually understand exactly how much traffic will be traversing each interface and at what rate.

All this said, there is a valid argument to constrain or prevent any DSCP value from being assigned to a single application, mostly due

to the limitation of the overall number of DSCP values available for use. [ID-4594-UPDATE] provides at least several applications per service class (or DSCP); a fact many have overlooked to date.

[ID-4594-UPDATE] is not only about or because of realtime traffic. It is also an overall update to the ideas and guidelines within RFC 4594, with the intent to make that document a standards track document for interoperability purposes.

#### 4. Conclusion and What's Next

Without attempting to fundamentally change the guidelines within RFC 5127, this effort should not be as controversial as it has been, if we understand that those networks that need more granular traffic treatments can be configured with more granularity while not violating the needs of other networks that do not wish to be made aware of the increased treatment differences.

Everyone involved in this discussion needs to have a clear understanding of the difference points of view within the RFC 4594 effort (i.e., the RFC and the update draft) as well as within RFC 5127. One focuses on defining each service class and the other focuses on determining which of the existing service classes go into which aggregate, if present.

We hope to form a BoF on this subject that will explicitly \*not\* form a working group or produce any documents, or even drafts, but will gather the community from several (if not all) areas, and not just within the transport area. That is the purpose of this draft: to stimulate discussion towards the goal of discussion within the community on DiffServ. If the community does not believe a BoF is necessary, the work will proceed, or not, in TSVWG. Knowing how many within the community have attended TSVWG in each meeting for the last 9 or so years, it is felt that a much wider audience is necessary, given how much impact [ID-4594-UPDATE] can potentially have.

#### 5. Acknowledgements

The author would like to thank Gorrry Fairhurst and David Black for their positive discussions towards the formation of a BoF in Vancouver IETF. The author would also like to thank Paul Jones for doing a valuable proof read to catch points I didn't make clear, as well as identify simple nits I should have caught the nth time I reread this.

#### 6. IANA Considerations

There are no IANA considerations as a result of this document.

## 7. Security Considerations

There are no security considerations within this document because it will not be progressed beyond this individual contributor stage, and all the specifying will be done in other drafts that will wholly contain all the security considerations for this goal/idea.

## 8. References

### 8.1 Normative References

There are no normative references within this document.

### 8.2 Informative References

- [RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers ", RFC 2474, December 1998
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC4594] J. Babiarz, K. Chan, F Baker, "Configuration Guidelines for Diffserv Service Classes", RFC 4594, August 2006
- [RFC5127] Chan, K., Babiarz, J., and F. Baker, "Aggregation of DiffServ Service Classes", RFC 5127, February 2008.
- [RFC5865] F. Baker, J. Polk, M. Dolly, "A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic", RFC 5865, May 2010
- [ID-4594-UPDATE] J. Polk, "Standard Configuration of DiffServ Service Classes", "work in progress", March 2012
- [ID-NEW-DSCPS] J. Polk, "New Differentiated Services Code Point Assignments for Rich Media Traffic", "work in progress", March 2012
- [ID-TCMTF] J. Saldana, D. Wing, J. Fernandez Navajas, Muthu A M. Perumal, J. Ruiz Mas, "Tunneling Compressed Multiplexed Traffic Flows (TCMTF)", "work in progress", March 2012

Authors' Address

James Polk  
3913 Treemont Circle  
Colleyville, Texas 76034

Phone: +1.817.271.3552  
Email: jmpolk@cisco.com

Network WG  
Internet-Draft  
Intended status: Standards Track (PS)  
Expires: August 25, 2013

James Polk  
Cisco  
Feb 25, 2013

New Differentiated Services Code Point Assignments  
for Rich Media Traffic  
draft-polk-tsvwg-new-dscp-assignments-02.txt

Abstract

This document requests five new Differentiated Services Code Point (DSCP) values (DSCP) from the Internet Assigned Numbers Authority (IANA) for new classes of rich media traffic and one additional DSCP value for the signaling of multimedia sessions.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 25, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.



## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	4
3. Evolution of the Proposed DSCPs . . . . .	4
4. New DSCP Assignments. . . . .	7
5. Acknowledgements . . . . .	8
6. IANA Considerations . . . . .	8
7. Security Considerations . . . . .	9
8. References . . . . .	9
8.1 Normative References . . . . .	9
8.2 Informative References . . . . .	10
Author's Address . . . . .	10

## 1. Introduction

This document requests five new Differentiated Services Code Point (DSCP) values (DSCP) from the Internet Assigned Numbers Authority (IANA) for new classes of rich media traffic and one additional DSCP value for the signaling of multimedia sessions. Four of the six new DSCP values are for traffic classes that are admitted by the network using an additional Capacity-Admission signaling procedure to the normal signaling that occurs between multiple endpoints establishing a traffic flow between endpoints. The additional capacity-admission signaling procedure is offered in RFC 5865 [RFC5865], which defined the Voice-Admit per hop behavior (PHB) DSCP. Each of these four traffic classes can conform to the Expedited Forwarding Per-Hop Behavior, if configured to do so, using the Priority Queuing system such as that defined in Section 1.4.1.1 of [ID-4594-UP].

It is expected that voice and video media samples will be carried using the Real-time Transport Protocol (RTP) [RFC3550], thus making voice by itself indistinguishable from video to routers and switches, unless one of two things occurs:

- o Deep packet inspection (DPI) at the ingress of each DiffServ edge node to determine that the packet is an RTP packet with a certain codec that properly identifies it as either a voice or video packet, or
- o have a separate marking for the packets (i.e., a different DSCP).

It is certainly the case that voice samples/frames can be in the same packet as video frames, thus making the packet marked either voice or video, but that will have to be left to the application to decide if that is a good idea. For what it is worth, most current implementations of mixing the media types have the packets marked as a video.

This effort is based on the work started in RFC 5865 [RFC5865], a Differentiated Services Code Point for Capacity-Admitted Traffic voice only traffic, which recommends the classes created within RFC 4594 [RFC4594] be extended for video traffic flows of different types. Nearly all of what is requested and referenced here is based on what started in RFC 4594, but with video as the dominant application as RFC 5865 recommends. Presently, RFC 4594 is being updated by [ID-4594-UP] for many reasons, including the inclusion of these six new DSCPs.

These four new video classes differ from their existing counterparts in behavior by not being subjected to capacity admission. All of the mentioned traffic classes and subsequent DSCPs within RFC 4594 are non-binding, given that it is a non-normative RFC. RFC 4594 also did not recommend the need for capacity admission traffic classes (aka with associated DSCP values). This document is symbiotic with [ID-4594-UP] which intends to replace RFC 4594 as a standards track update which includes the new DSCP assignments created within this document.

Thus, RFC 4594 defined the need for application assignment of certain DSCPs, but only non-normatively. RFC 5865 defined updated DSCP values for a capacity-admitted voice traffic class that is normative. This document takes what was in RFC 4594, creates 4 new capacity-admitted traffic classes and associated DSCPs. This document also moves one non-capacity-admitted traffic class as well as moves the recommended audio/video signaling DSCP value to another value.

Within RFC 5865, there is the specific call for additional DSCPs for capacity-admitted traffic flows of real-time rich media (video) flows in Section 3 of that document under the heading "Summary: Changes from RFC 4594".

It should be noted here that these flows are typically video flows, and frequently include the audio with the adjoining video traffic within that flow. The details of how that gets sorted out are outside the scope of this document. DiffServ is a known and proven mechanism. This document does not change or challenge the idea that Differentiated Services is a Per Hop Behavior (PHB) mechanism, and does not create a service. Here we merely want to add new DSCP assignments because of how at least some of the world is (or wants to) differentiate video from other traffic, including other video traffic.

Section 3 will discuss some of the evolution of DSCP assignments, focusing on those aspects pertinent to the creation of these six new DSCP values. Section 4 describes and defines each of the six DSCP values being requested. Heavy reliance exists on the text of RFC 5865 for its diagrams and charts. Those were not brought into this document at this time, but could be in the future.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

CAC - defined in RFC 5865

PHB - defined in RFC 5865

DSCP - defined in RFC 5865

Queue - defined in RFC 5865

## 3. Evolution of the Proposed DSCPs

First of all, full consideration of PHBs and DSCPs needs to originate with RFC 2474. Section 6 of that document states the following:

"The DSCP field within the DS field is capable of conveying 64 distinct codepoints. The codepoint space is divided into three pools for the purpose of codepoint assignment and management: a pool of 32 RECOMMENDED codepoints (Pool 1) to be assigned by Standards Action as defined in [CONS], a pool of 16 codepoints (Pool 2) to be reserved for experimental or Local Use (EXP/LU) as defined in [CONS], and a pool of 16 codepoints (Pool 3) which are initially available for experimental or local use, but which should be preferentially utilized for standardized assignments if Pool 1 is ever exhausted. The pools are defined in the following table (where 'x' refers to either '0' or '1'):

Pool	Codepoint space	Assignment Policy
----	-----	-----
1	xxxxx0	Standards Action
2	xxxxx11	EXP/LU
3	xxxxx01	EXP/LU (*)

(\*) may be utilized for future Standards Action allocations as Necessary"

The key part of the above quote is

"... which should be preferentially utilized for standardized assignments if Pool 1 is ever exhausted..."

which we here take to mean 'SHOULD NOT use unless you have a really good reason to use'. We propose what we consider a really good

reason to use some of the assignments from Pool 3 before Pool 1 is exhausted. One reason for assigning out of Pool 3 is to get similar marking from layer 2 technologies that only have 3 bits to use for their value, not 6 bits. Technologies such as 802.3 Ethernet, 802.11 Wireless Ethernet, and MPLS are 3 examples of technologies that only have 3 bits to use.

[Editor's Note: If this aspect of assigning DSCPs from Pool 3 before Pool 1 is exhausted requires an update to RFC 2474, please let the authors know so we can point this out to the community for additional feedback.]

Just as RFC 5865 matched the first 3 (or 4) bits with EF for Voice-Admit (101110 and 101100), we RECOMMEND the admitted DSCP for an existing value be its XXXX01 version of the non-admitted DSCP (XXXXX0). We note that the last two bits MUST NOT be x11 because that would mean the value is a Pool 2 value, which is forbidden currently by RFC 2474.

Thus, a DSCP value commonly traverses a layer 2 device by ignoring the last 3 bits of the DSCP value, i.e., taking EF, which is 101110, and reducing it to 101 only, and transmitting this over the layer 2 infrastructure.

RFC 4954, and its intended replacement document [ID-4594-UP], create several service classes primarily intended for video traffic with slightly different characteristics. It was stated there that not all video DSCP values from RFC 4594 are expected to be within the same network, but that could be the case.

RFC 4594 listed these voice and video services classes:

- o "Telephony" using the EF DSCP
- o "Realtime Interactive" using the CS4 DSCP
- o "Multimedia Conferencing" using the AF4X DSCP
- o "Multimedia Streaming" using the AF3X DSCP
- o "Broadcast Video" using the CS3 DSCP

Plus, for Telephony Signaling

- o "Signaling" using the CS5 DSCP

[ID-4594-UP] lists these 'non-admitted' voice and video services classes (some with changed service names, as well as some DSCPs changed):

- o Audio using the EF DSCP

- o Video using the AF4X DSCP
- o Hi-Res using the CS4 DSCP
- o Realtime-Interactive using the CS5 DSCP
- o Multimedia Streaming using the AF3X DSCP
- o Broadcast using the CS3 DSCP

The Multimedia Conferencing purpose and meaning has been changed within [ID-DSCP-UP], as has its DSCPs, which will be listed in the next set of bullets and defined within this document.

RFC 5865 created the new capacity-admitted Voice-Admit, which mentions specifically that a reservation protocol, "such as RSVP" is used to establish those sessions or traffic flows.

This document creates six additional services classes that are incorporated into [ID-4594-UP]:

- o Hi-Res-Admit using the CS4-Admit (100001) DSCP
- o Realtime-Interactive-Admit using the CS5-Admit (101001) DSCP
- o Multimedia Conferencing using the MC (011101) DSCP
- o Multimedia Conferencing-Admit using the MC-Admit (100101) DSCP
- o Broadcast-Admit using the CS3-Admit (011001) DSCP

Plus, for Conversational Signaling (a term described in [ID-4594-UP]), which is no longer to use the CS5 DSCP,

- o "A/V-Sig" using the 010001 DSCP

The results of this are that the

- CS4-Admit is the xxxxxx1 version of CS4.
- CS5-Admit is the xxxxxx1 version of CS5.
- CS3-Admit is the xxxxxx1 version of CS3.

MC-Admit is not the xxxxxx1 version of the new MC DSCP value (100101), because there are no more 100xxx values that are available, outside of the two x11 values from Pool 2, which cannot be assigned for public use.

[Editor's Note: The author is open to suggestions from the community for how to resolve this issue, if anyone considers it an issue.]

The new goal for the signaling service class is to not be starved. It has been shown that mission critical voice and video call set-up does not require expedited forwarding as a PHB. However, this service class MUST NOT be starved, and so it is RECOMMENDED to use a codepoint similar in characteristics to the RFC 4594 (and [ID-4594-UP] defined Low-Latency Data service class of 010xxx.

#### 4. New DSCP Assignments

##### 4.1 The CS5-Admit PHB

'CS5-Admit' MUST be used with a capacity-admission signaling procedure similar to what is required of 'Voice-Admit' [RFC5865]. RSVP [RFC2205] and NSIS [RFC4080] are two good examples for data-path signaling for capacity-admission. Neither is mandatory, but one of them SHOULD be used.

CS5-Admit has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for CS5-Admit is 101001.

##### 4.2 The CS4-Admit DSCP

'CS4-Admit' MUST be used with a capacity-admission signaling procedure similar to what is required of 'Voice-Admit' [RFC5865]. RSVP [RFC2205] and NSIS [RFC4080] are two good examples for data-path signaling for capacity-admission. Neither is mandatory, but one of them SHOULD be used.

CS4-Admit has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for CS4-Admit is 100001.

##### 4.3 The CS3-Admit DSCP

'CS3-Admit' MUST be used with a capacity-admission signaling procedure similar to what is required of 'Voice-Admit' [RFC5865]. RSVP [RFC2205] and NSIS [RFC4080] are two good examples for data-path signaling for capacity-admission. Neither is mandatory, but one of them SHOULD be used.

CS3-Admit has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for CS3-Admit is 011001.

#### 4.4 The MC DSCP

'MC' SHOULD NOT use a capacity-admission signaling procedure. Rather, the MC-Admit is used with a capacity-admission signaling procedure if needed. This PHB MUST be non-admitted.

MC has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for MC is 011001.

#### 4.5 The MC-Admit DSCP

'MC-Admit' MUST be used with a capacity-admission signaling procedure similar to what is required of 'Voice-Admit' [RFC5865]. RSVP [RFC2205] and NSIS [RFC4080] are two good examples for data-path signaling for capacity-admission. Neither is mandatory, but one of them SHOULD be used.

MC-Admit has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for MC-Admit is 100101.

#### 4.6 The Conversational Signaling (A/V-Sig) DSCP

'A/V-Sig' MUST be used with a capacity-admission signaling procedure similar to what is required of 'Voice-Admit' [RFC5865]. RSVP [RFC2205] and NSIS [RFC4080] are two good examples for data-path signaling for capacity-admission. Neither is mandatory, but one of them SHOULD be used.

A/V-Sig has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for A/V-Sig is 010001.

### 5. Acknowledgements

The author would like to thank Paul Jones, Glen Lavers, Mo Zanaty, David Benham, Michael Ramalho for their comments and questions about this effort that ultimately helped shape this document.

### 6. IANA Considerations

IANA is requested to make the following registry assignments from Pool 1 and Pool 3 from the dscp-parameters section within IANA. Justification for assigning from Pool 3 is in Section 3 of this document, and are the only possible parallel assignments to existing assignments of similar registries - very much for the reason Voice-Admit [RFC5865] was assigned a codepoint similar to EF. That

aspect is the main point of this document.

## 6.1 DSCP Assignments from Pool 1

The code point described in this document is requested to be added to the Pool 1 Codepoint table as follows:

Sub-registry: Pool 1 Codepoints

Reference: [RFC2474]

Registration Procedures: Standards Action

Registry: Name	Space	Reference
-----	-----	-----
A/V-Sig	010010	[this document]

## 6.2 DSCP Assignments from Pool 3

A new "Pool 3 Codepoints" table is requested to be built by IANA similar to the Pool 1 Codepoint table in the form:

Sub-registry: Pool 3 Codepoints

Reference: [RFC2474]

Registration Procedures: Standards Action

Registry: Name	Space	Reference
-----	-----	-----
CS5-Admit	101001	[this document]
CS4-Admit	100001	[this document]
CS3-Admit	011001	[this document]
MC-Admit	100101	[this document]
MC	011001	[this document]

## 7. Security Considerations

The Security Considerations are identical to those of RFC 5865.

Every newly proposed DSCP (save A/V-Sig) serves the same security risk and properties of the Voice-Admit DSCP. Section 3 of this document discusses why these DSCP values are should be parallel to their non-admitted counterparts, just as Voice-Admit states in RFC 5865 it is parallel to the existing (at the time) EF.

The A/V-Sig merely has a new DSCP name, RFC 4594 currently has this service class called "Signaling", serving the same purpose.

## 8. References

### 8.1 Normative References



- [ID-4594-UP] J. Polk, "Standard Configuration of DiffServ Service Classes", "work in progress", February 2013
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2205] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997
- [RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers ", RFC 2474, December 1998
- [RFC5865] F. Baker, J. Polk, M. Dolly, "A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic", RFC 5865, May 2010

## 8.2 Informative References

- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC4080] R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005
- [RFC4594] J. Babiarez, K. Chan, F Baker, "Configuration Guidelines for Diffserv Service Classes", RFC 4594, August 2006

## Author's Addresses

James Polk  
3913 Treemont Circle  
Colleyville, Texas 76034

Phone: +1.817.271.3552  
Email: jmpolk@cisco.com

Network WG  
Internet-Draft  
Intended status: Standards Track (PS)  
Obsoletes: RFC 4594  
Updates: RFC 5865  
Expires: August 25, 2013

James Polk, ed.  
Cisco  
Feb, 2013

Standard Configuration of DiffServ Service Classes  
draft-polk-tsvwg-rfc4594-update-03.txt

Abstract

This document describes service classes configured with DiffServ and identifies how they are used and how to construct them using Differentiated Services Code Points (DSCPs), traffic conditioners, Per-Hop Behaviors (PHBs), and Active Queue Management (AQM) mechanisms. There is no intrinsic requirement that particular DSCPs, traffic conditioners, PHBs, and AQM be used for a certain service class, but for consistent behavior under the same network conditions, configuring networks as described here is appropriate.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 25, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction .....	3
1.1. Requirements Notation .....	
1.2. Expected Use in the Network .....	
1.3. Service Class Definition .....	
1.4. Key Differentiated Services Concepts .....	
1.4.1. Queuing .....	
1.4.1.1. Priority Queuing .....	
1.4.1.2. Rate Queuing .....	
1.4.2. Active Queue Management .....	
1.4.3. Traffic Conditioning .....	
1.4.4. Differentiated Services Code Point (DSCP) .....	
1.4.5. Per-Hop Behavior (PHB) .....	
1.5. Key Service Concepts .....	
1.5.1. Default Forwarding (DF) .....	
1.5.2. Assured Forwarding (AF) .....	
1.5.3. Expedited Forwarding (EF) .....	1
1.5.4. Class Selector (CS) .....	1
1.5.5. Admission Control .....	1
1.6 What Changes are Proposed Here from RFC 4594?.....	1
2. Service Differentiation .....	1
2.1. Service Classes .....	1
2.2. Categorization of User Oriented Service Classes .....	1
2.3. Service Class Characteristics .....	1
2.4. Service Classes vs. Treatment Aggregate (from RFC 5127)...	2
2.4.1 Examples of Service Classes in Treatment Aggregates...	2
3. Network Control Traffic .....	2
3.1. Current Practice in the Internet .....	2
3.2. Network Control Service Class .....	2
3.3. OAM Service Class .....	2
4. User Oriented Traffic .....	3
4.1. Conversational Service Class Group .....	3
4.1.1 Audio Service Class .....	3
4.1.2 Video Service Class .....	3
4.1.3 Hi-Res Service Class .....	3
4.2. Realtime-Interactive Service Class .....	3
4.3. Multimedia Conferencing Service Class .....	3
4.4. Multimedia Streaming Service Class .....	3
4.5. Broadcast Video Service Class .....	4
4.6. Low-Latency Data Service Class .....	4
4.7. Conversational Signaling Service Class .....	4
4.8. High-Throughput Data Service Class .....	4
4.9. Standard Service Class .....	4
4.10. Low-Priority Data .....	4
5. Additional Information on Service Class Usage .....	4
5.1. Mapping for NTP .....	5

5.2. VPN Service Mapping .....	5
6. Security Considerations .....	5
7. Contributing Authors .....	5
8. Acknowledgements .....	5
9. References .....	5
9.1. Normative References .....	5
9.2. Informative References .....	5
Author's Address .....	5
Appendix A - Changes .....	5

## 1. Introduction

Differentiated Services [RFC2474][RFC2475] provides the ability to mark/label/classify IP packets differently to distinguish how individual packets need to be treated differently through (or throughout) a network on a per hop basis. Local administrators are who configure each router for which Differentiated Services Code Points (DSCP) are to be treated differently, which are to be ignored (i.e., no differentiated treatment), and which DSCPs are to have their packets remarked (to different DSCPs) as they pass through a router. Local administrators are also who assign which applications, or traffic types, should use which DSCPs to receive the treatment the administrators expect within their network.

What most people fail to understand is that DSCPs provide a per hop behavior (PHB) through that router, but not the previous or next router. In this way of understanding PHB markings, one can understand that Differentiated Services (DiffServ) is not a Quality of Service (QoS) mechanism, but rather a Classification of Service (CoS) mechanism.

For instance, there are 64 possible DSCP values, i.e., using 6 bits of the old Type of Service (TOS) byte [RFC0791]. Each can be configured locally to have greater or less treatment relative to any other DSCP with two exceptions\*.

- \* Expedited Forwarding (EF) [RFC3246] DSCPs have a treatment requirement that any packet marked within an EF class has to be the next packet transmitted out its egress interface. If there are more than one EF marked packet in the queue, obviously the queue sets the order they are transmitted. Further, if there are more than one EF DSCP, local configuration determines if each are treated the same or differently relate to each other EF DSCP. Currently, there are two Expedited Forwarding DSCPs: EF (101110) [RFC3246] and VOICE-ADMIT (101100) [RFC5865].

- \* Class Selector 6 (CS6) [RFC2474] is for routing protocol traffic. There are deemed important because if the network does not transmit and receive its routing protocol traffic in a timely manner, the network stops operating properly.

Not all are configured to mean anything other than best effort forwarding by local administrators of a network. Let us say there are 5 DSCPs configured within network A. Network A's administrator chooses and configures which order (obeying the two exceptions noted above) which application packets are treated differently than any other packets within that network (A). The DSCPs are not fixed to a linear order for relative priority on a per hop basis. Further, and this is often the case, there might be packets with the same DSCP arriving at multiple interfaces of a node, each egressing that node out the same interface. At ingress to this node, everything was fine, with no poor behavior or noticeably excessive amount of packets with the same DSCP. However, at the egress interface, there might not be enough capacity to satisfy the load, thus the departing packets transmit at their maximum rate for that DSCP, but have additional latency due to the overload within that one node. This is called fan-in congestion (or problem). By itself, DiffServ will not remedy this problem for the application that is intolerant to added latency because DiffServ only functions within 1 node at a time.

An additional mechanism is needed to ensure each flow or session receives the amount of packets at its destination that the application requires to perform properly; a mechanism such as IntServ, by way of RSVP [RFC2205] or NSIS [RFC4080]. With this added capability to be session aware, something DiffServ is not, the packets transmitted within a single session have a very good probability of arriving in such a way the receiving application can make full use of each. That said, signaling reservations for each session or flow adds complexity, which creates more work for those who maintain and administer such a network. Adding bandwidth and using DiffServ marking is an easier pill to swallow. The deployment of not few, but more and more audio and (particularly bandwidth hogging) video codecs and their respective application rigidity has caused some to conclude that throwing bandwidth at the problem is no longer acceptable.

With this in mind, this document incorporates five of the six new DSCPs from [ID-DSCP] identified as capacity-admitted DSCPs for most of the service classes in this document. As explained in [ID-DSCP], the five new capacity-admitted DSCPs are from Pool 3. [ID-DSCP] goes further to explain that many layer 2 technologies use fewer bits for marking and prioritization. Instead of six bits like DiffServ, they have three bits, which yields a maximum of 8 values, which tend to line up quite well with the TOS field values. Thus, aggregation of DSCPs is typically accomplished by simply ignoring or reducing the number of bits used to the most significant ones available, such as

EF is 101110, at layer 2 this is merely 101;

Broadcast is 011000, at layer 2 this is merely 011.

However, that was not a premise DiffServ was built upon, to merely

reduce the number of bits. In other words, within DiffServ, XXX is not the same as XXX000 (where XXX is the same binary value in both cases).

This document is originally built upon the RFC 4594 effort, while updating some of the usages and expanding the scope for newer applications that are in use today. The idea in RFC 4594 remains true here, to define a set of service classes, each having unique traffic characteristics, and assigning one or more DSCPs to each service class. As much as the focus could be on the DSCP values, it is not. The focus of this document is the unique traffic characteristics of each service class.

There are many services classes defined in this document, not all will be used in each network at any period of time. This consistency packet markings we talk about is for several reasons, including in a network that does not currently implement a certain service class because they do not have that type of traffic in their network, or that the network merely gives that traffic best effort service. Having a solid guideline to know where to progress or reconfigure a network and endpoints to, say from best effort for a particular traffic type, is a very good thing to do more uniformly than not. A fair amount of burden is placed at DS boundaries needing to keep up with which markings turn into which other markings at both ingress and egress to a network. The same holds true for application developers choosing a default DSCP for their application, lacking a guideline means everyone picks for themselves - and usually with a highly inflated sense of self importance for their application or service.

Another point to make is that there are 20+ service classes defined within the IETF, and that is far too many for most service providers to manage effectively. So, they have formed groups around certain aggregation solutions of service classes. One such aggregation group is based on RFC 5127, which defines what it calls a treatment aggregate, which is taking RFC 4594's service classes and placing them each into one of four treatment aggregates for service providers to handle as a group. SG12 within the ITU-T has an alternative that has nine aggregate groups, so there is work to be done to harmonize aggregates of service classes. This discussion is articulated more in section 2.4. At the end of Section 2.4 we have introduced a series of example configurations which provide examples of how only a few service classes - yet still most treatment aggregates - can be configured in example networks.

Does RFC 4594 need updating? That document is an informational guideline on how networks can or should mark certain packet flows with differing traffic characteristics using DiffServ. There are several reasons why this informational RFC lacks the necessary clarity and strength to reach widespread adoption:

- o confusion between RFC 4594 and RFC 5127 [RFC5127], the latter of

which is for aggregating many 6-bit DSCP values into a 3-bit (8 value) field used specifically by service provider (SP) networks.

- o some believe both RFCs are for SPs, while others ignore RFC 5127 and use RFC 4594 as if it were standards track or BCP.
- o some believe RFC 5127 is for SPs only, and want RFC 4594 to reduce the number of DSCPs within its guidelines to recommend using only 3 or 4 DSCPs. This seems to stem from a manageability and operational perspective.
- o some know RFC 4594 is informational and do not follow its guidelines specifically because it is informational.
- o some use DSCP values that are not defined within RFC 4594, making mapping between different networks using similar or identical application flows difficult.
- o some believe enterprise networks should not use either RFC except at the edge of their networks, where they directly connect to SP networks.
- o some argue that the services classes guidance per class is too broad and are therefore not sure in which service class a particular application is to reside.
- o time has shown that video has become a dominant application on the Internet, and many believe it now requires to be treated uniquely in environments that want to. Video also does not always plan nice with audio, so knowing the two use the same transport (RTP) [RFC3550], a means of separation is in order.

Service class definitions are based on the different traffic characteristics and required performance of the applications/services. There are a greater number of service classes in this document than there were when RFC 4594 [RFC4594] was published (the RFC this document intends to obsolete). The required performance of applications/services has also changed since the publication of RFC 4594, specifically in the area of conversational real time communications. As a result, this document has a greater number of real time applications with more granular set of DSCPs due to their different required performances. Like RFC 4594 before, this approach allows those applications with similar traffic characteristics and performance requirements to be placed in the same service class.

The notion of traffic characteristics and required performance is a per application concept, therefore the label name of each service class remains the same on an end-to-end basis, even if we understand that DiffServ is only a PHB and cannot guarantee anything, even packet delivery at the intended destination node. That said, several applications can be configured to have the same DSCP, or

each have different DSCPs that have the same treatment per hop within a network.

Since RFC 4594 was first published, a new concept has been introduced that will appear throughout this document, including DSCP assignments -- the idea of "admitted" traffic, initially introduced into DiffServ within RFC 5865 [RFC5865]. The VOICE-ADMIT Expedited Forwarding class differentiates itself from the EF Expedited Forwarding by having the packets marked be for admitted traffic. This concept of "admitted" traffic is spread throughout the real time traffic classes.

Thus, the document flow is as follows:

- o maintain the general format of RFC 4594;
- o augment the content with the concept of capacity-admission;
- o incorporate more video into this document, as it has become a dominant application in enterprises and other managed networks, as well as on the open public Internet;
- o reduce the discussion on voice and its examples;
- o articulate the subtle differences learned since RFC 4594 was published.

The goal here is to provide a standard configuration for DiffServ DSCP assignments and expected PHBs for enterprises and other managed networks, as well as towards the public Internet with specific traffic characteristics per Service class/DSCP, and example applications shown for each.

This document describes service classes configured with DiffServ and defines how they can be used and how to construct them using Differentiated Services Code Points (DSCPs), and recommends how to construct them using traffic conditioners, Per-Hop Behaviors (PHBs), and Active Queue Management (AQM) mechanisms. There is no intrinsic requirement that particular traffic conditioners, PHBs, and AQM be used for a certain service class, but as a policy and for interoperability it is useful to apply them consistently.

We differentiate services and their characteristics in Section 2. Network control traffic, as well as user oriented traffic are discussed in Sections 3 and 4, respectively. We analyze the security considerations in Section 6. Section 7 offers a tribute to the authors of RFC 4594, from which this document is based. It is in its own section, and not part of the normal acknowledgements portion of each IETF document.



### 1.1. Requirements Notation

The key words "SHOULD", "SHOULD NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

### 1.2. Expected Use in the Network

In the Internet today, corporate LANs and ISP WANs are increasingly utilized, to the point in which network congestion is affecting performance of applications. For this reason, congestion, loss, and variation in delay within corporate LANs and ISP backbones is becoming known to the users collectively as "the network is slow for this application" or just "right now" or "for today". Users do not directly detect network congestion. They react to applications that run slow, or to downloads that take too long in their mind(s). The explosion of video traffic on the internet recently has caused much of this, and is often the application the user is using when they have this slowness.

In the past, application slowness occurred for three very good reasons.

- o the networks the user oriented traffic traverses moves through cycles of bandwidth boom and bandwidth bust, the latter of which become apparent with the periodic deployment of new bandwidth-hungry applications.
- o In access networks, the state is often different. This may be because throughput rates are artificially limited or over-subscribed, or because of access network design trade-offs.
- o Other characteristics, such as database design on web servers (that may create contention points, e.g., in filestore) and configuration of firewalls and routers, often look externally like a bandwidth limitation.

The intent of this document is to provide a standardized marking, plus a conditioning and packet treatment strategy so that it can be configured and put into service on any link that is itself congested.

### 1.3. Service Class Definition

A "service class" represents a similar set of traffic characteristics for delay, loss, and jitter as packets traverse routers in a network. For example, "High-Throughput Data" service class for store-and-forward applications, or a "Broadcast" service

class for minimally time-shifted IPTV or Internet radio broadcasts. Such a service class may be defined locally in a Differentiated Services (DS) domain, or across multiple DS domains, possibly extending end to end. A goal of this document is to have most/all networks assign the same type of traffic the same for consistency.

A service class is a naming convention which is defined as a word, phrase or initialism/acronym representing a set of necessary traffic characteristics of a certain type of data flow. The necessary characteristics of these traffic flows can be realized by the use of defined per-hop behavior that started with [RFC2474]. The actual specification of the expected treatment of a traffic aggregate within a domain may also be defined as a per-domain behavior (PDB) [RFC3086].

Each domain will locally choose to

- o implement one or more service classes with traffic characteristics as defined here, or
- o implement one or more service classes with similar traffic characteristics as defined here, or
- o implement one or more service classes with similar traffic characteristics as defined here and to aggregate one or more service classes to reduce the number of unique DSCPs within their network, or
- o implement one or more non-standard service classes with traffic characteristics not as defined here, or
- o not use DiffServ within their domain.

For example, low delay, low loss, and minimal jitter may be realized using the EF PHB, or with an over-provisioned AF PHB. This must be done with care as it may disrupt the end-to-end performance required by the applications/services. If the packet sizes are similar within an application, but different between two applications, say small voice packets and large video packets, these two applications may not realize optimum results if merged into the same aggregate if there are any bottlenecks in the network. We provide for this flexibility on a per hop or per domain basis within this document.

This document provides standardized markings for traffic with similar characteristics, and usage expectations for PHBs for specific service classes for their consistent implementation.

The Default Forwarding "Standard" service class is REQUIRED; all other service classes are OPTIONAL. That said, each service class lists traffic characteristics that are expected when using that type of traffic. It is RECOMMENDED that applications and protocols that fit a certain traffic characteristic use the appropriate service

class mark, i.e., the DSCP, for consistent behavior. It is expected that network administrators will base their endpoint application and router configuration choices on the level of service differentiation they require to meet the needs of their customers (i.e., their end-users).

#### 1.4. Key Differentiated Services Concepts

In order to fully understand this document, a reader needs to familiarize themselves with the principles of the Differentiated Services Architecture [RFC2474]. We summarize some key concepts here only to provide convenience for the reader, the referenced RFCs providing the authoritative definitions.

##### 1.4.1. Queuing

A queue is a data structure that holds packets that are awaiting transmission. A router interface can only transmit one packet at a time, however fast the interface speed is. If there is only 1 queue at an interface, the packets are transmitted in the order they are received into that queue - called FIFO, or "first in, first out". Sometimes there is a lag in the time between a packets arrives in the queue and when it is transmitted. This delay might be due to lack of bandwidth, or if there are multiple queues on that interface, because a packet is low in priority relative to other packets that are awaiting to transmit. The scheduler is the system entity that chooses which packet is next in line for transmission when more than one packet are awaiting transmission out the same router interface.

##### 1.4.1.1 Priority Queuing

A priority queuing system is a combination of a set of queues and a scheduler that empties the queues (of packets) in priority sequence. When asked for a packet, the scheduler inspects the highest priority queue and, if there is data present, returns a packet from that queue. Failing that, it inspects the next highest priority queue, and so on. A freeway onramp with a stoplight for one lane that allows vehicles in the high-occupancy-vehicle lane to pass is an example of a priority queuing system; the high-occupancy-vehicle lane represents the "queue" having priority.

In a priority queuing system, a packet in the highest priority queue will experience a readily calculated delay. This is proportional to the amount of data remaining to be serialized when the packet arrived plus the volume of the data already queued ahead of it in the same queue. The technical reason for using a priority queue relates exactly to this fact: it limits delay and variations in delay and should be used for traffic that has that requirement.

A priority queue or queuing system needs to avoid starvation of lower-priority queues. This may be achieved through a variety of means, such as admission control, rate control, or network engineering.

#### 1.4.1.2. Rate Queuing

Similarly, a rate-based queuing system is a combination of a set of queues and a scheduler that empties each at a specified rate. An example of a rate-based queuing system is a road intersection with a stoplight. The stoplight acts as a scheduler, giving each lane a certain opportunity to pass traffic through the intersection.

In a rate-based queuing system, such as Weighted Fair Queuing (WFQ) or Weighted Round Robin (WRR), the delay that a packet in any given queue will experience depends on the parameters and occupancy of its queue and the parameters and occupancy of the queues it is competing with. A queue whose traffic arrival rate is much less than the rate at which it lets traffic depart will tend to be empty, and packets in it will experience nominal delays. A queue whose traffic arrival rate approximates or exceeds its departure rate will tend not to be empty, and packets in it will experience greater delay. Such a scheduler can impose a minimum rate, a maximum rate, or both, on any queue it touches.

#### 1.4.2 Active Queue Management

Active Queue Management, or AQM, is a generic name for any of a variety of procedures that use packet dropping or marking to manage the depth of a queue. The canonical example of such a procedure is Random Early Detection (RED), in that a queue is assigned a minimum and maximum threshold, and the queuing algorithm maintains a moving average of the queue depth. While the mean queue depth exceeds the maximum threshold, all arriving traffic is dropped. While the mean queue depth exceeds the minimum threshold but not the maximum threshold, a randomly selected subset of arriving traffic is marked or dropped. This marking or dropping of traffic is intended to communicate with the sending system, causing its congestion avoidance algorithms to kick in. As a result of this behavior, it is reasonable to expect that TCP's cyclic behavior is desynchronized and that the mean queue depth (and therefore delay) should normally approximate the minimum threshold.

A variation of the algorithm is applied in Assured Forwarding PHB [RFC2597], in that the behavior aggregate consists of traffic with multiple DSCP marks, which are intermingled in a common queue. Different minima and maxima are configured for the several DSCPs separately, such that traffic that exceeds a stated rate at ingress is more likely to be dropped or marked than traffic that is within its contracted rate.

#### 1.4.3 Traffic Conditioning

In addition, at the first router in a network that a packet crosses, arriving traffic may be measured and dropped or marked according to a policy, or perhaps shaped on network ingress, as in "A Rate Adaptive Shaper for Differentiated Services" [RFC2963]. This may be used to bias feedback loops, as is done in "Assured Forwarding PHB" [RFC2597], or to limit the amount of traffic in a system, as is done in "Expedited Forwarding PHB" [RFC3246]. Such measurement procedures are collectively referred to as "traffic conditioners". Traffic conditioners are normally built using token bucket meters, for example with a committed rate and burst size, as in Section 1.5.3 of the DiffServ Model [RFC3290]. The Assured Forwarding PHB [RFC2597] uses a variation on a meter with multiple rate and burst size measurements to test and identify multiple levels of conformance.

Multiple rates and burst sizes can be realized using multiple levels of token buckets or more complex token buckets; these are implementation details. The following are some traffic conditioners that may be used in deployment of differentiated services:

- o For Class Selector (CS) PHBs, a single token bucket meter to provide a rate plus burst size control.
- o For Expedited Forwarding (EF) PHB, a single token bucket meter to provide a rate plus burst size control.
- o For Assured Forwarding (AF) PHBs, usually two token bucket meters configured to provide behavior as outlined in "Two Rate Three Color Marker (trTCM)" [RFC2698] or "Single Rate Three Color Marker (srTCM)" [RFC2697]. The two-rate, three-color marker is used to enforce two rates, whereas the single-rate, three-color marker is used to enforce a committed rate with two burst lengths.

#### 1.4.4 Differentiated Services Code Point (DSCP)

The DSCP is a number in the range 0..63 that is placed into an IP packet to mark it according to the class of traffic it belongs in. These are divided into 3 groups, or pools, defined in RFC 2474, arranged as follows:

- o Pool-1 has 32 values designated for standards assignment (of the form 'xxxxx0').
- o Pool-2 has 16 values designated for experimental or local use only (EXP/LU) assignment (of the form 'xxxx11').
- o Pool-3 has 16 values designated for experimental or local use (EXP/LU) assignment (of the form 'xxxx01').

However, pool-3 is allowed to be assigned for one of two reasons,

#1 - if the values in pool-1 are exhausted, or

#2 - if there is a justifiable reason for assigning a pool-3 DSCP prior to pool-1's exhaustion.

#### 1.4.5 Per-Hop Behavior (PHB)

In the end, the mechanisms described above are combined to form a specified set of characteristics for handling different kinds of traffic, depending on the needs of the application. This document seeks to identify useful traffic aggregates and to specify what PHB should be applied to them.

#### 1.5 Key Service Concepts

While Differentiated Services is a general architecture that may be used to implement a variety of services, three fundamental forwarding behaviors have been defined and characterized for general use. These are basic Default Forwarding (DF) behavior for elastic traffic, the Assured Forwarding (AF) behavior, and the Expedited Forwarding (EF) behavior for real-time (inelastic) traffic. The facts that four code points are recommended for AF and that one code point is recommended for EF are arbitrary choices, and the architecture allows any reasonable number of AF and EF classes simultaneously. The choice of four AF classes and one EF class in the current document is also arbitrary, and operators MAY choose to operate more or fewer of either.

The terms "elastic" and "real-time" are defined in [RFC1633], Section 3.1, as a way of understanding broad-brush application requirements. This document should be reviewed to obtain a broad understanding of the issues in quality of service, just as [RFC2475] should be reviewed to understand the data plane architecture used in today's Internet.

##### 1.5.1 Default Forwarding (DF)

The basic forwarding behaviors applied to any class of traffic are those described in [RFC2474] and [RFC2309]. Best-effort service may be summarized as "I will accept your packets" and is typically configured with some bandwidth guarantee. Packets in transit may be lost, reordered, duplicated, or delayed at random. Generally, networks are engineered to limit this behavior, but changing traffic loads can push any network into such a state.

Application traffic in the internet that uses default forwarding is expected to be "elastic" in nature. By this, we mean that the sender of traffic will adjust its transmission rate in response to

changes in available rate, loss, or delay.

For the basic best-effort service, a single DSCP value is provided to identify the traffic, a queue to store it, and active queue management to protect the network from it and to limit delays.

#### 1.5.2 Assured Forwarding (AF)

The Assured Forwarding PHB [RFC2597] behavior is explicitly modeled on Frame Relay's Discard Eligible (DE) flag or ATM's Cell Loss Priority (CLP) capability. It is intended for networks that offer average-rate Service Level Agreements (SLAs) (as FR and ATM networks do). This is an enhanced best-effort service; traffic is expected to be "elastic" in nature. The receiver will detect loss or variation in delay in the network and provide feedback such that the sender adjusts its transmission rate to approximate available capacity.

For such behaviors, multiple DSCP values are provided (two or three, perhaps more using local values) to identify the traffic, a common queue to store the aggregate, and active queue management to protect the network from it and to limit delays. Traffic is metered as it enters the network, and traffic is variously marked depending on the arrival rate of the aggregate. The premise is that it is normal for users occasionally to use more capacity than their contract stipulates, perhaps up to some bound. However, if traffic should be marked or lost to manage the queue, this excess traffic will be marked or lost first.

#### 1.5.3. Expedited Forwarding (EF)

The intent of Expedited Forwarding PHB [RFC3246] is to provide a building block for low-loss, low-delay, and low-jitter services. It can be used to build an enhanced best-effort service: traffic remains subject to loss due to line errors and reordering during routing changes. However, using queuing techniques, the probability of delay or variation in delay is minimized. For this reason, it is generally used to carry voice and for transport of data information that requires "wire like" behavior through the IP network. Voice is an inelastic "real-time" application that sends packets at the rate the codec produces them, regardless of availability of capacity. As such, this service has the potential to disrupt or congest a network if not controlled. It also has the potential for abuse.

To protect the network, at minimum one SHOULD police traffic at various points to ensure that the design of a queue is not overrun, and then the traffic SHOULD be given a low-delay queue (often using priority, although it is asserted that a rate-based queue can do this) to ensure that variation in delay is not an issue, to meet application needs.

#### 1.5.4 Class Selector (CS)

Class Selector, those DSCPs that end in zeros (xxx000), provide support for historical codepoint definitions and PHB requirement. The CS fields provide a limited backward compatibility with legacy practice, as described in [RFC2474], Section 4. Backward compatibility is addressed in two ways,

- First, there are per-hop behaviors that are already in widespread use (e.g., those satisfying the IPv4 Precedence queuing requirements specified in [RFC1812]), and
- this document will continue to permit their use in DS-compliant networks.

In addition, there are some DSCPs that correspond to historical use of the IP Precedence field,

- CS0 (000000) will remain 'Default Forwarding' (also known as 'Best Effort')
- 11xxxx will remain for routing traffic

and will map to PHBs that meet the general requirements specified in [RFC2474], Section 4.2.2.2.

No attempt is made to maintain backward compatibility with the "DTR" or Type of Service (TOS) bits of the IPv4 TOS octet, as defined in [RFC0791] and [RFC1349].

A DS-compliant network can be deployed exclusively by using one or more CS-compliant PHB groups. Thus, for example, codepoint '011000' would map to the same PHB as codepoint '011010'.

#### 1.5.5 Admission Control

Admission control (including refusal when policy thresholds are crossed) can ensure high-quality communication by ensuring the availability of bandwidth to carry a load. Inelastic real-time flows such as Voice over Internet Protocol (VoIP) (audio) or video conferencing services can benefit from use of an admission control mechanism, as generally the audio or video service is configured with over-subscription, meaning that some users may not be able to make a call during peak periods.

For VoIP (audio) service, a common approach is to use signaling protocols such as SIP, H.323, H.248, MEGACO, along with Resource Reservation Protocol (RSVP) to negotiate admittance and use of network transport capabilities. When a user has been authorized to send voice traffic, this admission procedure has verified that data rates will be within the capacity of the network that it will use.



Many RTP voice and video payloads are inelastic and cannot react to loss or delay in any substantive way. For these payload types, the network needs to police at ingress to ensure that the voice traffic stays within its negotiated bounds. Having thus assured a predictable input rate, the network may use a priority queue to ensure nominal delay and variation in delay.

#### 1.5.5.1 Capacity Admitted (\*-Admit)

This is a newer group of traffic types that started with RFC 5865 and the Voice-Admit service type. Voice-Admit is an EF class marking but has capacity-admission always applied to it to ensure each of these flows are managed through a network, though not necessarily on an end-to-end basis. This depends on how many networks each flow transits and the load on each transited network. There are a series of new DSCPs proposed in [ID-DSCP], each specifying unique characteristics necessitating a separate marking from what existing before that document.

This document will import in four new '\*-Admit' DSCPs from [ID-DSCP], 2 others that are new but not capacity-admitted, one from RFC 5865, and change the existing usage of 2 DSCPs from RFC 4594. This is discussed throughout the rest of this document.

#### 1.6 What Changes are Proposed Here from RFC 4594?

Changing an entire network DiffServ configuration has proven to be a painful experience for both individuals and companies. It is not done very often, and for good reason. This effort is based on experience learned since the publication of RFC 4594 (circa 2006). Audio, once thought to be ok grouped with video, needs to be in separate service classes. Collaboration has taken off, mostly because of mobility, but also because of a worldwide recession that has limited physical travel, and relying on people to do more with their computers. With that in mind, there has been an explosion in application development for the individual (seems everyone has an "app-store"). The following set of bullets has this world - that needs a robust layer 3 - in mind.

- o Scope of document is changed to tighten it up for standards track consideration.
- o This document explicitly states there is a fundamental requirement that a particular DSCP(s) be used for each service class, each with a recommended set of applications to be used by that service class - at least on that individual's externally facing (public) interface.
- o Created the Conversational group of service classes to focus on realtime, mostly bidirectional communications (unless multicast is

used).

- o "Realtime-Interactive"  
Moved to (near) realtime TCP-based apps

Why the change? TCP based transports have proven, in certain environments, to be a bidirectional realtime transport, e.g., for multiplayer gaming and virtual desktops applications.

- o "Audio"  
Same as Telephony (which is now gone), adds Voice-Admit for capacity-admitted traffic

Why the change? RFC 5865 (Voice-Admit) needed to be added to the Audio service class. Video needed to be separate from audio, hence the name change from Telephony (which includes video) to just audio.

- o "Video"  
NEW for video and audio/video conferencing, was in Multimedia-Conferencing service classification

Why the change? Many networks are using the AF4X for video, but others are throwing anything "multimedia" into the same service class (like elastic TCP flows). Video has become so dominant that it should be what mostly goes into one service class.

- o "Hi-Res"  
NEW for video and audio/video conferencing

Why the change? This entirely new service class is for local policy based higher end video (think Telepresence). Without congestion, this service class has the same treatment as Video, but if there is any pushback from the network, Hi-Res (note: not married to the name) has a better PHB.

- o "Multimedia-Conferencing"  
Now without audio or human video

Why the change? The change is taking bidirectional human audio and video out of this service class. This is all about non-realtime collaboration - even in conjunction with an audio and/or video flow.

- o "Broadcast"  
Remains the same, added CS3-Admit for capacity-admitted

Why the change? Removing the "-Video" from the name because there are so many more flows that are Broadcast in realtime than video.

- o "Low-Latency Data"  
Remains the same, adds IM & Presence traffic explicitly

Why the change? Merely explicitly stating a place for some

additional traffic types that otherwise could go elsewhere.

- o "Conversational Signaling" (A/V-Sig)  
Was 'Signaling'

Why the change? This change is merely a renaming of a service class, and acknowledgement that some of the previous authors inaccurate beliefs that DSCPs were linearly ordered with those values having a higher value definitely getting better treatment than lower values.

## 2. Service Differentiation

There are practical limits on the level of service differentiation that should be offered in the IP networks. We believe we have defined a practical approach in delivering service differentiation by defining different service classes that networks may choose to support in order to provide the appropriate level of behaviors and performance needed by current and future applications and services. The defined structure for providing services allows several applications having similar traffic characteristics and performance requirements to be grouped into the same service class. This approach provides a lot of flexibility in providing the appropriate level of service differentiation for current and new, yet unknown applications without introducing significant changes to routers or network configurations when a new traffic type is added to the network.

### 2.1 Service Classes

Traffic flowing in a network can be classified in many different ways. We have chosen to divide it into two groupings, network control and user/subscriber traffic. To provide service differentiation, different service classes are defined in each grouping. The network control traffic group can further be divided into two service classes (see Section 3 for detailed definition of each service class):

- o "Network Control" for routing and network control function.
- o "OAM" (Operations, Administration, and Management) for network configuration and management functions.

The user/subscriber traffic group is broken down into ten service classes to provide service differentiation for all the different types of applications/services (see Section 4 for detailed definition of each service class):

- o Conversational service group consists of three service classes:
  - Audio, which includes both 'admitted' and 'unadmitted' audio

service classes, is for non-one way (i.e., generally bidirectional) audio media packets between human users of smaller size and at a constant delivery rate.

- Hi-Res Video, which includes both 'admitted' and 'unadmitted' Hi-Res Video service classes, is for video traffic from higher end endpoints between human users necessitating different treatment than from desktop or video phone endpoints. This has a clearly business differentiation, and not a technical differentiation - as both Hi-Res-Video and Video will be treated similarly on the wire when no congestion occurs.
- Video, which includes both 'admitted' and 'unadmitted' video service classes, is for video traffic from lower end endpoints between human users necessitating different treatment than from higher end (i.e., Telepresence) endpoints. This has a clearly business differentiation, and not a technical differentiation - as both Hi-Res-Video and Video will be treated similarly on the wire when no congestion occurs.
- o Conversational Signaling service class is for peer-to-peer and client-server signaling and control functions using protocols such as SIP, H.323, H.248, and Media Gateway Control Protocol (MGCP). This traffic needs to not be starved on the network.

Editor's note: RFC 4594 had this DSCP marking as CS5, but with clearly different characteristics (i.e., no sensitivity to jitter or (unreasonable) delay), this DSCP has been moved to a more appropriate (new) value, defined in [ID-DSCP].

- o Real-Time Interactive, which includes both 'admitted' and 'unadmitted' Realtime-Interactive service class, is for bidirectional variable rate inelastic applications that require low jitter and loss and very low delay, such as interactive gaming applications that use RTP/UDP streams for game control commands, and Virtualized Desktop applications between the user and content source, typically in a centralized data center.
- o Multimedia Conferencing, which includes both 'admitted' and 'unadmitted' multimedia conferencing service class, is for applications that require minimal delay, but not like those of realtime application requirements. This service class can be bursty in nature, as well as not transmit packets for some time. Applications such as presentation data or collaborative application sharing will use this service class.
- o Multimedia Streaming, which includes both 'admitted' and 'unadmitted' multimedia streaming service class, is for one-way bufferable streaming media applications such as Video on Demand (VOD) and webcasts.

- o Broadcast, which includes both 'admitted' and 'unadmitted' broadcast service class, is for inelastic streaming media applications that may be of constant or variable rate, requiring low jitter and very low packet loss, such as broadcast TV and live events, video surveillance, and security.
- o Low-Latency Data service class is for data processing applications such as client/server interactions or Instant Messaging (IM) and Presence data.
- o Conversational Signaling (A/V-Sig) service class is for all signaling messages, whether in-band (i.e., along the data path) or out-of-band (separate from the data path), for the purposes of setting up, maintaining, managing and terminating bi- or multi-directional realtime sessions.
- o High-Throughput Data service class is for store and forward applications such as FTP and billing record transfer.
- o Standard service class, commonly called best effort (BE), is for traffic that has not been identified as requiring differentiated treatment.
- o Low-Priority Data service class, which some could call the scavenger class, is for packet flows where bandwidth assurance is not required.

## 2.2 Categorization of User Oriented Service Classes

The ten defined user/subscriber service classes listed above can be grouped into a small number of application categories. For some application categories, it was felt that more than one service class was needed to provide service differentiation within that category due to the different traffic characteristic of the applications, control function, and the required flow behavior. Figure 1 provides a summary of service class grouping into four application categories.

### Application Control Category

- o The Conversational Signaling service class is intended to be used to control applications or user endpoints. Examples of protocols that would use this service class are SIP, XMPP or H.323 for voice and/or video over IP services. User signaling flows have similar performance requirements as Low-Latency Data, they require a separate DSCP to be distinguished other traffic and allow for a treatment that is unique.

### Media-Oriented Category

Due to the vast number of new (in process of being deployed) and already-in-use media-oriented services in IP networks, seven service

classes have been defined.

- o Audio service class is intended for Voice-over-IP (VoIP) services. It may also be used for other applications that meet the defined traffic characteristics and performance requirements.
- o Video service class is intended for Video over IP services. It may also be used for other applications that meet the defined traffic characteristics and performance requirements.
- o Hi-Res service class is intended for higher end video services that have the same traffic characteristics as the video service class, but have a business requirement(s) to be treated differently. One example of this is Telepresence video applications.
- o Realtime-Interactive service class is intended for inelastic applications such as desktop virtualization applications and for interactive gaming.
- o Multimedia Conferencing service class is for everything about or within video conferencing solutions that does not include the voice or (human) video components. Several examples are
  - the presentation data part of an IP conference (call).
  - the application sharing part of an IP conference (call).
  - the whiteboarding aspect of an IP conference (call).

Each of the above can be part of a lower end web-conferencing application or part of a higher end Telepresence video conference. Each also has the ability to reduce their transmission rate on detection of congestion. These flows can therefore be classified as rate adaptive and most often more elastic than their voice and video counterparts.

- o Broadcast Video service class is to be used for inelastic traffic flows specifically with minimal buffering expected by the source or destination, which are intended for broadcast HDTV service, as well as for transport of live video (sports or concerts) and audio events.
- o Multimedia Streaming service class is to be used for elastic multimedia traffic flows where buffering is expected. This is the fundamental difference between the Broadcast and multimedia streaming service classes. Multimedia streaming content is typically stored before being transmitted. It is also buffered at the receiving end before being played out. The buffering is sufficiently large to accommodate any variation in transmission rate that is encountered in the network. Multimedia entertainment over IP delivery services that are being developed

can generate both elastic and inelastic traffic flows; therefore, two service classes are defined to address this space, respectively: Multimedia Streaming and Broadcast Video.

#### Data Category

The data category is divided into three service classes.

- o Low-Latency Data for applications/services that require low delay or latency for bursty but short-lived flows.
- o High-Throughput Data for applications/services that require good throughput for long-lived bursty flows. High Throughput and Multimedia Streaming are close in their traffic flow characteristics with High Throughput being a bit more bursty and not as long-lived as Multimedia Streaming.
- o Low-Priority Data for applications or services that can tolerate short or long interruptions of packet flows. The Low-Priority Data service class can be viewed as "don't care" to some degree.

#### Best-Effort Category

- o All traffic that is not differentiated in the network falls into this category and is mapped into the Standard service class. If a packet is marked with a DSCP value that is not supported in the network, it SHOULD be forwarded using the Standard service class.

Figure 1, below, provides a grouping of the defined user/subscriber service classes into four categories, with indications of which ones use an independent flow for signaling or control; type of flow behavior (elastic, rate adaptive, or inelastic); and the last column provides end user Class of Service (CoS) rating as defined in ITU-T Recommendation G.1010.

Application Categories	Service Class	Signaled	Flow Behavior	G.1010 Rating
Application Control	A/V Sig	Not applicable	Inelastic	Responsive
Media-	Realtime Interactive	Yes	Inelastic	Interactive
	Audio	Yes	Inelastic	Interactive
	Video	Yes	Inelastic	Interactive
	Hi-Res	Yes	Inelastic	Interactive
	Multimedia	Yes	Rate	Moderately

Oriented	Conferencing		Adaptive	Interactive
	Broadcast	Yes	Inelastic	Responsive
	Multimedia Streaming	Yes	Elastic	Timely
Data	Low-Latency Data	No	Elastic	Responsive
	Conversational Signaling	No	Elastic or Inelastic	Timely
	High-Throughput Data	No	Elastic	Timely
	Low-Priority Data	No	Elastic	Non-critical
Best Effort	Standard	Not Specified		Non-critical

Figure 1. User/Subscriber Service Classes Grouping

Here is a short explanation of the end user CoS category as defined in ITU-T Recommendation G.1010. User oriented traffic is divided into four different categories, namely, interactive, responsive, timely, and non-critical. An example of interactive traffic is between two humans and is most sensitive to delay, loss, and jitter. Another example of interactive traffic is between two servers where very low delay and loss are needed. Responsive traffic is typically between a human and a server but can also be between two servers. Responsive traffic is less affected by jitter and can tolerate longer delays than interactive traffic. Timely traffic is either between servers or servers and humans and the delay tolerance is significantly longer than responsive traffic. Non-critical traffic is normally between servers/machines where delivery may be delay for period of time.

### 2.3. Service Class Characteristics

This document specifies what network administrators are to expect when configuring service classes identified by their differing characteristics. Figure 2 identifies these service classes along with their characteristics, as well as the tolerance to loss, delay and jitter for each service class. Properly engineered networks to these PHBs will achieve expected results. That said, not all of the identified service classes are expected in each operator's network.



Service Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
Network Control	Variable size packets, mostly inelastic short messages, but traffic can also burst (BGP)	Low	Low	Yes
Realtime Interactive	Inelastic, mostly variable rate	Low	Very Low	Low
Audio	Fixed-size small packets, inelastic	Very Low	Very Low	Very Low
Video	Fixed-size small-large packets, inelastic	Very Low	Very Low	Very Low
Hi-Res A/V	Fixed-size small-large packets, inelastic	Very Low	Very Low	Very Low
Multimedia Conferencing	Variable size packets, constant transmit interval, rate adaptive, reacts to loss	Low - Medium	Low - Medium	Low - Medium
Multimedia Streaming	Variable size packets, elastic with variable rate	Low - Medium	Medium	High
Broadcast	Constant and variable rate, inelastic, non-bursty flows	Very Low	Medium	Low
Low-Latency Data	Variable rate, bursty short-lived elastic flows	Low	Low - Medium	Yes
Conversational Signaling	Variable size packets, some what bursty short-lived flows	Low	Low	Yes
OAM	Variable size packets, elastic & inelastic flows	Low	Medium	Yes
High-Throughput Data	Variable rate, bursty long-lived elastic flows	Low	Medium - High	Yes
Standard	A bit of everything	Not Specified		
Low-Priority Data	Non-real-time and elastic	High	High	Yes

Figure 2. Service Class Characteristics

Notes for Figure 2: A "Yes" in the jitter-tolerant column implies that received data is buffered at the endpoint and that a moderate level of server or network-induced variation in delay is not expected to affect the application. Applications that use TCP or SCTP as a transport are generally good examples. Routing protocols and peer-to-peer signaling also fall in this class; although loss can create problems in setting up calls, a moderate level of jitter merely makes call placement a little less predictable in duration.

Service classes indicate the required traffic forwarding treatment in order to meet user, application, and/or network expectations. Section 3 defines the service classes that MAY be used for forwarding network control traffic, and Section 4 defines the service classes that MAY be used for forwarding user oriented traffic with examples of intended application types mapped into each service class. Note that the application types are only examples and are not meant to be all-inclusive or prescriptive. Also, note that the service class naming or ordering does not imply any priority ordering. They are simply reference names that are used in this document with associated QoS behaviors that are optimized for the particular application types they support. Network administrators MAY choose to assign different service class names to the service classes that they will support. Figure 3 defines the RECOMMENDED relationship between service classes and DS codepoint assignment with application examples. It is RECOMMENDED that this relationship be preserved end to end.

Service Class Name	DSCP Name	DSCP Value	Application Examples
Network Control	CS6&CS7	11xxxx	Network routing
Realtime Interactive	CS5, CS5-Admit	101000, 101001	Remote/Virtual Desktop and Interactive gaming
Audio	EF Voice-Admit	101110 101100	Voice bearer
Hi-Res A/V	CS4, CS4-Admit	100000, 100001	Conversational Hi-Res Audio/Video bearer
Video	AF41,AF42 AF43	100010,100100 100110	Audio/Video conferencing bearer
Multimedia Conferencing	MC, MC-Admit	011101, 100101	Presentation Data and App Sharing/Whiteboarding
Multimedia Streaming	AF31,AF32 AF33	011010,011100 011110	Streaming video and audio on demand

Broadcast	CS3, CS3-Admit	011000, 011001	Broadcast TV, live events & video surveillance
Low-Latency Data	AF21,AF22 AF23	010010,010100 010110	Client/server trans., Web- based ordering, IM/Pres
Conversational Signaling	A/V-Sig	010001	Conversational signaling
OAM	CS2	010000	OAM&P
High-Throughput Data	AF11,AF12 AF13	001010,001100 001110	Store and forward applications
Low-Priority Data	CS1	001000	Any flow that has no BW assurance
Best Effort	CS0	000000	Undifferentiated applications

Figure 3. DSCP to Service Class Mapping

Notes for Figure 3:

- o Default Forwarding (DF) and Class Selector 0 (CS0) (i.e., Best Effort) provide equivalent behavior and use the same DS codepoint, '000000'.
- o RFC 2474 identifies any DSCP with a value of 11xxxx to be for network control. This remains true, while it removes 12 DSCPs from the overall pool of 64 available DSCP values (the 4 that are x11 from this group are within pool 2 of RFC 2474, and remain as only experimentally assignable/useable).
- o All PHB names that say "-Admit" are to be used only when a capacity-admission protocol is utilized for that or each traffic flow.

Changes from table 3 of RFC 4594 are as follows:

- o The old term "Signaling" was using CS5 (101000), now is exclusively for the "Conversational Signaling" service group using the DSCP name of "A/V-Sig" (010001), which is newly defined in [ID-DSCP]. This is because CS5 aggregates into the 101xxx aggregate when using layer 2 technologies such as 802.3 Ethernet, 802.11 Wireless Ethernet MPLS, etc - each of which only have 3 bits to mark with. A traffic type that can have very large packets and is not delay sensitive (within reason) is not appropriate for have a 101xxx marking. A REQUIRED behavior for this PHB is that it not be starved in any node.

- o "Conversational" is a new term to include all interactive audio and video. The Conversational service group consists of the audio service class, the video service class and the new Hi-Res service class.
- o "Audio" obsoletes the term "Telephony", which has generally not retained the "video" aspect within the IETF, where video is still commonly called out as a separate thing. Audio retains the nonadmitted traffic PHB of EF (101110), while capacity-admitted audio has been added via the RFC 5865 defined PHB Voice-Admit.
- o "Video" now is AF4x, with AF41 specifically for capacity-admitted video traffic, while AF42 and AF43 are nonadmitted video traffic.
- o "Hi-Res A/V", part of the Conversational service group, is created by [ID-DSCP] for an additional business differentiation interactive video marking for higher end traffic. It is within the 100xxx as CS4 (for nonadmitted traffic) and CS4-Admit (100001) (for capacity-admitted traffic).
- o "Realtime Interactive" is now using CS5 (for nonadmitted traffic), but adds a capacity-admitted DSCP CS5-Admit (101001).
- o "Multimedia Conferencing" is no longer using the AF4x DSCPs, rather it will use the new PHB MC (100101) (for capacity-admitted) and MC-Admit (011101) (for nonadmitted traffic).
- o "Multimedia Streaming" retains using AF3x, however, AF31 is now used for capacity-admitted traffic, while AF32/33 are nonadmitted.
- o "Broadcast" replaces "Broadcast Video" using CS3 (for nonadmitted traffic), and adds a capacity-admitted PHB CS3-Admit (011001).

It is expected that network administrators will base their choice of the service classes that they will support on their need.

Figure 4 provides a summary of DiffServ CoS mechanisms that MUST be used for the defined service classes that are further detailed in Sections 3 and 4 of this document. According to what applications/services need to be differentiated, network administrators MAY choose the service class(es) that need to be supported in their network.

Service Class	DSCP	Conditioning at DS Edge	PHB Used	Queuing	AQM
Network Control	CS6/CS7	See Section 3.1	RFC2474	Rate	Yes
Realtime	CS5,	Police using sr+bs	RFC2474	Rate	No

Interactive	CS5- Admit*		[[ID-DSCP]]		
Audio	EF, Voice- Admit*	Police using sr+bs	RFC3246 RFC5865	Priority	No
Hi-Res A/V	CS4, CS4- Admit*	Police using sr+bs	RFC2474 [[ID-DSCP]]	Priority	No
Video	AF41*, AF42 AF43	Using two-rate, three-color marker (such as RFC 2698)	RFC2597	Rate	Yes per DSCP
Multimedia Conferencing	MC, MC- Admit*	Police using sr+bs	[[ID-DSCP]] [[ID-DSCP]]	Rate	No
Multimedia Streaming	AF31*, AF32 AF33	Using two-rate, three-color marker (such as RFC 2698)	RFC2597	Rate	Yes per DSCP
Broadcast	CS3, CS3- Admit*	Police using sr+bs	RFC2474 [[ID-DSCP]]	Rate	No
Low- Latency Data	AF21 AF22 AF23	Using single-rate, three-color marker (such as RFC 2697)	RFC2597	Rate	Yes per DSCP
Conversational Signaling	AV-Sig	Police using sr+bs	[[ID-DSCP]]	Rate	No
OAM	CS2	Police using sr+bs	RFC2474	Rate	Yes
High- Throughput Data	AF11 AF12 AF13	Using two-rate, three-color marker (such as RFC 2698)	RFC2597	Rate	Yes per DSCP
Standard	DF	Not applicable	RFC2474	Rate	Yes
Low-Priority Data	CS1	Not applicable	RFC3662	Rate	Yes

Figure 4. Summary of CoS Mechanisms Used for Each Service Class

\* denotes each DSCP identified for capacity-admission traffic only.

Notes for Figure 4:

- o Conditioning at DS edge means that traffic conditioning is performed at the edge of the DiffServ network where untrusted user devices are connected to two different administrative DiffServ networks.
- o "sr+bs" represents a policing mechanism that provides single rate with burst size control.
- o The single-rate, three-color marker (srTCM) behavior SHOULD be equivalent to RFC 2697, and the two-rate, three-color marker (trTCM) behavior SHOULD be equivalent to RFC 2698.
- o The PHB for Realtime-Interactive service class SHOULD be configured to provide high bandwidth assurance. It MAY be configured as another EF PHB (one capacity-admitted and one non-capacity-admitted, if both are to be used) that uses relaxed performance parameters and a rate scheduler.
- o The PHB for Multimedia Conferencing service class SHOULD be configured to provide high bandwidth assurance. It MAY be configured as another EF PHB (one capacity-admitted and one non-capacity-admitted, if both are to be used) that uses relaxed performance parameters and a rate scheduler.
- o The PHB for Broadcast service class SHOULD be configured to provide high bandwidth assurance. It MAY be configured as another EF PHB (one capacity-admitted and one non-capacity-admitted, if both are to be used) that uses relaxed performance parameters and a rate scheduler.

#### 2.4. Service Classes vs. Treatment Aggregates (from RFC 5127)

There are misconceptions about the differences between RFC 4594 specified service classes, and RFC 5127 specified treatment aggregates. Often the two are conflated, and more often the phrase service class is used to mean both definitions. Almost all of the text previous to this section is used in defining service classes, and how one service class is different than another service class (based on traffic characteristics of the applications). Treatment aggregates are groupings of service classes with similar, but not identical, traffic characteristics to give similar treatment from a SP's network.

Below is taken from appendix of RFC 5127 as its recommended groupings of service classes into aggregates based in RFC 4594 specified traffic characteristic expectations.

+-----+			
Treatment	Treatment	DSCP	
Aggregate	Aggregate		
	Behavior		

Network Control	CS (RFC 2474)	CS6
Real-Time*	EF (RFC 3246)	EF, CS5, AF41, AF42, AF43, CS4, CS3
Assured Elastic	AF (RFC 2597)	CS2, AF31, AF21, AF11 AF32, AF22, AF12 AF33, AF23, AF13
Elastic	Default (RFC 2474)	Default, (CS0) CS1

Figure 5: RFC 5127 Defined Treatment Aggregate Behavior\*\*

\*NOTE: The RFC 5865 created VOICE-ADMIT is absence from the above figure because VOICE-ADMIT was created far later than this recommendation was. VOICE-ADMIT is appropriate for the Realtime Traffic Aggregate.

\*\*NOTE: Figure 5 is directly from the appendix of RFC 5127 as that RFC's recommendation for configuration. This draft does not directly affect RFC 5127. That is left for an update to RFC 5127 itself. Based on the WG's take on this draft, RFC 5127 will necessitate an update to match this document's new service classes and additional DSCPs. The number of treatment aggregates are not expected to change in the RFC 5127 Update draft though, with the possible exception of a new treatment aggregate for capacity admitted flows; meaning there *might* be a 5th treatment aggregate proposed.

Treatment Aggregates are designed to nicely fit into technologies that do not have many different treatment levels to use. Here are 3 examples of technologies limited to an 8-value field,

- MPLS with its 3 Traffic Class (TC) bits [RFC5462].
- IEEE LANs with its 8-value Priority Code Point (PCP) field, as part of the 802.1Q header spec [IEEE1Q].
- IEEE 802.1e, which defines QoS over Wi-Fi, also only defines 8 levels (called User Priority or UP codes) [IEEE1E].

Treatment Aggregates are dependent on service classes to exist. Therefore many service classes can exist without the need to consider the use of treatment aggregates or their 8-value technologies. For example, a Layer 3 VPN can be all that is needed

to transit traffic flows, regardless of desired treatment, between enterprise LAN campuses. From this reality, the number of treatment aggregates has no direct bearing on the number of service classes.

#### 2.4.1 Examples of Service Classes in Treatment Aggregates

It is *\*not\** expected that all traffic characteristics are to be experienced across an SP's network for any given customer. For example, if VOICE-ADMIT is added to the Realtime Treatment Aggregate in Figure 5, there are 8 different service classes within the Realtime Treatment Aggregate. It is not expected that all 8 service classes will be deployed by customer networks traversing SP networks. RFC 5127's Treatment Aggregates are a table to configure which service class goes into which treatment aggregate. If there are 8 services classes in the Realtime treatment aggregate, there is very little difference than if there were one service within that same Realtime treatment aggregate - it would still be necessary to configure that treatment aggregate. Thus, it becomes a question of not

"how many service classes are there that go into treatment aggregates?"

but

"how many treatment aggregates have one or more services classes requiring configuration"?

Of the 4 treatment aggregates shown in Figure 5, if there are existing service classes in only 3 of the aggregates, then only 3 treatment aggregates are necessary. Of the 3 following examples, notice that examples 2 and 3 have the same number of treatment aggregates, but example 3 has more applications in their own service classes.

Examples 2 and 3 are made under the following assumptions:

- this draft's Service Classes and DSCP assignments are utilized.
- the new AF-Sig DSCP in the Assured Elastic treatment aggregate.
- the Audio, Video service classes are in the EF treatment aggregate.
- the VOICE-ADMIT DSCP is in the EF treatment aggregate.

##### 2.4.1.1 Example 1 - Simple Voice Configuration/SLA

For example 1, we have an SP running MPLS and has an SLA to deliver Network Control, Voice and everything else is Best Effort. The



following table would apply to this configuration/SLA:

Applications	Service Class	DSCP(s)	Treatment Aggregate
Network Control	Network Control	CS6	Network Control
Voice	Audio	EF	Realtime
Everything else	DF	Default (CS0)	Elastic

Figure 6. Example 1 Configuration

Insert different treatments for this example  
(i.e., AQM, RED, WFQ, colors, etc from above charts)

#### 2.4.1.2 Example 2 - Voice/Video/Surveillance Configuration/SLA

For example 1, we have an SP running MPLS and has an SLA to deliver Control, audio, video, surveillance, audio & video signaling, and everything else is BE

Applications	Service Class	DSCP(s)	Treatment Aggregate
Network Control	Network Control	CS6	Network Control
Voice, video, surveillance	Audio, Video, Broadcast	EF, AF42, CS3	Realtime
audio & video signaling	Conversational Signaling	AV-Sig	Assured Elastic
Everything else	DF	Default (CS0)	Elastic

Figure 7. Example 2 Configuration

Insert different treatments for this example  
(i.e., AQM, RED, WFQ, colors, etc from above charts)

#### 2.4.1.2 Example 3 - Complex CAC realtime/Surveillance/+apps Configuration/SLA

For example 1, we have an SP running MPLS and has an SLA to deliver

Control, voice, CAC voice, CAC video, streaming, signaling, LL data, Network Mgmt., and everything else is BE (including non-CAC video because it is not authorized or authenticated on network)

Applications	Service Class	DSCP(s)	Treatment Aggregate
Network Control	Network Control	CS6	Network Control
Voice, CAC-Voice CAC-video, surveillance	Audio, Video, Broadcast	Voice-Admit EF, AF41 CS3	Realtime
audio & video signaling, VOD (streaming), Network Mgmt.	Conversational Signaling, Low- Latency Data, Multimedia Streaming, OAM	AV-Sig AF21 AF31 CS2	Assured Elastic
Everything else	DF	Default (CS0)	Elastic

Figure 8. Example 3 Configuration

Insert different treatments for this example  
(i.e., AQM, RED, WFQ, colors, etc from above charts)

### 3. Network Control Traffic

Network control traffic is defined as packet flows that are essential for stable operation of an administered network, as well as the information exchanged between neighboring networks across a peering point where SLAs are in place. Network control traffic is different from user application control (signaling) that may be generated by some applications or services. Network control traffic is mostly between routers and network nodes (e.g., routing or mgmt protocols) that are used for operating, administering, controlling, or managing whole networks, network parts or just network segments. Network Control Traffic may be split into two service classes, i.e., Network Control and OAM.

#### 3.1. Current Practice in the Internet

Based on today's routing protocols and network control procedures that are used in the Internet, we have determined that CS6 DSCP value SHOULD be used for routing and control and that CS7 DSCP value SHOULD be reserved for future use, specifically if needed for future

routing or control protocols. Network administrators MAY use a Local/Experimental DSCP, any value that contains 11xx11; therefore, they may use a locally defined service class within their network to further differentiate their routing and control traffic.

RECOMMENDED Network Edge Conditioning for CS7 DSCP marked packets:

- o Drop or remark 111xxx packets at ingress to DiffServ network domain.
- o 111xxx marked packets SHOULD NOT be sent across peering points. Exchange of control information across peering points SHOULD be done using CS6 DSCP and the Network Control service class.
- o any internally defined 11xxx1 values, valid within that network domain, be remarked to CS6 upon egress at network peering points.

### 3.2. Network Control Service Class

The Network Control service class is used for transmitting packets between network devices (routers) that require control (routing) information to be exchanged between similar devices within the administrative domain, as well as across a peering point between adjacent administrative domains. Traffic transmitted in this service class is very important as it keeps the network operational, and it needs to be forwarded in a timely manner.

The Network Control service class SHOULD be configured using the DiffServ CS6 PHB, defined in [RFC2474]. This service class MUST be configured so that the traffic receives a minimum bandwidth guarantee, to ensure that the packets always receive timely service. The configured forwarding resources for Network Control service class MUST be such that the probability of packet drop under peak load is very low. The Network Control service class SHOULD be configured to use a Rate Queuing system such as defined in Section 1.4.1.2 of this document.

The following are examples of protocols and applications that MUST use the Network Control service class if present in a network:

- o Routing packet flows: OSPF, BGP, ISIS, RIP.
- o Control information exchange within and between different administrative domains across a peering point where SLAs are in place.
- o LSP setup using CR-LDP and RSVP-TE.

The following protocols and applications MUST NOT use the Network Control service class:

- o User oriented traffic is not allowed to use this service class.

By user oriented traffic, we mean packet flows that originate from user-controlled end points that are connected to the network.

- o even if originating from a server or a device acting on behalf of a user or endpoint,
- o even if it is application or in-band signaling to establish a connection wholly within a single network or across peering points of/to adjacent networks (e.g., creating a tunnel such as a VPN, or data path control signaling).

The following are traffic characteristics of packet flows in the Network Control service class:

- o Mostly messages sent between routers and network servers.
- o Variable size packets, normally one packet at a time, but traffic can also burst (BGP, OSPF, etc).
- o IGMP, hen is used only for the normal multicast routing purpose.

The REQUIRED DSCP marking is CS6 (Class Selector 6).

RECOMMENDED Network Edge Conditioning:

- o At peering points (between two DiffServ networks) where SLAs are in place, CS6 marked packets MUST be policed, e.g., using a single rate with burst size (sr+bs) token bucket policer to keep the CS6 marked packet flows to within the traffic rate specified in the SLA.
- o CS6 marked packet flows from untrusted sources (for example, end user devices) MUST be dropped or remarked at ingress to the DiffServ network. What a network admin remarks this user oriented traffic to is a matter of local policy, and inspection of the packets can determine which application is used for proper marking to a more appropriate DSCP, such as from table 3. of this document.
- o Packets from users/subscribers are not permitted access to the Network Control service classes.

The fundamental service offered to the Network Control service class is enhanced best-effort service with high bandwidth assurance. Since this service class is used to forward both elastic and inelastic flows, the service SHOULD be engineered so that the Active Queue Management (AQM) [RFC2309] is applied to CS6 marked packets.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth, and the max-threshold specifies the queue depth above which all traffic is dropped or ECN marked. Thus,

in this service class, the following inequality should hold in queue configurations:

- o min-threshold CS6 < max-threshold CS6
- o max-threshold CS6 <= memory assigned to the queue

Note: Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

### 3.3. OAM Service Class

The OAM (Operations, Administration, and Management) service class is RECOMMENDED for OAM&P (Operations, Administration, and Management and Provisioning) using protocols such as Simple Network Management Protocol (SNMP), Trivial File Transfer Protocol (TFTP), FTP, Telnet, and Common Open Policy Service (COPS). Applications using this service class require a low packet loss but are relatively not sensitive to delay. This service class is configured to provide good packet delivery for intermittent flows.

The OAM service class SHOULD use the Class Selector (CS) PHB defined in [RFC2474]. This service class SHOULD be configured to provide a minimum bandwidth assurance for CS2 marked packets to ensure that they get forwarded. The OAM service class SHOULD be configured to use a Rate Queuing system such as defined in Section 1.4.1.2 of this document.

The following applications SHOULD use the OAM service class:

- o Provisioning and configuration of network elements.
- o Performance monitoring of network elements.
- o Any network operational alarms.

The following are traffic characteristics:

- o Variable size packets.
- o Intermittent traffic flows.
- o Traffic may burst at times.
- o Both elastic and inelastic flows.
- o Traffic not sensitive to delays.

RECOMMENDED DSCP marking:

- o All flows in this service class are marked with CS2 (Class Selector 2).

Applications or IP end points SHOULD pre-mark their packets with CS2 DSCP value. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

RECOMMENDED conditioning performed at DiffServ network edge:

- o Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods, defined in [RFC2475].
- o Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the traffic stays within its negotiated or engineered bounds.
- o Packet flows from trusted sources (routers inside administered network) MAY not require policing.
- o Normally OAM&P CS2 marked packet flows are not allowed to flow across peering points. If that is the case, then CS2 marked packets SHOULD be policed (dropped) at both egress and ingress peering interfaces.

The fundamental service offered to "OAM" traffic is enhanced best-effort service with controlled rate. The service SHOULD be engineered so that CS2 marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Since this service class is used to forward both elastic and inelastic flows, the service SHOULD be engineered so that Active Queue Management [RFC2309] is applied to CS2 marked packets.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth for each DSCP, and the max-threshold specifies the queue depth above which all traffic with such a DSCP is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- o min-threshold CS2 < max-threshold CS2
- o max-threshold CS2 <= memory assigned to the queue

Note: Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

#### 4. User Oriented Traffic

User oriented traffic is defined as packet flows between different users or subscribers, or from servers/nodes on behalf of a user. It is the traffic that is sent to or from end-terminals and that

supports a very wide variety of applications and services, to include traffic about a user or application that assists a user communicate. User oriented traffic can be classified in many different ways. What we have articulated throughout this document is a series of non-exhaustive list of categories for classifying user oriented traffic. We differentiated user oriented traffic that is real-time versus non-real-time, elastic or rate-adaptive versus inelastic, sensitive versus insensitive to loss as well as considering whether the traffic is interactive vs. one way communication, its responsiveness, whether it requires timely delivery, and critical versus non-critical. In the final analysis, we used all of the above for service differentiation, mapping application types that seemed to have different sets of performance sensitivities, and requirements to different service classes.

Network administrators can categorize their applications according to the type of behavior that they require and MAY choose to support all or a subset of the defined service classes. At the same time, we include a public facing default DSCP value, with its associated PHB, that is expected for each traffic type to ensure common or pervasive performance. Figure 3 provides some common applications and the forwarding service classes that best support them, based on their performance requirements.

#### 4.1. Conversational Service Class Group

The Conversational Service Class Group consists of 3 different service classes, audio, video, and Hi-Res. We are describing the media sample, or bearer, packets for applications (e.g., RTP from [RFC3550]) that require bi-directional real-time, very low delay, very low jitter, and very low packet loss for relatively constant-rate traffic sources (inelastic traffic sources). It is RECOMMENDED that RTCP feedback use the same service class and be marked with the same DSCP as the bearer traffic for that (audio and/or video) call. This ensures comparable treatment within the network between endpoints.

The signaling to set-up these bearer flows is part of the Conversational Signaling service group that will be discussed later in Section 4. The following 3 subsections will detail what is expected within each bearer service class.

##### 4.1.1 Audio Service Class

This service class MUST be used for IP Audio service.

The fundamental service offered to traffic in the Audio service class is minimum jitter, delay, and packet loss service up to a specified upper bound. There are two PHBs, both EF based, for the Audio service class:

Nonadmitted Audio traffic - MUST use the EF DSCP [RFC3246], and

is for traffic that has not had any capacity admission signaling performed for that flow or session.

Capacity-Admitted Audio traffic - MUST use the Voice-Admit DSCP [RFC5865], and is for traffic that has had any capacity admission signaling performed for that flow or session, e.g., RSVP [RFC2205] or NSIS [RFC4080].

The capacity-admitted Audio traffic operation is similar to an ATM CBR service, which has guaranteed bandwidth and which, if it stays within the negotiated rate, experiences nominal delay and no loss.

The nonadmitted Audio traffic, on the other hand, has had no such explicit guarantee, but has a favorable PHB ensuring high probability of delivery as well as nominal delay and no loss - implicitly assuming there is not too much like marked traffic between users within a flow.

There are two typical scenarios in which audio calls are established, on the public open Internet using protocols such as SIP, XMPP or H.323, or in more managed networks like enterprises or certain service providers which offer a audio service with some feature benefits and take part in the call signaling. These SPs or enterprises also use protocols like SIP, XMPP, H.323, but also use H.248/MEGACO and MGCP.

On the open Internet, typically there is no SP actively involved in the session set-up of calls, and therefore no servers providing assistance or features to help one user contact another user. Often, this traffic is marked or remarked with the DF (i.e., Best Effort) DSCP.

In more managed networks in which one of more operators have active servers aiding the audio call set-up, where DiffServ can be used and preserved to differentiate traffic, networks are offering a service, therefore need to do some, or a lot of engineering to ensure that capacity offered to one or more applications does not exceed the load to the network. Otherwise, the operator will have unhappy users, at least for that application's usage. This is true for any application, but is especially true for inelastic applications in which the application is rigid in its delivery requirements. Audio bearer traffic is typically such an application, video is another such application, but we will get to video in the next subsection.

When a user in a managed network has been authorized to send Audio traffic (i.e., call initiation via the operator's servers was not rejected), the call admission procedure should have verified that the newly admitted flow will be within the capacity of the Audio service class forwarding capability in the network. Capacity verification is a non-trivial thing, and can either be implicitly assumed by the call server(s) based on the operator's network design, or it can be explicitly signaled from an in-data-path



signaling mechanism that verifies the capacity is available now for this call, for each call made within that network. In the latter case, those that do not have verifiable network capacity along the data path are rejected. An in between means method is for call servers to count calls between two or more endpoints. By topologically understanding where the caller and called party is and have configured a known maximum it will allow between the two locations. This is especially true over WAN links that have far less capacity than LAN links or core parts of a network. Network operators will need to understand the topology between any two callers to ensure the appropriate amount of bandwidth is available for an expected number of simultaneous audio calls.

Once more than one bandwidth amount can be used for audio calls, for example - by allowing more than one codec with different bandwidths per codec for such calls, network engineering becomes more difficult. Since the inelastic nature of RTP payloads from this class do not react well to loss or significant delay in any substantive way, the Audio service class MUST forward packets as soon as possible.

The Audio service class that does not have capacity admission performed in the data path MUST use the Expedited Forwarding (EF) PHB, as defined in [RFC3246], so that all packets are forwarded quickly. The Audio service class that does have capacity admission performed in the data path MUST use the Voice-Admit PHB, as defined in [RFC5865], so that all packets are forwarded quickly. The Audio service class SHOULD be configured to use a Priority Queuing system such as that defined in Section 1.4.1.1 of this document.

The following applications SHOULD use the Audio service class:

- o VoIP (G.711, G.729, iLBC and other audio codecs).
- o Voice-band data over IP (modem, fax).
- o T.38 fax over IP.
- o Circuit emulation over IP, virtual wire, etc.
- o IP Virtual Private Network (VPN) service that specifies single-rate, mean network delay that is slightly longer than network propagation delay, very low jitter, and a very low packet loss.

The following are traffic characteristics:

- o Mostly fixed-size packets for VoIP (30, 60, 70, 120 or 200 bytes in size).
- o Packets emitted at constant time intervals.

- o Admission control of new flows is provided by Audio call server, media gateway, gatekeeper, edge router, end terminal, access node or in-data-path signaling that provides flow admission control function.

Applications or IP end points SHOULD pre-mark their packets with EF or Voice-Admit DSCP value, whichever is appropriate. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

The RECOMMENDED DSCP marking is EF for nonadmitted audio flows, and Voice-Admit for capacity-admitted flows for the following applications:

- o VoIP (G.711, G.729 and other codecs).
- o Voice-band data over IP (modem and fax).
- o T.38 fax over IP.
- o Circuit emulation over IP, virtual wire, etc.

RECOMMENDED Network Edge Conditioning:

- o Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods, defined in [RFC2475]. If untrusted, the network edge SHOULD know if capacity-admission has been applied, since the edge router will have taken part in the admission signaling; therefore will know whether EF or Voice-Admit is the proper marking for that flow.
- o Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the Audio traffic stays within its negotiated bounds.
- o Policing is OPTIONAL for packet flows from trusted sources whose behavior is ensured via other means (e.g., administrative controls on those systems).
- o Policing of Audio packet flows across peering points where SLA is in place is OPTIONAL as Audio traffic will be controlled by admission control mechanism between peering points.

The fundamental service offered to "Audio" traffic is enhanced best-effort service with controlled rate, very low delay, and very low loss. The service MUST be engineered so that EF marked packet flows have sufficient bandwidth in the network to provide guaranteed delivery. Otherwise, the service will have in place an explicit capacity-admission signaling protocol such as RSVP or NSIS and thus

mark the packets within the flow as Voice-Admit. Normally traffic in this service class does not respond dynamically to packet loss. As such, Active Queue Management [RFC2309] SHOULD NOT be applied to EF marked packet flows.

#### 4.1.2 Video Service Class

The Video service class is for bidirectional applications that require real-time service for both constant and rate-adaptive traffic. SIP and H.323/V2 (and later) versions of video conferencing equipment with constant and dynamic bandwidth adjustment are such applications. The traffic sources in this service class either have a fixed bandwidth requirement (e.g., MPEG2, etc.), or have the ability to dynamically change their transmission rate (e.g., MPEG4/H.264, etc.) based on feedback from the receiver. This feedback SHOULD be accomplished using RTCP [RFC3550]. One approach for this downspeeding has the receiver detect packet loss, thus signaling in an RTCP message to the source the indication of lost (or delayed or out of order) packets in transit. When necessary the source then selects a lower rate encoding codec. When available, the source merely sends less data, resulting in lower resolution of the same visual display.

The Video service class is not for video downloads, webcasts, or single directional video or audio/video traffic of any kind. It is for human-to-human visual interaction between two users, or more if an MTP is used.

Typical video conferencing configurations negotiate the setup of audio/video session using protocols such as SIP and H.323. Just as with networks that have audio traversing them, video typically traverses the same two types of networks: the open big "I" Internet, in which most every type of traffic is best effort (DF), or on a more managed network such as an enterprise or SP's managed network in which servers within either network take part in the call signaling, thereby offering the video service.

When a user in a managed network has been authorized to send video traffic (i.e., call initiation via the operator's servers was not rejected), the call admission procedure should have verified that the newly admitted flow will be within the capacity of the video service class forwarding capability in the network. Capacity verification is a non-trivial thing, and can either be implicitly assumed by the call server(s) based on the operator's network design, or it can be explicitly signaled from an in-data-path signaling mechanism that verifies the capacity is available now for this call, for each call made within that network. In the latter case, those that do not have verifiable network capacity along the data path are rejected. An in between means method is for call servers to count calls between two or more endpoints. By topologically understanding where the caller and called party is and

have configured a known maximum it will allow between the two locations. Video is larger in bandwidth than audio, and the difference can be significant. For example, for a single G.711 audio call that is 80kbps, an associated video bandwidth for the same call can easily be 4Mbps. This is especially true over WAN links that have far less capacity than LAN links or core parts of a network. Network operators will need to understand the topology between any two callers to ensure the appropriate amount of bandwidth is available for an expected number of simultaneous video and/or audio/video calls.

Note that it is OPTIONALLY the case in these networks that the accompanying audio for the video call will be marked as the video is marked (i.e., using the same DSCP), but not always. One reason this has been done is for lip-sync.

The Video service class MUST use the Assured Forwarding (AF) PHB, defined in [RFC2597]. This service class MUST be configured to provide a bandwidth assurance for AF41, AF42, and AF43 marked packets to ensure that they get forwarded. The Video service class SHOULD be configured to use a Rate Queuing system for AF42 and AF43 traffic flows, such as that defined in Section 1.4.1.2 of this document. However, AF41 MUST be designated as the DSCP for use when capacity-admission signaling has been used, such as RSVP or NSIS, to guarantee delivery through the network. AF42 and AF43 will be used for non-admitted video calls, as well as overflows from AF41 sources that send more packets than they have negotiated bandwidth for that call.

The following applications MUST use the Video service class:

- o SIP and H.323/V2 (and later) versions of video conferencing applications (interactive video).
- o Video conferencing applications with rate control or traffic content importance marking.
- o Interactive, time-critical, and mission-critical applications.

NOTE with regards to the above bullet: this usage SHOULD be minimized, else the video traffic will suffer - unless this is engineered into the topology.

The following are traffic characteristics:

- o Variable size packets (i.e., small to large in size).
- o The higher the resolution or change rate between each image, the higher the duration of large packets.
- o Usually constant inter-packet time interval.

- o Can be Variable rate in transmission.
- o Source is capable of reducing its transmission rate based on being told receiver is detecting packet loss (e.g., via RTCP).

Applications or IP end points SHOULD pre-mark their packets with DSCP values as shown below. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475] and mark all packets as AF4x. Note: In this case, the two-rate, three-color marker will be configured to operate in Color-Blind mode.

Mandatory DSCP marking when performed by router closest to source:

- o AF41 = up to specified rate "A", which is dedicated to non-Hi-Res capacity-admitted video traffic.

Note the audio of an A/V call can be marked AF41 as well.

- o AF42 = all non-Hi-Res video traffic marked AF41 in excess of specified rate "A", or new non-admitted video traffic but below specified rate "B".
- o AF43 = in excess of specified rate "B".
- o Where "A" < "B".

Note: One might expect "A" to approximate the peak rates of sum of all admitted video flows, plus the sum of the mean rates and "B" to approximate the sum of the peak rates of those same two flows.

Mandatory DSCP marking when performed by SIP or H.323/V2 videoconferencing equipment:

- o AF41 = SIP or H.323 video conferencing audio stream RTP.
- o AF41 = SIP or H.323 video conferencing video control RTCP.
- o AF41 = SIP or H.323 video conferencing video stream up to specified rate "A".
- o AF42 = SIP or H.323 video conferencing video stream in excess of specified rate "A" but below specified rate "B".
- o AF42 = SIP or H.323 video conferencing video control RTCP, for those video streams that were generated using AF42.
- o AF43 = SIP or H.323 video conferencing video stream in excess of specified rate "B".

- o AF43 = SIP or H.323 video conferencing video control RTCP, for those video streams that were generated using AF43.
- o Where "A" < "B".

Mandatory conditioning performed at DiffServ network edge:

- o The two-rate, three-color marker SHOULD be configured to provide the behavior as defined in trTCM [RFC2698].
- o If packets are marked by trusted sources or a previously trusted DiffServ domain and the color marking is to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Aware mode.
- o If the packet marking is not trusted or the color marking is not to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Blind mode.

The fundamental service offered to nonadmitted "Video" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Video" traffic is a guaranteed service using in-data-path signaling to ensure expected delivery in a timely manner. For a non-admitted video conferencing service, if a 1% packet loss detected at the receiver triggers an encoding rate change, thus dropping to the next lower provisioned video encoding rate then Active Queue Management [RFC2309] SHOULD be used primarily to switch the video encoding rate under congestion, changing from high rate to lower rate, i.e., 1472 kbps to 768 kbps. This rule applies to all AF42 and 43 flows. The probability of loss of AF41 traffic MUST NOT exceed the probability of loss of AF42 traffic, which in turn MUST NOT exceed the probability of loss of AF43 traffic.

Capacity-admitted video service should not result in packet loss. However, administratively this MAY be allowed to cause a purposeful downspeeding event (i.e., a change in resolution or a change in codec) to occur due to congestion.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth for each DSCP, and the max-threshold specifies the queue depth above which all traffic with such a DSCP is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- o min-threshold AF43 < max-threshold AF43
- o max-threshold AF43 <= min-threshold AF42
- o min-threshold AF42 < max-threshold AF42
- o max-threshold AF42 <= min-threshold AF41

- o min-threshold AF41 < max-threshold AF41
- o max-threshold AF41 <= memory assigned to the queue

Note: This configuration tends to drop AF43 traffic before AF42 and AF42 before AF41. Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

#### 4.1.3 Hi-Res Service Class

The Hi-Res service class is for higher end (i.e., deemed 'more important') bidirectional applications that require real-time service for both constant and rate-adaptive traffic. There are two PHBs, both EF based, for the Hi-Res video conferencing service class:

Nonadmitted Hi-Res traffic - MUST use the CS4 DSCP [RFC2474], and is for traffic that has not had any capacity admission signaling performed for that flow or session.

Capacity-Admitted Hi-Res traffic - MUST use the CS4-Admit DSCP [ID-DSCP], and is for traffic that has had any capacity admission signaling performed for that flow or session, e.g., RSVP [RFC2205] or NSIS [RFC4080].

The capacity-admitted Hi-Res video conferencing traffic operation is similar to an ATM CBR service, which has guaranteed bandwidth and which, if it stays within the negotiated rate, experiences nominal delay and no loss.

SIP and H.323/V2 (and later) versions of video conferencing equipment with constant and dynamic bandwidth adjustment are such applications. The traffic sources in this service class either have a fixed bandwidth requirement (e.g., MPEG2), or have the ability to dynamically change their transmission rate (e.g., MPEG4/H.264) based on feedback from the receiver. This feedback SHOULD be accomplished using RTCP [RFC3550]. One approach for this downspeeding has the receiver detect packet loss, thus signaling in an RTCP message to the source the indication of lost (or delayed or out of order) packets in transit. When necessary the source then selects a lower rate encoding codec. When available, the source merely sends less data, resulting in lower resolution of the same visual display.

The Hi-Res service class, as with the Video service class, is not for video downloads, webcasts, or single directional video or audio/video traffic of any kind. It is for human-to-human visual interaction between two users, or more if a video conference bridge is used.

Typical Hi-Res video conferencing configurations negotiate the setup

of audio/video session using protocols such as SIP and H.323. Hi-Res video conferencing is generally not over the big "I" Internet, rather nearly exclusively over more managed networks such as an enterprise or special purpose SP's managed network in which servers within either network take part in the call signaling, thereby offering the video service. In addition, typically this type of audio/video service has high business expectations for minimized packet loss, pixilation or other issues with the audio/video experience. In the recent past, entire T3s have been dedicated to a signal Hi-Res call; sometimes one T3 per site of a multi-site video conference.

Hi-Res video conferencing often has larger in bandwidth than the typical video call. The audio portion can be increased as well, as stereo capabilities are often necessary to provide an in-room experience from a distance. The difference can be significant (or another step up from just a typical video service). For example, for a single G.711 audio call that is 80kbps, a Hi-Res conference usually runs G.722 wideband audio at 256kbps. Typical video delivery is up to 4Mbps, whereas a Hi-Res conference can have three 1080p/30fps widescreen displays requiring at least 12Mbps, with a burst capability of much more.

If there were no congestion on the wire, the expected treatment between a video service and a Hi-Res conference would be the same. However, it is typically the case that the Hi-Res conferencing flows have more rigid requirements for quality and business-wise, need to be experience far less errors than the regular video service on the same network.

Note that it is likely the case in these networks that the accompanying audio to the Hi-Res video call will be marked as the Hi-Res video is marked (i.e., using the same DSCP).

The Hi-Res service class MUST use the Class Selector 5 (CS4) PHB, defined in [RFC2474], for non-capacity-admitted conferences. While the capacity-admitted Hi-Res conferences MUST use the CS4-Admit PHB, defined in [ID-DSCP]. This service class MUST be configured to provide a bandwidth assurance for CS4 and CS4-Admit marked packets to ensure that they get forwarded. The Hi-Res service class SHOULD be configured to use a Priority Queuing system such as that defined in Section 1.4.1.1 of this document. Further, CS4-Admit will be designated as the DSCP for use when capacity-admission signaling has been used, such as RSVP or NSIS, to guarantee delivery through the network. CS4 will be used for non-admitted Hi-Res conferences, as well as overflows from CS4-Admit sources that send more packets than they have negotiated bandwidth for that call.

The following applications MUST use the Hi-Res service class:

- o SIP and H.323/V2 (and later) versions of Hi-Res video conferencing applications (interactive Hi-Res video).



- o Video conferencing applications with rate control or traffic content importance marking.

The following are traffic characteristics:

- o Variable size packets.
- o The higher the resolution or change rate between each image, the higher the duration of large packets.
- o Usually constant inter-packet time interval.
- o Can be Variable rate in transmission.
- o Source is capable of reducing its transmission rate based on being told receiver is detecting packet loss.

Applications or IP end points SHOULD pre-mark their packets with DSCP values as shown below. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475] and mark all packets as AF4x.

Mandatory DSCP marking when performed by router closest to source:

- o CS4-Admit = up to specified rate "A", which is dedicated to capacity-admitted Hi-Res traffic.

Note the audio of an A/V call can be marked CS4-Admit as well.

- o CS4 = all video traffic marked CS4-Admit in excess of specified rate "A", or new non-admitted video traffic but below specified rate "B".
- o Where "A" < "B".

Note: One might expect "A" to approximate the peak rates of sum of all admitted video flows, plus the sum of the mean rates and "B" to approximate the sum of the peak rates of those same two flows.

Mandatory DSCP marking when performed by SIP or H.323/V2 videoconferencing equipment:

- o CS4-Admit = SIP or H.323 video conferencing audio stream RTP/UDP.
- o CS4-Admit = SIP or H.323 video conferencing video control RTCP/TCP.
- o CS4-Admit = SIP or H.323 video conferencing video stream up to specified rate "A".

- o CS4 = SIP or H.323 video conferencing video stream in excess of specified rate "A" but below specified rate "B".
- o Where "A" < "B".

Mandatory conditioning performed at DiffServ network edge:

- o The two-rate, three-color marker SHOULD be configured to provide the behavior as defined in trTCM [RFC2698].
- o If packets are marked by trusted sources or a previously trusted DiffServ domain and the color marking is to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Aware mode.
- o If the packet marking is not trusted or the color marking is not to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Blind mode.

The fundamental service offered to nonadmitted "Hi-Res" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Hi-Res" traffic is a guaranteed service using in-data-path signaling to ensure expected or timely delivery. Capacity-admitted video service SHOULD NOT result in packet loss. However, administratively this MAY be allowed to cause a purposeful downspeeding event (i.e., a change in resolution or a change in codec) to occur.

#### 4.2. Realtime-Interactive Service Class

The Realtime-Interactive service class is for bidirectional applications that require low loss and jitter and very low delay for constant or variable rate inelastic traffic sources. Interactive gaming applications that do not have the ability to change encoding rates or to mark packets with different importance indications is one good example of such an application. Another set of applications is virtualized desktop applications in which a remote user has a keyboard, mouse and display monitor, but the desktop is virtualized with the memory/processor/applications back in a common data center, requiring near instantaneous feedback on the user's monitor of any changes caused by the application or an action by the user. Rich media protocols for voice and video MUST NOT use the Realtime-Interactive service class, but rather the appropriate service class from the Conversational service group discussed early in Section 4.1.

The Realtime-Interactive service class will use two PHBs:

Nonadmitted Realtime-Interactive traffic - MUST use the CS5 DSCP [RFC2474], and is for traffic that has not had any capacity

admission signaling performed for that flow or session.

Capacity-Admitted Realtime-Interactive traffic - MUST use the CS5-Admit DSCP [ID-DSCP], and is for traffic that has had any capacity admission signaling performed for that flow or session, e.g., RSVP [RFC2205] or NSIS [RFC4080].

The capacity-admitted Realtime-Interactive traffic operation is similar to an ATM CBR service, which has guaranteed bandwidth and which, if it stays within the negotiated rate, experiences nominal delay and no loss.

Either of the above service classes can be configured as EF based by using a relaxed performance parameter and a rate scheduler.

When a user/endpoint has been authorized to start a new session (i.e., joins a networked game or logs onto a virtualized workstation), the admission procedure should have verified that the newly admitted data rates will be within the engineered capacity of the Realtime-Interactive service class. The bandwidth in the core network and the number of simultaneous Realtime-Interactive sessions that can be supported SHOULD be engineered to control traffic load for this service.

This service class SHOULD be configured to provide a high assurance for bandwidth for CS5 PHB, defined in [RFC2474], or CS5-Admit [ID-DSCP] for guaranteed service through a capacity-admission signaling protocol. The Realtime-Interactive service class SHOULD be configured to use a Rate Queuing system such as that defined in Section 1.4.1.2 of this document. Note that either Realtime-Interactive PHB MAY be configured as another EF PHB, specifically CS5-Admit, that uses a relaxed performance parameter and a rate scheduler, in the priority queue as defined in Section 1.4.1.1 of this document.

The following applications MUST use the Realtime-Interactive service class:

- o Interactive gaming and control.
- o Remote Desktop applications
- o Virtualized Desktop applications.
- o Application server-to-application server non-bursty data transfer requiring very low delay.
- o Inelastic, interactive, time-critical, and mission-critical applications requiring very low delay.

The following are traffic characteristics:

- o Variable size packets.
- o Variable rate, though sometimes bursty, which will require engineering of the network to accommodate.
- o Application is sensitive to delay variation between flows and sessions.
- o Lost packets, if any, are usually ignored by application.

RECOMMENDED DSCP marking:

- o All non-admitted flows in this service class are marked with CS5 (Class Selector 5).
- o All capacity-admitted flows in this service class are marked with CS5-Admit.

Applications or IP end points SHOULD pre-mark their packets with CS5 or CS5-Admit DSCP value, depending on whether a capacity-admission signaling protocol is used for a flow. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

RECOMMENDED conditioning performed at DiffServ network edge:

- o Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods defined in [RFC2475].
- o Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the traffic stays within its negotiated or engineered bounds.
- o Packet flows from trusted sources (application servers inside administered network) MAY not require policing.
- o Policing of packet flows across peering points MUST adhere to the Service Level Agreement (SLA).

The fundamental service offered to nonadmitted "Realtime-Interactive" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Realtime-Interactive" traffic is a guaranteed service using in-data-path signaling to ensure expected or timely delivery. Capacity-admitted Realtime-Interactive service SHOULD NOT result in packet loss. The service SHOULD be engineered so that CS5 marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Normally, traffic in this

service class does not respond dynamically to packet loss. As such, Active Queue Management [RFC2309] SHOULD NOT be applied to CS5 marked packet flows.

#### 4.3. Multimedia Conferencing Service Class

The Multimedia Conferencing service class is for applications that have a low to medium tolerance to delay, and are rate adaptive to lost packets in transit from sources. Presentation Data applications that are operational in conjunction with an audio/video conference is one good example of such an application. Another set of applications is application sharing or whiteboarding applications, also in conjunction to an A/V conference. In either case, the audio & video part of the flow MUST NOT use the Multimedia Conferencing service class, rather the more appropriate service class within the Conversational service group discussed earlier in Section 4.1.

The Multimedia Conferencing service class will use two PHBs:

Nonadmitted Multimedia Conferencing traffic - MUST use the (new) MC DSCP [ID-DSCP], and is for traffic that has not had any capacity admission signaling performed for that flow or session.

Capacity-Admitted Multimedia Conferencing traffic - MUST use the (new) MC-Admit DSCP [ID-DSCP], and is for traffic that has had any capacity admission signaling performed for that flow or session, e.g., RSVP [RFC2205] or NSIS [RFC4080].

The capacity-admitted Multimedia Conferencing traffic operation is similar to an ATM CBR service, which has guaranteed bandwidth and which, if it stays within the negotiated rate, experiences nominal delay and no loss.

When a user/endpoint initiates a presentation data, application sharing or whiteboarding session, it will typically be part of an audio or audio/video conference such as web-conferencing or an existing Telepresence call. The authorization procedure SHOULD be controlled through the coordinated effort to bind the A/V call with the correct Multimedia Conferencing packet flow through some use of identifiers not in scope of this document. The managed network this flow traverse and the number of simultaneous Multimedia Conferencing sessions that can be supported SHOULD be engineered to control traffic load for this service.

The non-capacity admitted Multimedia Conferencing service class SHOULD use the new MC PHB, defined in [ID-DSCP]. This service class SHOULD be configured to provide a high assurance for bandwidth for CS5 marked packets to ensure that they get forwarded. The Multimedia Conferencing service class SHOULD be configured to use a

Rate Queuing system such as that defined in Section 1.4.1.2 of this document. Note that this service class MAY be configured as another EF PHB that uses a relaxed performance parameter, a rate scheduler, and MC-Admit DSCP value, which MUST use the priority queue as defined in Section 1.4.1.1 of this document.

The following applications MUST use the Multimedia Conferencing service class:

- o Presentation Data applications, which can utilize vector graphics, raster graphics or video delivery.
- o Virtualized Desktop applications.
- o Application server-to-application server non-bursty data transfer requiring very low delay.

The following are traffic characteristics:

- o Variable size packets.
- o Variable rate, though sometimes bursty, which will require engineering of the network to accommodate.
- o Application is sensitive to delay variation between flows and sessions.
- o Lost packets, if any, can be ignored by the application.

RECOMMENDED DSCP marking:

- o All non-admitted flows in this service class are marked with the new MC DSCP.
- o All capacity-admitted flows in this service class are marked with MC-Admit.

Applications or IP end points SHOULD pre-mark their packets with the MC DSCP value. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

RECOMMENDED conditioning performed at DiffServ network edge:

- o Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods defined in [RFC2475].
- o Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the traffic

stays within its negotiated or engineered bounds.

- o Packet flows from trusted sources (application servers inside administered network) MAY not require policing.
- o Policing of packet flows across peering points MUST adhere to the Service Level Agreement (SLA).

The fundamental service offered to nonadmitted "Multimedia Conferencing" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Multimedia Conferencing" traffic is a guaranteed service using in-data-path signaling to ensure expected or timely delivery. Capacity-admitted Multimedia Conferencing service SHOULD NOT result in packet loss. The service SHOULD be engineered so that Multimedia Conferencing marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Normally, traffic in this service class does not respond dynamically to packet loss. As such, Active Queue Management [RFC2309] SHOULD NOT be applied to MC or MC-Admit marked packet flows.

#### 4.4. Multimedia Streaming Service Class

The Multimedia Streaming service class is RECOMMENDED for applications that require near-real-time packet forwarding of variable rate elastic traffic sources that are not as delay sensitive as applications using the Broadcast service class. Such applications include streaming audio and video, some video (movies) on-demand applications, and non-interactive webcasts. In general, the Multimedia Streaming service class assumes that the traffic is buffered at the source/destination; therefore, it is less sensitive to delay and jitter.

The Multimedia Streaming service class MUST use the Assured Forwarding (AF3x) PHB, defined in [RFC2597]. This service class MUST be configured to provide a minimum bandwidth assurance for AF31, AF32, and AF33 marked packets to ensure that they get forwarded. The Multimedia Streaming service class SHOULD be configured to use Rate Queuing system for AF32 and AF33 traffic flows, such as that defined in Section 1.4.1.2 of this document. However, AF31 MUST be designated as the DSCP for use when capacity-admission signaling has been used, such as RSVP or NSIS, to guarantee delivery through the network. AF32 and AF33 will be used for non-admitted streaming flows, as well as overflows from AF31 sources that send more packets than they have negotiated bandwidth for that call.

The following applications SHOULD use the Multimedia Streaming service class:

- o Buffered streaming audio (unicast).

- o Buffered streaming video (unicast).
- o Non-interactive Webcasts.
- o IP VPN service that specifies two rates and is less sensitive to delay and jitter.

The following are traffic characteristics:

- o Variable size packets.
- o The higher the rate, the higher the density of large packets.
- o Variable rate.
- o Elastic flows.
- o Some bursting at start of flow from some applications, as well as an expected stepping up and down on the rate of the flow based on changes in resolution due to network conditions.

Applications or IP end points SHOULD pre-mark their packets with DSCP values as shown below. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475], and mark all packets as AF3x. Note: In this case, the two-rate, three-color marker will be configured to operate in Color-Blind mode.

RECOMMENDED DSCP marking:

- o AF31 = up to specified rate "A".
- o AF32 = all traffic marked AF31 in excess of specified rate "A", or new AF32 traffic but below specified rate "B".
- o AF33 = in excess of specified rate "B".
- o Where "A" < "B".

Note: One might expect "A" to approximate the peak rates of sum of all streaming flows, plus the sum of the mean rates and "B" to approximate the sum of the peak rates of those same two flows.

RECOMMENDED conditioning performed at DiffServ network edge:

- o The two-rate, three-color marker SHOULD be configured to provide the behavior as defined in trTCM [RFC2698].
- o If packets are marked by trusted sources or a previously trusted DiffServ domain and the color marking is to be preserved, then



the two-rate, three-color marker SHOULD be configured to operate in Color-Aware mode.

- o If the packet marking is not trusted or the color marking is not to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Blind mode.

The fundamental service offered to nonadmitted "Multimedia Streaming" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Multimedia Streaming" traffic is a guaranteed service using in-data-path signaling to ensure expected delivery in a reasonable manner. The service SHOULD be engineered so that AF31 marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Since the AF3x traffic is elastic and responds dynamically to packet loss, Active Queue Management [RFC2309] SHOULD be used primarily to reduce forwarding rate to the minimum assured rate at congestion points, unless AF31 has had a capacity-admission signaling protocol applied to the flow, such as RSVP or NSIS.

If a capacity-admission signaling protocol applied to the AF31 flow, which SHOULD be the case always, the AF31 PHB MAY be configured as another EF PHB that uses a relaxed performance parameter and a rate scheduler, in the priority queue as defined in Section 1.4.1.1 of this document.

The probability of loss of AF31 traffic MUST NOT exceed the probability of loss of AF32 traffic, which in turn MUST NOT exceed the probability of loss of AF33.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth for each DSCP, and the max-threshold specifies the queue depth above which all traffic with such a DSCP is dropped or ECN marked. Thus, in this service class, the following inequality MUST hold in queue configurations:

- o min-threshold AF33 < max-threshold AF33
- o max-threshold AF33 <= min-threshold AF32
- o min-threshold AF32 < max-threshold AF32
- o max-threshold AF32 <= min-threshold AF31
- o min-threshold AF31 < max-threshold AF31
- o max-threshold AF31 <= memory assigned to the queue

Note#1: this confirmation MUST be modified if AF31 has a capacity-admission signaling protocol applied to those flows, and the above will only apply to AF32 and AF33, while

AF31 (theoretically) has no packet loss.

Note#2: This configuration tends to drop AF33 traffic before AF32 and AF32 before AF31. Note: Many other AQM algorithms exist and are used; they SHOULD be configured to achieve a similar result.

#### 4.5. Broadcast Service Class

The Broadcast service class is RECOMMENDED for applications that require near-real-time packet forwarding with very low packet loss of constant rate and variable rate inelastic traffic sources that are more delay sensitive than applications using the Multimedia Streaming service class. Such applications include broadcast TV, streaming of live audio and video events, some video-on-demand applications, and video surveillance. In general, the Broadcast service class assumes that the destination end point has a dejitter buffer, for video application usually a 2 - 8 video-frame buffer (66 to several hundred of milliseconds), thus expecting far less buffering before play-out than Multimedia Streaming, which can buffer in the seconds to minutes (to hours).

The Broadcast service class will use two PHBs:

Nonadmitted Broadcast traffic - MUST use the CS3 DSCP [RFC2474], and is for traffic that has not had any capacity admission signaling performed for that flow or session.

Capacity-Admitted Broadcast traffic - MUST use the CS3-Admit DSCP [ID-DSCP], and is for traffic that has had any capacity admission signaling performed for that flow or session, e.g., RSVP [RFC2205] or NSIS [RFC4080].

The capacity-admitted Broadcast traffic operation is similar to an ATM CBR service, which has guaranteed bandwidth and which, if it stays within the negotiated rate, experiences nominal delay and no loss.

Either of the above service classes can be configured as EF based by using a relaxed performance parameter and a rate scheduler.

When a user/endpoint initiates a new Broadcast session (i.e., starts an Internet radio application, starts a live Internet A/V event or a camera comes online to do video-surveillance), the admission procedure should be verified within the application that triggers the flow. The newly admitted data rates will SHOULD be within the engineered capacity of the Broadcast service class within that network. The bandwidth in the core network and the number of simultaneous Broadcast sessions that can be supported SHOULD be engineered to control traffic load for this service.

This service class SHOULD be configured to provide high assurance for bandwidth for CS3 marked packets to ensure that they get forwarded. The Broadcast service class SHOULD be configured to use Rate Queuing system such as that defined in Section 1.4.1.2 of this document. Note that either Broadcast PHB MAY be configured as another EF PHB, specifically CS3-Admit, that uses a relaxed performance parameter and a rate scheduler, in the priority queue as defined in Section 1.4.1.1 of this document.

The following applications SHOULD use the Broadcast service class:

- o Video surveillance and security (unicast).
- o TV broadcast including HDTV (likely multicast, but can be unicast).
- o Video on demand (unicast) with control (virtual DVD).
- o Streaming of live audio events (both unicast and multicast).
- o Streaming of live video events (both unicast and multicast).

The following are traffic characteristics:

- o Variable size packets.
- o The higher the rate, the higher the density of large packets.
- o Mixture of variable rate and constant rate flows.
- o Fixed packet emission time intervals.
- o Inelastic flows.

RECOMMENDED DSCP marking:

- o All non-admitted flows in this service class are marked with CS3 (Class Selector 3).
- o All capacity-admitted flows in this service class are marked with CS3-Admit.
- o In some cases, such as those for security and video surveillance applications, it is NOT RECOMMENDED, but allowed to use a different DSCP marking.

If so, then locally user definable (EXP/LU) codepoints in the range '011x11' MAY be used to provide unique traffic identification. The locally administrator definable (EXP/LU, from pool 2 of RFC 2474) codepoint(s) MAY be associated with the PHB that is used for CS3 or CS3-Admit traffic. Furthermore, depending on the network scenario, additional network edge

conditioning policy MAY be needed for the EXP/LU codepoint(s) used.

Applications or IP end points SHOULD pre-mark their packets with CS3 or CS3-Admit DSCP value. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

RECOMMENDED conditioning performed at DiffServ network edge:

- o Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods defined in [RFC2475].
- o Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the traffic stays within its negotiated or engineered bounds.
- o Packet flows from trusted sources (application servers inside administered network) MAY not require policing.
- o Policing of packet flows across peering points MUST be performed to the Service Level Agreement (SLA) of those peering entities.

The fundamental service offered to "Broadcast" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Broadcast" traffic is a guaranteed service using in-data-path signaling to ensure expected or timely delivery. Capacity-admitted Broadcast service SHOULD NOT result in packet loss. The service SHOULD be engineered so that CS3 and CS3-Admit marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Normally, traffic in this service class does not respond dynamically to packet loss. As such, Active Queue Management [RFC2309] SHOULD NOT be applied to CS3 marked packet flows.

#### 4.6. Low-Latency Data Service Class

The Low-Latency Data service class is RECOMMENDED for elastic and responsive typically client-/server-based applications. Applications forwarded by this service class are those that require a relatively fast response and typically have asymmetrical bandwidth need, i.e., the client typically sends a short message to the server and the server responds with a much larger data flow back to the client. The most common example of this is when a user clicks a hyperlink (~ few dozen bytes) on a web page, resulting in a new web page to be loaded (Kbytes or MBs of data). This service class is configured to provide good response for TCP [RFC1633] short-lived flows that require real-time packet forwarding of variable rate

traffic sources.

The Low-Latency Data service class SHOULD use the Assured Forwarding (AF) PHB, defined in [RFC2597]. This service class SHOULD be configured to provide a minimum bandwidth assurance for AF21, AF22, and AF23 marked packets to ensure that they get forwarded. The Low-Latency Data service class SHOULD be configured to use a Rate Queuing system such as that defined in Section 1.4.1.2 of this document.

The following applications SHOULD use the Low-Latency Data service class:

- o Client/server applications.
- o Systems Network Architecture (SNA) terminal to host transactions (SNA over IP using Data Link Switching (DLSw)).
- o Web-based transactions (E-commerce).
- o Credit card transactions.
- o Financial wire transfers.
- o Enterprise Resource Planning (ERP) applications (e.g., SAP/BaaN).
- o VPN service that supports Committed Information Rate (CIR) with up to two burst sizes.
- o Instant Messaging and Presence protocols (e.g., SIP, XMPP).

The following are traffic characteristics:

- o Variable size packets.
- o Variable packet emission rate.
- o With packet bursts of TCP window size.
- o Short traffic bursts.
- o Source capable of reducing its transmission rate based on detection of packet loss at the receiver or through explicit congestion notification.

Applications or IP end points SHOULD pre-mark their packets with DSCP values as shown below. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475] and mark all packets as AF2x. Note: In this case, the single-rate, three-color marker will be configured to operate in Color-Blind mode.

RECOMMENDED DSCP marking:

- o AF21 = flow stream with packet burst size up to "A" bytes.
- o AF22 = flow stream with packet burst size in excess of "A" but below "B" bytes.
- o AF23 = flow stream with packet burst size in excess of "B" bytes.
- o Where "A" < "B".

RECOMMENDED conditioning performed at DiffServ network edge:

- o The single-rate, three-color marker SHOULD be configured to provide the behavior as defined in srTCM [RFC2697].
- o If packets are marked by trusted sources or a previously trusted DiffServ domain and the color marking is to be preserved, then the single-rate, three-color marker SHOULD be configured to operate in Color-Aware mode.
- o If the packet marking is not trusted or the color marking is not to be preserved, then the single-rate, three-color marker SHOULD be configured to operate in Color-Blind mode.

The fundamental service offered to "Low-Latency Data" traffic is enhanced best-effort service with controlled rate and delay. The service SHOULD be engineered so that AF21 marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Since the AF2x traffic is elastic and responds dynamically to packet loss, Active Queue Management [RFC2309] SHOULD be used primarily to control TCP flow rates at congestion points by dropping packets from TCP flows that have large burst size. The probability of loss of AF21 traffic MUST NOT exceed the probability of loss of AF22 traffic, which in turn MUST NOT exceed the probability of loss of AF23. Explicit Congestion Notification (ECN) [RFC3168] MAY also be used with Active Queue Management.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth for each DSCP, and the max-threshold specifies the queue depth above which all traffic with such a DSCP is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- o min-threshold AF23 < max-threshold AF23
- o max-threshold AF23 <= min-threshold AF22
- o min-threshold AF22 < max-threshold AF22
- o max-threshold AF22 <= min-threshold AF21

- o min-threshold AF21 < max-threshold AF21
- o max-threshold AF21 <= memory assigned to the queue

Note: This configuration tends to drop AF23 traffic before AF22 and AF22 before AF21. Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

#### 4.7. Conversational Signaling Service Class

The Signaling service class is MUST be limited to delay-sensitive signaling traffic only, and then only applying to signaling that involves the Conversational service group. Audio signaling includes signaling between IP phone and soft-switch, soft-client and soft-switch, and media gateway and soft-switch as well as peer-to-peer using various protocols. Video and Hi-Res signaling includes video endpoint to video endpoint, as well as to Media transfer Point (MTP), to call control server(S), etc. This service class is intended to be used for control of voice and video sessions and applications. Protocols using this service class require a relatively fast response, as there are typically several messages of different sizes sent for control of the session. This service class is configured to provide good response for short-lived, intermittent flows that require real-time packet forwarding. This is not the service class for Instant Messaging (IM), that's within the bounds of the Low-Latency Data service class. The Conversational Signaling service class MUST be configured so that the probability of packet drop or significant queuing delay under peak load is very low in IP network segments that provide this interface.

The Conversational Signaling service class MUST use the new A/V-Sig PHB, defined in [ID-DSCP]. This service class MUST be configured to provide a minimum bandwidth assurance for A/V-Sig marked packets to ensure that they get forwarded. In other words, this service class MUST NOT be starved from transmission within a reasonable timeframe, given that the entire Conversational service group depends on these signaling messages successful delivery. Network engineering SHOULD be done to ensure there is roughly 1-4% available per node interface that audio and video traverse. Local conditions MUST be considered when determining exactly how much bandwidth is given to this service class. The Conversational Signaling service class SHOULD be configured to use a Rate Queuing system such as that defined in Section 1.4.1.2 of this document.

The following applications SHOULD use the Conversational Signaling service class:

- o Peer-to-peer IP telephony signaling (e.g., SIP, H.323, XMPP).
- o Peer-to-peer signaling for multimedia applications (e.g., SIP, H.323, XMPP).

- o Peer-to-peer real-time control function.
- o Client-server IP telephony signaling using H.248, MEGACO, MGCP, IP encapsulated ISDN, or other proprietary protocols.
- o Signaling to control IPTV applications using protocols such as IGMP.
- o Signaling flows between high-capacity telephony call servers or soft switches using protocol such as SIP-T. Such high-capacity devices may control thousands of telephony (VoIP) calls.
- o Signaling for one-way video flows, such as RTSP [RFC2326].
- o IGMP, when used for multicast session control such as channel changing in IPTV systems.
- o OPTIONALLY, this service class can be used for on-path reservation signaling for the traffic flows that will use the "admitted" DSCPs. The alternative is to have the on-path signaling (for reservations) use the DSCP within that service class. This provides a similar treatment of the signaling to the data flow, which might be desired.

The following are traffic characteristics:

- o Variable size packets, normally one packet at a time.
- o Intermittent traffic flows.
- o Traffic may burst at times.
- o Delay-sensitive control messages sent between two end points.

RECOMMENDED DSCP marking:

- o All flows in this service class are marked with A/V-Sig.

Applications or IP end points SHOULD pre-mark their packets with A/V-Sig DSCP value. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

RECOMMENDED conditioning performed at DiffServ network edge:

- o Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods defined in [RFC2475].



- o Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the traffic stays within its negotiated or engineered bounds.
- o Packet flows from trusted sources (application servers inside administered network) MAY not require policing.
- o Policing of packet flows across peering points in which each peer is participating in the call set-up MUST be performed to the Service Level Agreement (SLA).

The fundamental service offered to "Conversational Signaling" traffic is enhanced best-effort service with controlled rate and delay. The service SHOULD be engineered so that A/V-Sig marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery and low delay. Normally, traffic in this service class does not respond dynamically to packet loss. As such, Active Queue Management [RFC2309] SHOULD NOT be applied to A/V-Sig marked packet flows.

#### 4.8. High-Throughput Data Service Class

The High-Throughput Data service class is RECOMMENDED for elastic applications that require timely packet forwarding of variable rate traffic sources and, more specifically, is configured to provide good throughput for TCP longer-lived flows. TCP [RFC1633] or a transport with a consistent Congestion Avoidance Procedure [RFC2581] [RFC3782] normally will drive as high a data rate as it can obtain over a long period of time. The FTP protocol is a common example, although one cannot definitively say that all FTP transfers are moving data in bulk.

The High-Throughput Data service class SHOULD use the Assured Forwarding (AF) PHB, defined in [RFC2597]. This service class SHOULD be configured to provide a minimum bandwidth assurance for AF11, AF12, and AF13 marked packets to ensure that they are forwarded in a timely manner. The High-Throughput Data service class SHOULD be configured to use a Rate Queuing system such as that defined in Section 1.4.1.2 of this document.

The following applications SHOULD use the High-Throughput Data service class:

- o Store and forward applications.
- o File transfer applications (e.g., FTP, HTTP, etc).
- o Email.
- o VPN service that supports two rates (committed information rate

and excess or peak information rate).

The following are traffic characteristics:

- o Variable size packets.
- o Variable packet emission rate.
- o Variable rate.
- o With packet bursts of TCP window size.
- o Source capable of reducing its transmission rate based on detection of packet loss at the receiver or through explicit congestion notification.

Applications or IP end points SHOULD pre-mark their packets with DSCP values as shown below. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475], and mark all packets as AF1x. Note: In this case, the two-rate, three-color marker will be configured to operate in Color-Blind mode.

RECOMMENDED DSCP marking:

- o AF11 = up to specified rate "A".
- o AF12 = in excess of specified rate "A" but below specified rate "B".
- o AF13 = in excess of specified rate "B".
- o Where "A" < "B".

RECOMMENDED conditioning performed at DiffServ network edge:

- o The two-rate, three-color marker SHOULD be configured to provide the behavior as defined in trTCM [RFC2698].
- o If packets are marked by trusted sources or a previously trusted DiffServ domain and the color marking is to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Aware mode.
- o If the packet marking is not trusted or the color marking is not to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Blind mode.

The fundamental service offered to "High-Throughput Data" traffic is enhanced best-effort service with a specified minimum rate. The service SHOULD be engineered so that AF11 marked packet flows have

sufficient bandwidth in the network to provide assured delivery. It can be assumed that this class will consume any available bandwidth and that packets traversing congested links may experience higher queuing delays or packet loss. Since the AF1x traffic is elastic and responds dynamically to packet loss, Active Queue Management [RFC2309] SHOULD be used primarily to control TCP flow rates at congestion points by dropping packets from TCP flows that have higher rates first. The probability of loss of AF11 traffic MUST NOT exceed the probability of loss of AF12 traffic, which in turn MUST NOT exceed the probability of loss of AF13. In such a case, if one network customer is driving significant excess and another seeks to use the link, any losses will be experienced by the high-rate user, causing him to reduce his rate. Explicit Congestion Notification (ECN) [RFC3168] MAY also be used with Active Queue Management.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth for each DSCP, and the max-threshold specifies the queue depth above which all traffic with such a DSCP is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- o min-threshold AF13 < max-threshold AF13
- o max-threshold AF13 <= min-threshold AF12
- o min-threshold AF12 < max-threshold AF12
- o max-threshold AF12 <= min-threshold AF11
- o min-threshold AF11 < max-threshold AF11
- o max-threshold AF11 <= memory assigned to the queue

Note: This configuration tends to drop AF13 traffic before AF12 and AF12 before AF11. Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

#### 4.9. Standard Service Class

The Standard service class is RECOMMENDED for traffic that has not been classified into one of the other supported forwarding service classes in the DiffServ network domain. This service class provides the Internet's "best-effort" forwarding behavior. This service class typically has minimum bandwidth guarantee.

The Standard service class MUST use the Default Forwarding (DF) PHB, defined in [RFC2474], and SHOULD be configured to receive at least a small percentage of forwarding resources as a guaranteed minimum. This service class SHOULD be configured to use a Rate Queuing system such as that defined in Section 1.4.1.2 of this document.

The following applications SHOULD use the Standard service class:

- o Network services, DNS, DHCP, BootP.
- o Any undifferentiated application/packet flow transported through the DiffServ enabled network.

The following is a traffic characteristic:

- o Non-deterministic, mixture of everything.

The RECOMMENDED DSCP marking is DF (Default Forwarding) '000000'.

Network Edge Conditioning:

There is no requirement that conditioning of packet flows be performed for this service class.

The fundamental service offered to the Standard service class is best-effort service with active queue management to limit overall delay. Typical configurations SHOULD use random packet dropping to implement Active Queue Management [RFC2309] or Explicit Congestion Notification [RFC3168], and MAY impose a minimum or maximum rate on the queue.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth, and the max-threshold specifies the queue depth above which all traffic is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- o min-threshold DF < max-threshold DF
- o max-threshold DF <= memory assigned to the queue

Note: Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

#### 4.10. Low-Priority Data

The Low-Priority Data service class serves applications that run over TCP [RFC0793] or a transport with consistent congestion avoidance procedures [RFC2581] [RFC3782] and that the user is willing to accept service without guarantees. This service class is specified in [RFC3662] and [QBSS].

The following applications MAY use the Low-Priority Data service class:

- o Any TCP based-application/packet flow transported through the DiffServ enabled network that does not require any bandwidth assurances.

The following is a traffic characteristic:

- o Non-real-time and elastic.

Network Edge Conditioning:

There is no requirement that conditioning of packet flows be performed for this service class.

The RECOMMENDED DSCP marking is CS1 (Class Selector 1).

The fundamental service offered to the Low-Priority Data service class is best-effort service with zero bandwidth assurance. By placing it into a separate queue or class, it may be treated in a manner consistent with a specific Service Level Agreement.

Typical configurations SHOULD use Explicit Congestion Notification [RFC3168] or random loss to implement Active Queue Management [RFC2309].

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth, and the max-threshold specifies the queue depth above which all traffic is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- o min-threshold CS1 < max-threshold CS1
- o max-threshold CS1 <= memory assigned to the queue

Note: Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

## 5. Additional Information on Service Class Usage

In this section, we provide additional information on how some specific applications should be configured to use the defined service classes.

### 5.1. Mapping for NTP

From tests that were performed, indications are that precise time distribution requires a very low packet delay variation (jitter) transport. Therefore, we suggest that the following guidelines for Network Time Protocol (NTP) be used:

- o When NTP is used for providing high-accuracy timing within an administrator's (carrier's) network or to end users/clients, the audio service class SHOULD be used, and NTP packets should be marked with EF DSCP value.

- o For applications that require "wall clock" timing accuracy, the Standard service class should be used, and packets should be marked with DF DSCP.

## 5.2. VPN Service Mapping

"Differentiated Services and Tunnels" [RFC2983] considers the interaction of DiffServ architecture with IP tunnels of various forms. Further to guidelines provided in RFC 2983, below are additional guidelines for mapping service classes that are supported in one part of the network into a VPN connection. This discussion is limited to VPNs that use DiffServ technology for traffic differentiation.

- o The DSCP value(s) that is/are used to represent a PHB or a PHB group SHOULD be the same for the networks at both ends of the VPN tunnel, unless remarking of DSCP is done as ingress/egress processing function of the tunnel. DSCP marking needs to be preserved along the tunnel, end to end.
- o The VPN MAY be configured to support one or more service classes. It is left up to the administrators of the two networks to agree on the level of traffic differentiation that will be provided in the network that supports VPN service. Service classes are then mapped into the supported VPN traffic forwarding behaviors that meet the traffic characteristics and performance requirements of the encapsulated service classes.
- o The traffic treatment in the network that is providing the VPN service needs to be such that the encapsulated service class or classes receive comparable behavior and performance in terms of delay, jitter, and packet loss and that they are within the limits of the service specified.
- o The DSCP value in the external header of the packet forwarded through the network providing the VPN service can be different from the DSCP value that is used end to end for service differentiation in the end network.
- o The guidelines for aggregation of two or more service classes into a single traffic forwarding treatment in the network that is providing the VPN service is for further study.

## 6. Security Considerations

This document discusses policy and describes a common policy configuration, for the use of a Differentiated Services Code Point by transports and applications. If implemented as described, it should require that the network do nothing that the network has not already allowed. If that is the case, no new security issues should arise from the use of such a policy.

It is possible for the policy to be applied incorrectly, or for a wrong policy to be applied in the network for the defined service class. In that case, a policy issue exists that the network SHOULD detect, assess, and deal with. This is a known security issue in any network dependent on policy-directed behavior.

A well-known flaw appears when bandwidth is reserved or enabled for a service (for example, voice and/or video transport) and another service or an attacking traffic stream uses it. This possibility is inherent in DiffServ technology, which depends on appropriate packet markings. When bandwidth reservation or a priority queuing system is used in a vulnerable network, the use of authentication and flow admission is recommended. To the author's knowledge, there is no known technical way to respond to an unauthenticated data stream using service that it is not intended to use, and such is the nature of the Internet.

The use of a service class by a user is not an issue when the SLA between the user and the network permits him to use it, or to use it up to a stated rate. In such cases, simple policing is used in the Differentiated Services Architecture. Some service classes, such as Network Control, are not permitted to be used by users at all; such traffic should be dropped or remarked by ingress filters. Where service classes are available under the SLA only to an authenticated user rather than to the entire population of users, authentication and authorization services are required, such as those surveyed in [AUTHMECH].

## 7. Contributing Authors

This section specifically calls out the authors of RFC 4594, from which this document is based on.

Jozef Babiarez  
Nortel Networks

Kwok Ho Chan  
Nortel Networks  
Email: khchan.work@gmail.com

Fred Baker  
Cisco Systems  
EMail: fred@cisco.com

Of note, two of the three mentioned authors above worked for Nortel Networks at the time of writing RFC 4594, a company that no longer exists. This author has not seen or heard from those two in many, many years or IETF meetings - as a result of not knowing their new email addresses (or phone numbers).

While much of this document has been rewritten with either edited or

brand new material, there are many short paragraphs that remain as they were from RFC 4594, as well as many sentences that were also left unchanged. Additionally, there were no new graphs, charts, diagrams, or tables introduced, meaning the first 4 tables within this document existed in RFC 4594, created by those authors. Presently, each of those tables contain modified and new information. The last 3 tables, specifically tables 5, 6, & 7 were removed because the examples section was removed.

This author believes there must be proper credit given for all the contributions, including the framework this document retains from that RFC. Periodically, throughout this document, what was written remains the best way of conveying a thought, rule, or otherwise stated behavior or mechanism. Because RFC 4594 was rather large, there is no realistic way of identifying each part that was left untouched. Further, properly quoting that RFC and leaving those sentences embedded in this document would render this document highly unreadable. Another application could be used to show the changes, deletions and additions - but not one that the IETF accepts presently.

This author has created this "Contributing Authors" section as a way of properly identifying those 3 individuals that provided text within this document. We will let the community judge if this is 'good enough' (i.e., rough consensus), or if another way is better.

## 8. Acknowledgements

The author would like to thank Paul Jones, Glen Lavers, Mo Zanaty, David Benham, Michael Ramalho, Gorrry Fairhurst, David Black, Brian Carpenter, Al Morton, Ruediger Geib and Shitanshu Shah for their comments and questions about this effort that ultimately helped shape this document.

Below are the folks that were acknowledged in RFC 4594, and this author does not want to lose their recognition of contributions to the original effort.

"The authors thank the TSVWG reviewers, David Black, Brian E. Carpenter, and Alan O'Neill for their review and input to this document.

The authors acknowledge a great many inputs, most notably from Bruce Davie, Dave Oran, Ralph Santitoro, Gary Kenward, Francois Audet, Morgan Littlewood, Robert Milne, John Shuler, Nalin Mistry, Al Morton, Mike Pierce, Ed Koehler Jr., Tim Rahrer, Fil Dickinson, Mike Fidler, and Shane Amante. Kimberly King, Joe Zebarth, and Alistair Munroe each did a thorough proofreading,



and the document is better for their contributions."

## 9. References

### 9.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.
- [RFC1349] Almquist, P., "Type of Service in the Internet Protocol Suite", RFC 1349, July 1992.
- [RFC1812] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Service", RFC 2475, December 1998.
- [RFC2597] Heinanen, J., Baker, F., Weiss, W., and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.
- [RFC3246] Davie, B., Charny, A., Bennet, J.C., Benson, K., Le Boudec, J., Courtney, W., Davari, S., Firoiu, V., and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3662] Bless, R., Nichols, K., and K. Wehrle, "A Lower Effort Per-Domain Behavior (PDB) for Differentiated Services",

RFC 3662, December 2003.

- [RFC5865] F. Baker, J. Polk, M. Dolly, "A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic", RFC 5865, May 2010

## 9.2. Informative References

- [AUTHMECH] Rescorla, E., "A Survey of Authentication Mechanisms", Work in Progress, September 2005.
- [QBSS] "QBone Scavenger Service (QBSS) Definition", Internet2 Technical Report Proposed Service Definition, March 2001.
- [IEEE1Q] IEEE, 802.1Q Specification
- [IEEE1E] IEEE, 802.1E Wireless LAN User Priority Specification
- [RFC1633] Braden, R., Clark, D., and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994.
- [RFC2205] Braden, R., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.
- [RFC2581] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", RFC 2581, April 1999.
- [RFC2697] Heinanen, J. and R. Guerin, "A Single Rate Three Color Marker", RFC 2697, September 1999.
- [RFC2698] Heinanen, J. and R. Guerin, "A Two Rate Three Color Marker", RFC 2698, September 1999.
- [RFC2963] Bonaventure, O. and S. De Cnodder, "A Rate Adaptive Shaper for Differentiated Services", RFC 2963, October 2000.
- [RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.
- [RFC2996] Bernet, Y., "Format of the RSVP DCLASS Object", RFC 2996, November 2000.
- [RFC3086] Nichols, K. and B. Carpenter, "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification", RFC 3086, April 2001.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC

3168, September 2001.

- [RFC3175] Baker, F., Iturralde, C., Le Faucheur, F., and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", RFC 3175, September 2001.
- [RFC3290] Bernet, Y., Blake, S., Grossman, D., and A. Smith, "An Informal Management Model for Diffserv Routers", RFC 3290, May 2002.
- [RFC3782] Floyd, S., Henderson, T., and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 3782, April 2004.
- [RFC5462] L. Andersson, R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: EXP Field Renamed to Traffic Class Field", RFC 5462, February 2009

#### Authors' Address

James Polk  
3913 Treemont Circle  
Colleyville, Texas 76034

Phone: +1.817.271.3552  
Email: jmpolk@cisco.com

#### Appendix A - Changes

Here is a list of all the changes that were captured during the editing process. This will not be a complete list, and others are free to point out what the authors missed, and we'll include that in the next release.

##### A.1 Since Individual -02 to -03

- o Inserted section 1.6 to explain fundamentally what has changed since RFC 4594, and why changes are necessary.

##### A.2 Since Individual -01 to -02

- o Added text to the Intro section on the justification from DiffServ Problem Statement draft, as to more of why this update is necessary.
- o Added text to the Intro section expanding on the concept of service classes vs. treatment aggregates (from RFC 5127).

## A.3 Since Individual -00 to -01

- o Added Section 2.4 which covers the conflation issues regarding the differences between service classes and treatment aggregates.
- o Added example operational configurations of treatment aggregates applied to this draft's new set of service classes and additional DSCPs.
- o Added references RFC 5865, RFC 5462, IEEE 802.1E and IEEE 802.1Q.

## A.4 Since RFC 4594 to Individual Update -00

- o rewrote Intro to emphasize current topics
- o Created a Conversational Service group, comprising the audio, video and Hi-Res service classes - because they have similar characteristics.
- o Incorporated the 6 new DSCPs from [ID-DSCP].
- o moved the example section, en mass, to an appendix that might not be kept for this version. We're not sure it accomplishes what it needs to, and might not provide any real usefulness.
- o Moved 'Realtime-Interactive' service class to CS5, from CS4
- o Changed 'Broadcast Video' service class to 'Broadcast' service class
- o Changed AF4X to 'Video' service class, replacing 'Multimedia Conferencing' service class
- o Moved 'Multimedia Conferencing' service class to different DSCPs
- o Added the 'Hi-Res' service class
- o Removed section 5.1 on signaling choices. It has been included in the main body of the text.
- o Changed document title
- o ...

Network WG  
Internet-Draft  
Expires: January 16, 2013  
Intended Status: Standards Track  
Updates: RFC 2872 (if accepted)

James Polk  
Subha Dhesikan  
Cisco Systems  
July 16, 2012

Resource Reservation Protocol (RSVP) Application-ID  
Profiles for Voice and Video Streams  
draft-polk-tsvwg-rsvp-app-id-vv-profiles-04

Abstract

RFC 2872 defines an Resource Reservation Protocol (RSVP) object for application identifiers. This document uses that App-ID and gives implementers specific guidelines for differing voice and video stream identifications to nodes along a reservation path, creating specific profiles for voice and video session identification.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Application ID Template . . . . .	3
3. The Voice and Video Application-ID Profiles . . . . .	4
3.1 The Broadcast video Profile . . . . .	4
3.2 The Real-time Interactive Profile . . . . .	5
3.3 The Multimedia Conferencing Profile . . . . .	5
3.4 The Multimedia Streaming Profile . . . . .	6
3.5 The Conversational Profile . . . . .	6
4. Security considerations . . . . .	7
5. IANA considerations . . . . .	7
5.1 New RSVP Policy Element (P-Type) . . . . .	7
5.2 Application Profiles . . . . .	7
5.2.1 Broadcast Profiles IANA Registry . . . . .	8
5.2.2 Realtime-Interactive Profiles IANA Registry . . . . .	8
5.2.3 Multimedia-Conferencing Profiles IANA Registry . . . . .	9
5.2.4 Multimedia-Streaming Profiles IANA Registry . . . . .	10
5.2.5 Conversational Profiles IANA Registry . . . . .	10
6. Acknowledgments . . . . .	12
7. References . . . . .	12
7.1. Normative References . . . . .	12
7.2. Informative References . . . . .	13
Authors' Addresses . . . . .	13
Appendix . . . . .	14

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC 2119].

## 1. Introduction

RFC 2872 [RFC2872] describes the usage of policy elements for providing application information in Resource Reservation Protocol (RSVP) signaling [RFC2205]. The intention of providing this information is to enable application-based policy control. However, RFC 2872 does not enumerate any application profiles. The absence of explicit, uniform profiles leads to incompatible handling of these values and misapplied policies. An application profile used by a sender might not be understood by the intermediaries or receiver in a different domain. Therefore, there is a need to enumerate application profiles that are universally understood and applied for correct policy control.

Call control between endpoints has the ability to bind or associate many attributes to a reservation. One new attribute currently being defined is to establish the type of traffic contained that reservation. This is accomplished via assigning a traffic label to

the call (or session or flow) [ID-TRAF-CLASS].

This document takes the application traffic classes from [ID-TRAF-CLASS] and places those strings in the APP-ID object defined in RFC 2872. Thus, the intermediary devices (e.g., routers) processing the RSVP message can learn the identified profile within the Application-ID policy element for a particular reservation, and possibly be configured with the profile(s) to understand them correctly, thus performing the correct admission control.

Another goal of this document is to the ability to signal an application profile which can then be translated into a DSCP value as per the choice of each domain. While the DCLASS object [RFC2996] allows the transfer of DSCP value in an RSVP message, it does not allow the flexibility of having different domains choosing the DSCP value for the traffic classes that that they maintain.

How these labels indicate the appropriate Differentiated Services Codepoint (DSCP) is out of scope for this document.

This document will break out each application type and propose how the values in application-id template should be populated for uniformity and interoperability.

## 2. Application ID Template

The template from RFC 2872 is as follows:

0	1	2	3
PE Length (8)		P-type = AUTH_APP	
Attribute Length		A-type = POLICY_LOCATOR	Sub-type = ASCII_DN
Application name as ASCII string (e.g. SAP.EXE)			

In line with how this policy element is constructed in RFC 2872, the A-type will remain "POLICY\_LOCATOR".

The P-type field is first created in [RFC2752]. This document creates the new P-type "APP\_TC" for application traffic class, which is more appropriately named for the purpose described in this extension.

The first Sub-type will be mandatory for every profile within this document, and will be "ASCII\_DN". No other Sub-types are defined by

any profile within this document, but MAY be included by individual implementations - and MUST be ignored if not understood by receiving implementations along the reservation path.

RFC 2872 states the #1 sub-element from RFC 2872 as the "identifier that uniquely identifies the application vendor", which is optional to include. This document modifies this vendor limitation so that the identifier need only be unique - and not limited to an application vendor (identifier). For example, this specification now allows an RFC that defines an industry recognizable term or string to be a valid identifier. For example, a term or string taken from another IETF document, such as "conversational" or "avconf" from [ID-TRAF-CLASS]. This sub-element is still optional to include.

The following subsections will define the values within the above template into specific profiles for voice and video identification.

### 3. The Voice and Video Application-ID Profiles

This section contains the elements of the Application ID policy object which is used to signal the application classes defined in [ID-TRAF-CLASS].

#### 3.1 The Broadcast Profiles

Broadcast profiles are for minimally buffered one-way streaming flows, such as video surveillance, or Internet based concerts or non-VOD TV broadcasts such as live sporting events.

There will be Broadcast profiles for

- Broadcast IPTV for audio and video
- Broadcast Live-events for audio and video
- Broadcast Surveillance for audio and video

Here is an example profile for identifying Broadcast Video-Surveillance

```
APP_TC, POLICY_LOCATOR, ASCII_DN,  
"GUID=http://www.ietf.org/internet-drafts/  
    draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=broadcast.video.surveillance, VER="
```

Where the Globally Unique Identifier (GUID) indicates the documented reference that created this well known string [ID-TRAF-CLASS], the APP is the profile name with no spaces, and the "VER=" is included, but has no value at this time.

#### 3.2 The Realtime Interactive Profiles



Realtime Interactive profiles are for on-line gaming, and both remote and virtual avconf applications, in which the timing is particularly important towards the feedback to uses of these applications. This traffic type will generally not be UDP based, with minimal tolerance to RTT delays.

There will be Realtime Interactive profiles for

- Realtime-Interactive Gaming
- Realtime-Interactive Remote-Desktop
- Realtime-Interactive Virtualized-Desktop

Here is the profile for identifying Realtime-Interactive Gaming

```
APP_TC, POLICY_LOCATOR, ASCII_DN,  
"GUID=http://www.ietf.org/internet-drafts/  
    draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=realtime-interactive.gaming, VER="
```

Where the Globally Unique Identifier (GUID) indicates the documented reference that created this well known string [ID-TRAF-CLASS], the APP is the profile name with no spaces, and the "VER=" is included, but has no value, but MAY if versioning becomes important.

### 3.3 The Multimedia Conferencing Profiles

There will be Multimedia Conferencing profiles for presentation data, application sharing and whiteboarding, where these applications will most often be associated with a larger Conversational (audio and/or audio/video) conference. Timing is important, but some minimal delays are acceptable, unlike the case for Realtime-Interactive traffic.

- Multimedia-Conferencing presentation-data
- Multimedia-Conferencing application-sharing
- Multimedia-Conferencing whiteboarding

Here is the profile for identifying Multimedia-Conferencing Application-sharing

```
APP_TC, POLICY_LOCATOR, ASCII_DN,  
"GUID=http://www.ietf.org/internet-drafts/  
    draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=multimedia-conferencing.application-sharing, VER="
```

Where the Globally Unique Identifier (GUID) indicates the RFC reference that created this well known string [ID-TRAF-CLASS], the APP is the profile name with no spaces, and the "VER=" is included, but has no value, but MAY if versioning becomes important.

### 3.4 The Multimedia Streaming Profiles

Multimedia Streaming profiles are for more significantly buffered one-way streaming flows than Broadcast profiles. These include...

There will be Multimedia Streaming profiles for

- Multimedia-Streaming multiplex
- Multimedia-Streaming webcast

Here is the profile for identifying Multimedia Streaming webcast

```
APP_TC, POLICY_LOCATOR, ASCII_DN,  
"GUID=http://www.ietf.org/internet-drafts/  
    draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=multimedia-streaming.webcast, VER="
```

Where the Globally Unique Identifier (GUID) indicates the documented reference that created this well known string [ID-TRAF-CLASS], the APP is the profile name with no spaces, and the "VER=" is included, but has no value, but MAY if versioning becomes important.

### 3.5 The Conversational Profiles

Conversational category is for realtime bidirectional communications, such as voice or video, and is the most numerous due to the choices of application with or without adjectives. The number of profiles is then doubled because there needs to be one for unadmitted and one for admitted. The IANA section lists all that are currently proposed for registration at this time, therefore there will not be an exhaustive list provided in this section.

There will be conversational profiles for

- Conversational Audio
- Conversational Audio Admitted
- Conversational Video
- Conversational Video Admitted
- Conversational Audio Avconf
- Conversational Audio Avconf Admitted
- Conversational Video Avconf
- Conversational Video Avconf Admitted
- Conversational Audio Immersive
- Conversational Audio Immersive Admitted
- Conversational Video Immersive
- Conversational Video Immersive Admitted

Here is an example profile for identifying Conversational Audio:

```
APP_TC, POLICY_LOCATOR, ASCII_DN,  
"GUID=http://www.ietf.org/internet-drafts/  
    draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.audio, VER="
```

Where the Globally Unique Identifier (GUID) indicates the documented reference that created this well known string [ID-TRAF-CLASS], the APP is the profile name with no spaces, and the "VER=" is included, but has no value, but MAY if versioning becomes important.

#### 4. Security considerations

The security considerations section within RFC 2872 sufficiently covers this document, with one possible exception - someone using the wrong template values (e.g., claiming a reservation is Multimedia Streaming when it is in fact Real-time Interactive). Given that each traffic flow is within separate reservations, and RSVP does not have the ability to police the type of traffic within any reservation, solving for this appears to be administratively handled at best. This is not meant to be a 'punt', but there really is nothing this template creates that is going to make things any harder for anyone (that we know of now).

#### 5. IANA considerations

##### 5.1 New RSVP Policy Element (P-Type)

In line with the convention created in RFC 3182, the following P-Type is created in the RSVP Policy Element registry [TBD]:

4	APP_TC	Traffic Class identification of applications
---	--------	--

[Editor's note: Unfortunately, RFC 2750 specified the creation of the "RSVP Policy Element" IANA registration, which does not appear at the <http://www.iana.org/assignments/rsvp-parameters> page, therefore it appears this registry does not yet exist. We will get with the chairs to work on this.]

##### 5.2 Application Profiles

This document requests IANA create a new registry for the application identification classes similar to the following table within the Resource Reservation Protocol (RSVP) Parameters registry:

Registry Name: RSVP APP-ID Profiles  
Reference: [this document]  
Registration procedures: Standards Track document [RFC5226]

## 5.2.1 Broadcast Profiles IANA Registry

## Broadcast Audio IPTV Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=broadcast.audio.iptv, VER="

Reference: [this document]

## Broadcast Video IPTV Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=broadcast.video.iptv, VER="

Reference: [this document]

## Broadcast Audio Live-events Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=broadcast.audio.live-events, VER="

Reference: [this document]

## Broadcast Audio Live-events Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=broadcast.video.live-events, VER="

Reference: [this document]

## Broadcast Audio-Surveillance Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=broadcast.audio.surveillance, VER="

Reference: [this document]

#### Broadcast Video-Surveillance Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=broadcast.video.surveillance, VER="

Reference: [this document]

### 5.2.2 Realtime-Interactive Profiles IANA Registry

#### Realtime-Interactive Gaming Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP= realtime-interactive.gaming, VER="

Reference: [this document]

#### Real-time Interactive Remote-Desktop Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=realtime-interactive.remote-desktop, VER="

Reference: [this document]

#### Real-time Interactive Virtualized-Desktop Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=realtime-interactive.virtualized-desktop,  
VER="

Reference: [this document]

### 5.2.3 Multimedia-Conferencing Profiles IANA Registry

#### Multimedia-Conferencing Presentation-Data Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.ietf.org/internet-drafts/  
    draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
    APP= multimedia-conferencing.presentation-data,  
    VER="

Reference: [this document]

#### Multimedia-Conferencing Application-Sharing Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.ietf.org/internet-drafts/  
    draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
    APP= multimedia-conferencing.application-sharing,  
    VER="

Reference: [this document]

#### Multimedia-Conferencing Whiteboarding Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.ietf.org/internet-drafts/  
    draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
    APP= multimedia-conferencing.whiteboarding, VER="

Reference: [this document]

### 5.2.4 Multimedia-Streaming Profiles IANA Registry

#### Multimedia-Streaming Multiplex Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.ietf.org/internet-drafts/  
    draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
    APP=multimedia-streaming.multiplex, VER="

Reference: [this document]

#### Multimedia-Streaming Webcast Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.ietf.org/internet-drafts/  
    draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
    APP=multimedia-streaming.webcast, VER="

Reference: [this document]

## 5.2.5 Conversational Profiles IANA Registry

## Conversational Audio Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.audio, VER="

Reference: [this document]

## Conversational Audio Admitted Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.audio.aq:admitted, VER="

Reference: [this document]

## Conversational Video Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.video, VER="

Reference: [this document]

## Conversational Video Admitted Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.video.aq:admitted, VER="

Reference: [this document]

## Conversational Audio Avconf Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.audio.avconf, VER="

Reference: [this document]

## Conversational Audio Avconf Admitted Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.audio.avconf.aq:admitted,  
VER="

Reference: [this document]

## Conversational Video Avconf Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.video.avconf, VER="

Reference: [this document]

## Conversational Video Avconf Admitted Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.video.avconf.aq:admitted,  
VER="

Reference: [this document]

## Conversational Audio Immersive Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.audio.immersive, VER="

Reference: [this document]

## Conversational Audio Immersive Admitted Profile

P-type = APP\_TC  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.audio.immersive.aq:admitted,  
VER="



Reference: [this document]

Conversational Video Immersive Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.video.immersive, VER="

Reference: [this document]

Conversational Video Immersive Admitted Profile

P-type = APP\_TC

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.ietf.org/internet-drafts/  
draft-ietf-mmusic-traffic-class-for-sdp-01.txt,  
APP=conversational.video.immersive.ag:admitted,  
VER="

Reference: [this document]

## 7. Acknowledgments

To Francois Le Faucheur, Paul Jones and Glen Lavers for their helpful comments and encouragement.

## 8. References

### 8.1. Normative References

- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997
- [RFC2205] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997
- [RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers ", RFC 2474, December 1998
- [RFC2750] S. Herzog, "RSVP Extensions for Policy Control", RFC 2750, January 2000
- [RFC2872] Y. Bernet, R. Pabbati, "Application and Sub Application Identity Policy Element for Use with RSVP", RFC 2872, June 2000

- [RFC2996] Y. Bernet, "Format of the RSVP DCLASS Object", RFC 2996, November 2000
- [RFC3182] S. Yadav, R. Yavatkar, R. Pabbati, P. Ford, T. Moore, S. Herzog, R. Hess, "Identity Representation for RSVP", RFC 3182, October 2001
- [RFC5226] T. Narten, H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, May 2008
- [ID-TRAF-CLASS] J. Polk, S. Dhesikan, P. Jones, " The Session Description Protocol (SDP) 'trafficclass' Attribute ", work in progress, Oct 2011

## 8.2. Informative References

- [RFC4594] J. Babiarz, K. Chan, F Baker, "Configuration Guidelines for Diffserv Service Classes", RFC 4594, August 2006

## Authors' Addresses

James Polk  
3913 Treemont Circle  
Colleyville, Texas, USA  
+1.817.271.3552

mailto: jmpolk@cisco.com

Subha Dhesikan  
170 W Tasman St  
San Jose, CA, USA  
+1.408-902-3351

mailto: sdhesika@cisco.com

## Appendix - Changes to ID

### A.1 - Changes from Individual -03 to -04

The following changes were made in this version:

- clarified security considerations section to mean RSVP cannot police the type of traffic within a reservation to know if a traffic flow should be using a different profile, as defined in this document.
- changed existing informative language regarding "... other Sub-types ..." from 'can' to normative 'MAY'.

- editorial changes to clear up minor mistakes

#### A.2 - Changes from Individual -02 to -03

The following changes were made in this version:

- Added [ID-TRAF-CLASS] as a reference
- Changed to a new format of the profile string.
- Added many new profiles based on the new format into each parent category of Section 3.
- changed the GUID to refer to draft-ietf-mmusic-traffic-class-for-sdp-01.txt
- changed 'desktop' adjective to 'avconf' to keep in alignment with draft-ietf-mmusic-traffic-class-for-sdp-01.txt
- Have a complete IANA Registry proposal for each application-ID discussed in this draft.
- General text clean-up of the draft.

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: July 19, 2014

R. Stewart  
Adara Networks  
M. Tuexen  
Muenster Univ. of Appl. Sciences  
January 15, 2014

Quick Start Plus  
draft-stewart-tsvwg-qsp-03.txt

Abstract

This document describes an extension to Quick Start including the missing Stream Control Transmission Protocol (SCTP) QuickStart chunk types and procedures so that SCTP may also use the QuickStart extension.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 19, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions . . . . .	2
3. Terminology . . . . .	3
4. Data Format . . . . .	3
4.1. Quick Start Extended IP option . . . . .	3
4.2. Quick Start Echo (133) for SCTP . . . . .	4
5. Procedures . . . . .	5
5.1. Quick Start Added Procedures . . . . .	5
5.2. SCTP QSE Receiver Procedures . . . . .	6
6. Security Considerations . . . . .	7
7. IANA Considerations . . . . .	7
8. Acknowledgements . . . . .	7
9. References . . . . .	7
9.1. Normative references . . . . .	7
9.2. Informational References . . . . .	7
Authors' Addresses . . . . .	7

## 1. Introduction

QuickStart [RFC4782] was introduced as an experimental RFC in 2007. This document attempts to address several issues in QuickStart including granularity of request, active router participation in quick start besides setup, and the lack of QuickStart for SCTP [RFC4960]. In order to address these issues this document specifies:

- \* An extended format for QuickStart that allows a more finegrained rate to be specified by the router.
- \* A new code QuickStart Function Code to allow a router to initiate a rate change to a transport endpoint.
- \* Chunk format's and handling procedures for SCTP's use of QuickStart.

Note that with few exceptions this document does not change the general procedures laid out in [RFC4782] readers unfamiliar with that document are encouraged to read it before reading this document.

## 2. Conventions

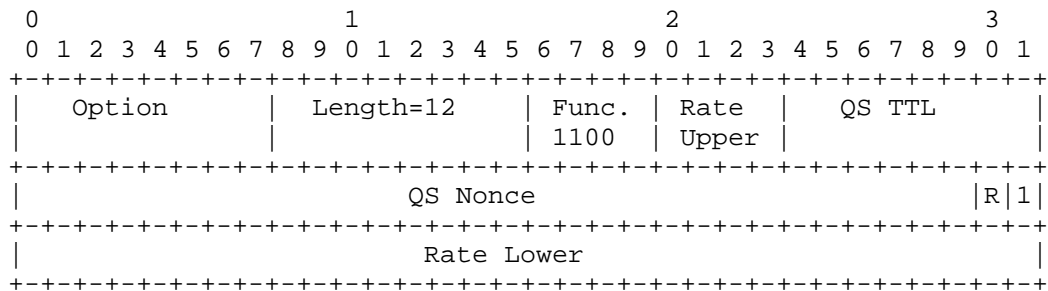
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

### 3. Terminology

All integer fields defined in this document **MUST** be transmitted in network byte order, unless otherwise stated.

### 4. Data Format

#### 4.1. Quick Start Extended IP option



Option: 8 bits

As defined in [RFC4782].

Length: 8 bits

Contains the value 12 to indicate the new extended option. If the field contains the value 8, then the shorter classic version of the rate is used per [RFC4782].

Function: 4 bits

Contains the values 0000 or 1000 as defined in [RFC4782] or the new value 1100 which indicates that a router is adjusting the rate and should be treated the same as a request 0000 (i.e. echoed by the destination endpoint in an appropriate transport option back to the sender of the packet).

Rate: 4 bits

In the extended form, these four bits are the upper bits of the rate being requested (or set). The combined rate field yeilds a 36 bit integer indicating the number of kilobits per second that is being requested or authorized.

QS TTL: 8 bits

As defined in [RFC4782].

QS Nonce: 30 bits

As defined in [RFC4782].

Reserved: 1 bits

This bit is reserved and SHOULD be set to 0 on transmit and ignored upon reception.

E bit: 1 bits

This bit is set to 1 to indicate that the extended form of the option is present. This can also be verified by confirming that the length field is set to 12.

Rate Lower: 32 bits

This field holds the lower 32 bits of the rate. This field is combined with the 4 bit Rate Upper field to yield a 36 bit rate in kilo-bits per second.

#### 4.2. Quick Start Echo (133) for SCTP

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Chunk Type=133| Flags=00000000|   Chunk Length = 12 or 16   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   QuickStart Option as copied from IP options field   |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Chunk Flags: 8 bits

Set to all zeros on transmit and ignored on receipt.

This parameter contains the exact copy of the IP options field received on the incoming packet that caused this chunk to be sent. The chunk is normally added to the next outgoing packet (often times a SACK or DATA chunk).

## 5. Procedures

### 5.1. Quick Start Added Procedures

Procedures for the extended form of Quickstart are identical to those defined in [RFC4782]. The only two differences are as follows.

First when presented with the extended version of QuickStart the receiver of the IP option must combine the two Rate fields defined to extract the rate information being requested (or authorized). The handling of the rate by a router is still done as previously, i.e. examine the rate, decrement the TTL and possibly lower the rate (these procedures do not change and remain the same). For an endpoint receiving a QuickStart option in the extended form, the transport option is also increased by four bytes and the entire option is copied and sent to the peer endpoint if the arriving Function Code is either 0000 or the new 1100.

Secondly both endpoints and routers may receive the new function code 1100. When such a function code arrives, it should be treated the same as a function code 0000 in classic QuickStart i.e. it is treated as a rate request. If a router is receiving this code, it again looks to see if it needs to lower the rate, decrements the TTL and possibly updates the rate and nonce fields. Transport endpoints receiving this Function code echo the complete 12 byte option within their transport specific manner to the peer transport endpoint (i.e. the sender of the arriving packet that contains the Quickstart option).

The sender of the 1100 function code, however is not a transport endpoint requesting a rate. Instead this option may be inserted by any router on the path to adjust the rate of a flow passing through it. The router MUST have flow state and MUST have previously seen a QuickStart Rate Request (Func 0000) and a subsequent Quickstart Rate Report (Func 1000) from that flow before inserting this option. In other words the router MUST know that the flow supports QuickStart and that all downstream routers also support QuickStart. Furthermore the router must know if the extended form of QuickStart is in use by the flow or as an alternative use the non-extended format. This is determined based on if the initial QuickStart Rate Request which was received is in the extended format (aka 12 bytes with the E bit set) or in the old original format (aka 8 bytes with two reserved bits set to zero).

When a Router inserts the 1100 function code, the router MUST use the last saved QS Nonce and TTL's that were seen when the flow originally sent its QuickStart Rate Request. The router MUST adjust the TTL so that the difference between the original packet's Rate Request's TTL is



present in the packet being inserted. The router should also adjust the proper field in the nonce per [RFC4782] to reflect which rate range the router was adjusting the flow to. This allows the endpoint to validate both the Nonce and the TTL in the same manner in which it would if it had sent the IP option as a rate request.

## 5.2. SCTP QSE Receiver Procedures

Procedures for SCTP are identical with those associated with TCP and can be found within [RFC4782]. The key difference is the method in which the Response Option is carried in SCTP. SCTP does not use options, instead a new chunk is defined that carries either the traditional QuickStart option, or the extended form.

At SCTP association startup the two endpoint exchange an INIT and INIT-ACK chunk as defined in [RFC4960]. During this exchange both endpoints MUST include a Supported Extensions chunk as defined in [RFC5061]. Both endpoints MUST indicate support for the new QSE chunk type. If both endpoints do not indicate this then the use of QuickStart is not enabled for this SCTP association. If both endpoints do indicate the support of QuickStart, then the next packet out (usually the Cookie-Echo or Cookie-Ack) SHOULD include an IP option requesting a rate. Note the endpoint SHOULD use the extended format. The fields are initialized the same as defined in [RFC4782] with the exception that if the extended format is used, the rate is an exact 36 bit value representing the requested rate.

Upon receiving a packet with an IP option indicating either Function 0000 or Function 1100, the SCTP endpoint MUST include in the next outgoing packet (most likely a SACK or DATA chunk) the QSE chunk with a copy of the IP option that arrived.

When receiving a QSE chunk, an endpoint follows the procedures in [RFC4960] validating the TTL difference and the nonce before setting the local cwnd to the new rate. Any validation failure, for example if there is a TTL difference indicating that not all routers participated in the QuickStart exchange, then the endpoint MUST NOT use the QuickStart procedures to change the cwnd.

Note that because IP options are not always passed correctly by routers and middle boxes, an endpoint SHOULD be prepared to disable the use of QuickStart if the initial transmissions of the IP option is not acknowledged (e.g. the endpoint sends a COOKIE-ECHO with the QS IP option and the retransmit timer fires due to the lack of a COOKIE-ACK response from the endpoint).

## 6. Security Considerations

[RFC4782] defines the security considerations for Quickstart. These same consideration that are described for TCP are applicable to this document.

## 7. IANA Considerations

Nothing requested.

## 8. Acknowledgements

## 9. References

### 9.1. Normative references

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, January 2007.
- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.
- [RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", RFC 5061, September 2007.

### 9.2. Informational References

- [RFC2481] Ramakrishnan, K. and S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP", RFC 2481, January 1999.
- [RFC2960] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

## Authors' Addresses

Randall R. Stewart  
Adara Networks  
Chapin, SC 29036  
USA

Email: randall@lakerest.net

Michael Tuexen  
Muenster University of Applied Sciences  
Stegerwaldstr. 39  
48565 Steinfurt  
Germany

Email: tuexen@fh-muenster.de

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: July 19, 2014

R. Stewart  
Adara Networks  
M. Tuexen  
Muenster Univ. of Appl. Sciences  
X. Dong  
Huawei  
January 15, 2014

ECN for Stream Control Transmission Protocol (SCTP)  
draft-stewart-tsvwg-sctpecn-05.txt

Abstract

This document describes the addition of the ECN to the Stream Control Transmission Protocol (SCTP).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 19, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions . . . . .	3
3. Terminology . . . . .	3
4. Chunk and Parameter Formats . . . . .	3
4.1. ECN Support Parameter (32768) . . . . .	3
4.2. ECN Echo (12) . . . . .	3
4.3. CWR Chunk(13) . . . . .	4
5. Procedures . . . . .	5
5.1. SCTP Initialization . . . . .	5
5.2. The SCTP Sender . . . . .	6
5.3. The SCTP Receiver . . . . .	8
5.4. Congestion on the SACK path . . . . .	9
5.5. Retransmitted SCTP Packets . . . . .	9
5.6. SCTP Window Probes . . . . .	10
6. Security Considerations . . . . .	10
7. IANA Considerations . . . . .	10
8. Acknowledgements . . . . .	10
9. References . . . . .	10
9.1. Normative references . . . . .	10
9.2. Informational References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

At the time SCTP was initially defined in [RFC2960] ECN - [RFC2481] was still an experimental document. This left the authors of SCTP in a position where they could not directly refer to ECN without creating a normative reference in a standards track document to an experimental RFC. To work around this problem the authors of SCTP decided to add two reserved chunk types for ECN (CWR and ECNE) but did not fully specify how they were to be used except in a vague way within an appendix of the document. This worked around the document reference problem, but left ECN and its implementation for SCTP unspecified. This document is intended to fill in the details of ECN processing in SCTP in a standards track document.

This document assumes that the reader is familiar with ECN [RFC3168]. Readers unfamiliar with ECN are strongly encouraged to first read [RFC3168] since this document will not repeat any of the details on how the various IP level bits are set. This document will use the same terminology as [RFC3168]. For example the term ECT is used to indicate that the IP level packet is marked indicating the transport (SCTP) supports ECN.

## 2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. Terminology

All integer fields defined in this document included in an SCTP packet MUST be transmitted in network byte order, unless otherwise stated.

ECT - The term used to indicate that the IP level packet is marked indicating the transport is willing to support ECN for this packet.

not-ECT - The term used to indicate that the IP level packet is marked indicating the transport is NOT willing to support ECN for this packet.

CE - The term used to indicate that the IP level packet is marked indicating that a router in the network has marked the packet as having experienced congestion

## 4. Chunk and Parameter Formats

### 4.1. ECN Support Parameter (32768)

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Parameter Type = 32768 | Parameter Length = 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

This parameter is used to indicate the support for ECN. If this parameter is present, the sender of the chunk is indicating that it supports ECN and wishes to use ECN for the newly forming association.

#### Valid Chunk Appearance

The ECN Supported Parameter may appear in the INIT, or the INIT-ACK chunk type.

### 4.2. ECN Echo (12)

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Chunk Type=12 | Flags=00000000 |   Chunk Length = 12   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Lowest TSN Number   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Number CE Marked Packets Seen since CWR
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Chunk Flags: 8 bits

Set to all zeros on transmit and ignored on receipt.

Lowest TSN Number: 32 bits (unsigned integer)

This parameter contains the lowest TSN number contained in the last packet received that was marked by the network with a CE indication.

Number CE Marked Packets: 32 bits (unsigned integer)

This parameter contains the total number of CE marked packets that has been seen since the first CE mark received while waiting for a CWR chunk. Note that the CE counter will overflow from 0xffffffff to 0 if a CWR chunk is not recieved.

Note that the appendix of [RFC4960] did not have the field Number CE Marked Packets. Implementations SHOULD accept an 8 byte form of this chunk that does not include this field. In such a case the implementation SHOULD treat the missing field as indicating one CE marked packet for any purpose for which the implementation is using this field.

#### 4.3. CWR Chunk(13)

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Chunk Type=13 | Flags=0000000R |   Chunk Length = 8   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     TSN Number         |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Chunk Flags: 8 bits

The R Bit indicates if the CWR is a retransmission of an earlier CWR that may have been lost. If this bit is set, then the TSN number included is the latest TSN that a CWR has been responded to. If the o bit is clear, then the TSN indicated is the latest TSN for that destination.

Set to all zeros on transmit and ignored on receipt.

TSN Number: 32 bits (unsigned integer)

This parameter contains the TSN number to which the sender has reduced his congestion window to.

## 5. Procedures

### 5.1. SCTP Initialization

In the SCTP association setup phase, the source and destination SCTP endpoints exchange information about their willingness to use ECN. After the completion of this negotiation, an SCTP sender sets an ECT codepoint in the IP header of data packets to indicate to the network that the transport is capable and willing to participate in ECN for this packet. This indicates to the routers that they may mark this packet with the CE codepoint.

If the SCTP association does not wish to use ECN notification for a particular packet, the sending SCTP sets the ECN codepoint to not-ECT, and the SCTP receiver ignores the CE codepoint in the received packet.

For this discussion we will call the endpoint initiating the SCTP association as EP-A and the listening SCTP endpoint as EP-Z.

Before an SCTP association can use ECN, EP-A sends an INIT chunk which includes the ECN Support parameter. By including the ECN Support parameter the sending endpoint (EP-A) will participate in ECN as both a sender and a receiver. Specifically, as a receiver, it will respond to incoming data packets that have the CE codepoint set in the IP header by sending an ECN Echo chunk bundled with the next outgoing SACK Chunk. As a sender, it will respond to incoming packets that include an ECN Echo chunk by reducing the congestion window and sending a CWR chunk when appropriate.

Including an ECN Support parameter in an INIT or INIT-ACK does not commit the SCTP sender to setting the ECT codepoint in any or all of



the packets it may transmit. However, the commitment to respond appropriately to incoming packets with the CE codepoint set remains.

When EP-Z sends INIT-ACK chunk, it also includes an ECN Support parameter. Including the ECN Support parameter indicates that the SCTP transmitting the INIT-ACK chunk is ECN-Capable.

The following rules apply to the use of ECN for an SCTP association.

- \* If the SCTP Endpoint supports ECN a sender of either an INIT or INIT-ACK chunk MUST ALWAYS include the ECN Supported Parameter.
- \* After the exchange of the INIT and INIT-ACK if both endpoints have NOT indicated support of ECN by including an ECN Supported Parameter, then ECT MUST NOT be set on any IP packets sent by any endpoint which is ECN capable. Furthermore upon receiving IP packets with a CE codepoint set, the ECN capable endpoint SHOULD ignore the CE codepoint.
- \* If both endpoints have included an ECN Supported Parameter in the INIT and INIT-ACK exchange, then both endpoints MUST follow the ECN procedures defined in the rest of this document.
- \* A sending endpoint SHOULD set the ECT code points on IP packets that carry Data chunk. This includes IP packets that have other control chunks bundled with the Data.

## 5.2. The SCTP Sender

For an SCTP association using ECN, new data packets are transmitted with an ECT codepoint set in the IP header. When only one ECT codepoint is needed by a sender for all packets sent on an SCTP association ECT(0) SHOULD be used. If the sender receives an ECN-Echo chunk packet, then the sender knows that congestion was encountered in the network on the path from the sender to the receiver. The indication of congestion should be treated just as a congestion loss in non-ECN-Capable SCTP. That is, the SCTP source halves the congestion window "cwnd" for the destination address that the sender transmitted the data to and reduces the slow start threshold "ssthresh". A packet containing an ECN-Echo chunk shouldn't trigger new data to be sent. SCTP follows the normal procedures for increasing the congestion window when it receives a packet with a SACK chunk without the ECN Echo chunk.

SCTP should not react to congestion indications more than once every round-trip time. That is, the SCTP sender's congestion window should be reduced only once in response to a series of dropped and/or CE packets from a single window of data. In addition, the SCTP source

should not decrease the slow-start threshold, `ssthresh`, if it has been decreased within the last round trip time.

One method to accomplish this is as following:

- 1) During association setup, create a new state variable `ECN_ECHO_TSN` and `ECN_ECHO_LAST` for each destination. The initial value of these variables are set to the initial TSN that will be assigned minus 1.
- 2) When an ECN Echo chunk arrives, use the TSN in the ECN Echo to establish which destination the packet was sent to. We will call this destination the selected destination. If the chunk cannot be found note that an override is occurring. From the selected destination (if found) select its ECN Echo TSN.
- 3) Compare the ECN Echo TSN with the `ECN_ECHO_TSN` for the selected destination. If an override is not noted and the value of the `ECN_ECHO_TSN` is greater than the ECN Echo TSN proceed to step 4; else proceed to step 6b.
- 4) Reduce the `cwnd` and `ssthresh` for the selected destination the same as if a loss was detected during a fast retransmit. For details, see [RFC4960] Section 7.2.3 and Section 7.2.4.
- 5) Record in the `ECN_ECHO_TSN` value, the last TSN that was sent and recorded in `ECN_ECHO_LAST` the TSN number from the ECN Echo Chunk.
- 6a) If the implementation is tracking the number of marked packets, record the value found in the 'Number CE Marked Packets Seen since CWR' field and also add this number to the running loss count. If such a count is not being maintained, then proceed to step 7.
- 6b) If the implementation is tracking the number of marked packets, compare the number in the ECN Echo Chunk TSN to the `ECN_ECHO_LAST`. If it is greater than `ECN_ECHO_LAST`, update `ECN_ECHO_LAST` with this value. Take the difference between the stored 'Number CE Marked Packets' field and the value from the newly arriving 'Number CE Marked Packets' and add this difference to the total loss count. Then update the stored 'Number CE Marked Packets' with the ECN Echo Chunk TSN.
- 7) Create a CWR chunk with the value found in the `ECN_ECHO_LAST` for the selected destination. If an override was noted, set the 'O' bit within the CWR flags. Queue this chunk for transmission to the peer destination. Note if there is already such a chunk in queue to be sent, remove that chunk and replace it with the new chunk.

After the sending SCTP reduces its congestion window in response to a ECN Echo, incoming SACKs that continue to arrive can "clock out" outgoing packets as allowed by the reduced congestion window. Note that continued arrival of ECN Echo chunks should still be processed as described above, possibly reducing the cwnd, but always sending a CWR to the receiving SCTP. This assures that the ECN Echo and CWR are robust with regard to loss in either direction and that the implementation, if it desires, can maintain an accurate loss count per destination.

Note, originally in the appendix of [RFC4960] a definition was supplied for the ECN Echo chunk. This definition did NOT include the 'Number CE Marked Packets' field. An implementation SHOULD accept such a chunk, delineating it from the standards track version by the fact that the length field will be 8 bytes instead of 12. When processing this older style chunk, the 'Number CE Marked Packets' should be treated as if it contains the number 1. This may cause incorrect loss counts but will NOT cause any issues with SCTP's ECN handling.

### 5.3. The SCTP Receiver

When an SCTP endpoint first receives a CE data packet at the destination end-system, the SCTP data receiver creates an ECN Echo chunk and records the lowest TSN number found in the data packet. It also sets the 'Number CE Marked Packets' to 1 and queues this chunk for transmission at the next opportunity. If there is any ACK withholding implemented, as in current "delayed-SACK" SCTP implementations where the SCTP receiver can send an SACK for two arriving data packets, then the ECN Echo chunk will not be sent until the SACK is sent. If the next arriving data packet also has the CE codepoint set, then the receiver updates the queued ECN Echo chunk to have a higher TSN value (the lowest one in the newly arriving data packet) and increments the 'Number CE Marked Packets' field in the queued chunk.

Multi-homing requires one added restriction upon the ECN Echo chunk, such a chunk MUST be bundled with a SACK, and the SACK MUST follow the ECN Echo Chunk. This ordering is necessary so that the receiver of the ECN Echo chunk will at least one time find the proper destination to which the chunk was originally sent. Without this restriction it is possible a SACK could arrive ahead of the ECN Echo Chunk, no matter what the sending order, causing the sender to free the DATA chunk and thus lose the association with what destination it was sent to. For the same reason we also require the ECN Echo Chunk be earlier in the packet ahead of the SACK so that the SACK is not processed before the ECN Echo Chunk.

After transmission of the ECN Echo chunk, usually bundled with the SACK, the receiver does NOT discard the ECN Echo chunk. Instead it keeps the chunk in its queue and continues to send this chunk bundled with at least a SACK chunk on each outgoing packet, updating it as described above if other CE codepoint data packets arrive. The ECN Echo chunk should only be discarded when a CWR Chunk arrives holding a TSN value that is greater than or equal to the value inside the ECN Echo Chunk.

This provides robustness against the possibility of a dropped SACK packet carrying an ECN Echo chunk. The SCTP receiver continues to transmit the ECN Echo chunk in subsequent SACK packets until the correct CWR is received.

After the receipt of the CWR chunk, acknowledgments for subsequent non-CE data packets will not have an ECN Echo chunk bundled with them. If another CE packet is received by the data receiver, the receiver would once again send SACK packets bundled with a newly created ECN Echo chunk. The receipt of a CWR packet guarantees that the data sender has received the ECN Echo chunk for the TSN specified, and reduced its congestion window at some point *after* it sent the data packet for which the CE codepoint was set.

When processing a CWR, it is important that the receiver of the CWR validate the source address from which the CWR came from. It SHOULD match the destination the ECN Echo was sent to unless the override bit is set in the CWR Chunk.

#### 5.4. Congestion on the SACK path

For the current generation of SCTP congestion control algorithms, pure acknowledgement packets (e.g., packets that do not contain any accompanying data) MUST be sent with the not-ECT codepoint. Current SCTP receivers have no mechanisms for reducing traffic on the SACK-path in response to congestion notification. Mechanisms for responding to congestion on the SACK-path are areas for current and future research. For current SCTP implementations, a single dropped SACK generally has only a very small effect on SCTP's sending rate.

#### 5.5. Retransmitted SCTP Packets

This document specifies ECN-capable SCTP implementations MUST NOT set either ECT codepoint (ECT(0) or ECT(1)) in the IP header for retransmitted data packets, and that the SCTP data receiver SHOULD ignore the ECN field on arriving data packets that are outside of the receiver's current window. The reasons for this can be found in [RFC3168] Section 6.1.5.

## 5.6. SCTP Window Probes

When the SCTP data receiver advertises a zero window, the SCTP data sender sends window probes to determine if the receiver's window has increased. Window probe packets for SCTP do contain user data (one chunk). If a window probe packet is dropped in the network, this loss can be detected by the receiver. Therefore, the SCTP data sender MAY set an ECT codepoint on the initial send of the window probe, but the SCTP sender MUST NOT set the ECT codepoint on retransmissions of that TSN.

## 6. Security Considerations

[RFC3168] defines the security considerations for ECN. These same consideration that are described for TCP are applicable to SCTP.

## 7. IANA Considerations

TBD

## 8. Acknowledgements

Thanks to Richard Scheffenegger for his helpful comments and review.

## 9. References

### 9.1. Normative references

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

### 9.2. Informational References

- [RFC2481] Ramakrishnan, K. and S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP", RFC 2481, January 1999.
- [RFC2960] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

Authors' Addresses

Randall R. Stewart  
Adara Networks  
Chapin, SC 29036  
USA

Email: [randall@lakerest.net](mailto:randall@lakerest.net)

Michael Tuexen  
Muenster University of Applied Sciences  
Stegerwaldstr. 39  
48565 Steinfurt  
Germany

Email: [tuexen@fh-muenster.de](mailto:tuexen@fh-muenster.de)

Xuesong Dong  
Huawei  
Pleasanton, CA 94566  
USA

Email: [stevedong@huawei.com](mailto:stevedong@huawei.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 22, 2013

S. Shah  
P. Thubert  
Cisco Systems  
October 19, 2012

Differentiated Service Class Recommendations for LLN Traffic  
draft-svshah-lln-diffserv-recommendations-01

Abstract

Differentiated services architecture is widely deployed in traditional networks. There exist well defined recommendations for the use of appropriate differentiated service classes for different types of traffic (eg. audio, video) in these networks. Per-Hop Behaviors are typically defined based on this recommendations. With emerging Low-power and Lossy Networks (LLNs), it is important to have similar defined differentiated services recommendations for LLN traffic. Defined recommendations are for LLN class of traffic exiting out of LLNs towards high-speed backbones, converged campus network and for the traffic in the reverse direction.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 22, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

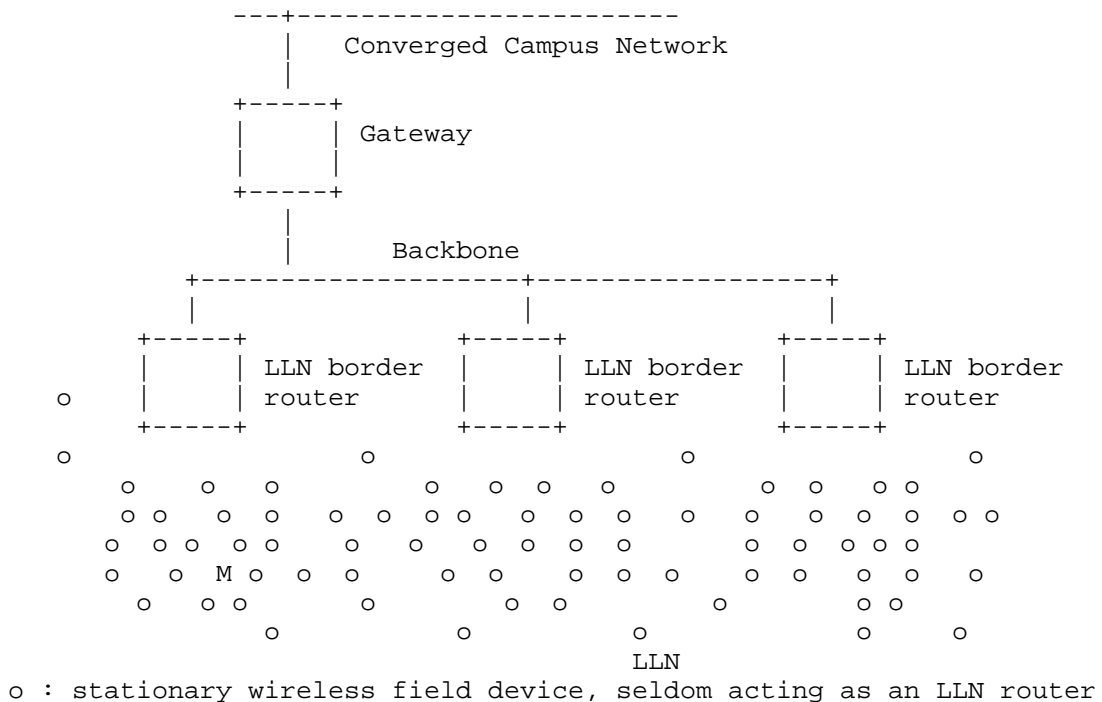


## Table of Contents

1. Introduction . . . . .	4
1.1. Definitions . . . . .	5
2. Terminology . . . . .	5
3. Application Types and Traffic Patterns . . . . .	6
3.1. Alert signals . . . . .	6
3.2. Control signals . . . . .	7
3.3. Monitoring data . . . . .	7
3.3.1. Video data . . . . .	8
3.3.2. Query based data . . . . .	8
3.3.3. Periodic reporting/logging, Software downloads . . . . .	8
3.4. Traffic Class Characteristics Table . . . . .	9
4. Differentiated Service recommendations for LLN traffic . . . . .	9
4.1. Alert signals . . . . .	9
4.2. Control signals . . . . .	10
4.3. Monitoring Data . . . . .	10
4.3.1. Video Data . . . . .	10
4.3.2. Query based data . . . . .	10
4.3.3. Periodic reporting/logging, Software downloads . . . . .	10
4.4. Summary of Differentiated Code-points and QoS Mechanics for them . . . . .	11
5. Deployment Scenario . . . . .	11
6. Security Considerations . . . . .	13
7. Acknowledgements . . . . .	13
8. References . . . . .	13
8.1. Normative References . . . . .	13
8.2. Informative References . . . . .	14
Authors' Addresses . . . . .	14

## 1. Introduction

With emerging LLN applications, it is anticipated that more and more LLNs will be federated by high-speed backbones, possibly supporting deterministic Ethernet service, and be further connected to some converged campus networks for less demanding usages such as supervisory control like traffic originated in a LLN, such as metering, command and control, may transit over a converged campus IP network.



In an example figure shown above, Per-Hop Behaviors (PHB) and Service Level Agreements (SLAs), for LLN traffic, require to be defined at the LLN Borders as well as Backbone and possibly in the Converged Campus network.

In this document, we will first categorize different types of LLN traffic into service classes and then provide recommendations for Differentiated Service Code-Point(DSCP) for those service classes. Mechanisms to be used, like Traffic Conditioning and Active Queue Management, for differentiated services is well defined in RFC4594.

This document does not focus to re-call them again here but the document will call out any specific mechanism that requires particular consideration.

This document focuses on Diffserv recommendations for LLN class of traffic in managed IP networks outside of LLNs, that is for the traffic from LLN towards LLN Border, Backbone, Campus Network as well as for the traffic in the reverse direction. It does not focus on Diffserv architecture or any other QoS recommendations within the LLNs itself. Given constraints of LLNs and their unique requirements, it is expected of a focus within a separate efforts. Though nodes inside LLNs MAY use code-points recommended here.

In Section 3 we categorize different types of traffic from Different LLNs. Section 4 recommends differentiated services, including DSCPs and QoS mechanics, for categorized classes of traffic. Section 5 evaluates one of the deployment scenario.

### 1.1. Definitions

DSCP: Differentiated Service Code Point. It is a 6-bits value in the TOS and Traffic Class field of the IPv4 and IPv6 header respectively. This 6-bits numerical value defines standard set of behavior to be performed by Differentiated Services capable hops.

Diffserv

Class: Diffserv Class in this document is used to refer to DSCP code-point(s) and associated Per-hop Behaviors for it.

LLN: Low-power and lossy Network. Network constructed with sensors, actuators, routers that are low-power and with higher loss/success transmission ratio, due to transmission medium and nature of dynamics of changing topology, compare to wired and other traditional networks.

SLA: Service Level Agreement. It is a collection of Traffic classification rules and set of services associated with each Traffic Class. Traffic classification may be defined based on just DSCP code-points or additionally (or otherwise) based on some other packet attributes.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this

document are to be interpreted as described in RFC2119.

### 3. Application Types and Traffic Patterns

Different types of traffic can be collapsed into following network service classes.

- Alert signals
- Critical control signals
- Monitoring data
  - Video data
  - Query-based response data
  - Periodic reporting/logging, Software downloads

#### 3.1. Alert signals

Alerts/alarms reporting fall in this category where such signal is triggered in a rare un-usual circumstances. An alert may be triggered for an example when environmental hazard level goes beyond certain threshold. Such alerts require to be reported with in the human tolerable time. Note that certain critical alert reporting in certain automation systems may be reported via very closely and tightly managed method that is not implemented within LLNs, due to the nature of transmission media of LLNs and due to the stringent latency requirement for those alerts. Such types of signals are not considered here since they are not within the scope of LLN or any other IP networks.

Examples :

- Environmental hazard level goes beyond certain threshold
- Measured blood pressure exceeds the threshold or a person falls to the ground
- Instructional triggers like start/stop traffic lights during certain critical event

Traffic Pattern:

Typically size of such packets is very small. any specific device of LLN is expected to trigger only handful of packets (may be only 1 packet). That too only during an event which is not a common occurrence.

In an affected vicinity, only a designated device or each affected devices may send alerts. In certain type of sensor networks, it is predictable and expected to have only a designated device to trigger such an alert while in certain other types scale number of alert flows may be expected.

Latency required for such traffic is not stringent but is to be within human tolerable time bound.

### 3.2. Control signals

Besides alerts, LLNs also trigger and/or receive different types of control signals, to/from control applications outside of LLNs. These control signals are important enough for the operation of sensors, actuators and underneath network. Administrator controlling applications, outside of LLN network, may trigger a control signal in response to alerts/data received from LLN (in some cases control signal trigger may be automated without explicit human interaction in the loop) or administrator may trigger an explicit control signal for a specific function.

#### Examples:

- auto [demand] response (e.g. manage peak load, service disconnect, start/stop street lights)
- manual remote service disconnect, remote demand reset
- open-loop regulatory control
- non-critical close-loop regulatory control
- trigger to start Video surveillance

#### Traffic Pattern:

Variable size packets but typically size of such packets is small. Certain control signals may be regular and so with number of devices in a particular LLN, it is predictable on average, how many such signals to expect. However, certain other control signals are irregular or on-demand.

Typically most of the open-loop, that requires manual interaction, signals are tolerant to latency above 1 second. Certain close-loop control signals require low jitter and low latency, latency in the order of 100s of ms.

Some of the tightly coupled closed loop control signals are very sensitive to latency and jitter. However they today, just like critical alerts, are implemented via other management methods outside of LLN.

### 3.3. Monitoring data

Reaction to control signals may initiate flow of data-traffic in either direction. Sensors/Actuators in LLNs may also trigger periodic data (eg. monitoring, reporting data). All different types

of data may be categorized in following classes.

### 3.3.1. Video data

A very common example of this type of monitoring data is Video surveillance or Video feed, triggered thru control signals. This Video feed is typically from LLN towards an application outside of LLN.

Traffic Pattern:

Video frame size is expected to be big with a flow of variable rate.

### 3.3.2. Query based data

Application at the controller, outside the LLN, or user explicitly may launch query for the data. For example, query for an urban environmental data, query for health report etc. Since this data is query based data, it is important to report data with reasonable latency though not stringent. In addition, some periodic logging data also may require timely reporting and so may expect same type of service (eg. at-home health reporting).

Traffic Pattern:

Size of packets can vary from small to big. While rate may be predictable in some cases, in most of the cases traffic rate for such data is variable. The traffic is bursty in the nature.

### 3.3.3. Periodic reporting/logging, Software downloads

Many sensors/actuator in different LLNs report data periodically. With some exceptions, as mentioned above for healthcare monitoring logs, most of such data do not have any latency requirement and can be forwarded either thru lower priority assured forwarding or with service of store and forward or even best effort.

Sensors/actuators may require software/firmware upgrades where software/ firmware may be downloaded on demand bases. These upgrades and so downloads do not have stringent requirement of timely delivery to the accuracy of seconds. This data also can be forwarded thru lower priority assured forwarding.

Traffic Pattern:

Periodic reporting/logging typically can be predicated as constant rate. Data may be bursty in the nature. Software download data also may be bursty in nature. Such traffic is tolerant to jitter and

latency.

### 3.4. Traffic Class Characteristics Table

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
Alerts/ alarms	Packet size = small Rate = typically 1-few packets Short lived flow Burst = not bursty	Low	Low	N/A
Control Signals	Packet size = variable, typically small Rate = few packets Short lived flow Burst = none to some-what	Yes	Low	Yes
Very low latency close-loop Control Signals	Packet size = variable, typically small Rate = few packets Short lived flow Burst = none to some-what	Low	Very Low	Low
Video Monitoring/feed	Packet size = big Rate = variable Long lived flow Burst = non-bursty	Low	Low - Medium	Low
Query-based Data	Packet size = variable Rate = variable Short lived elastic flow Burst = bursty	Low	Medium	Yes
Periodic Reporting/log, Software downloads	variable packet size, rate bursty	Yes	Medium - High	Yes

## 4. Differentiated Service recommendations for LLN traffic

### 4.1. Alert signals

Alerts/alarms signaling service requires transmission of few packets with low delay, tolerable to human. This requirement is very similar

to signaling traffic in the traditional networks. Alert signals MAY use Diffserv code-point CS5.

#### 4.2. Control signals

As described in earlier section, control signals over IP are divided in two categories. Control signals that require very low latency, service inline with EF PHB, and control signals that require low delay but do not mandate lowest latency. Service requirement for later class of control signals is very similar to service for signaling traffic in the traditional networks. Recommendation for this class of control signals is to use Diffserv code-point CS5.

Control signals, like some of the closed-loop signals, that require lower delay and jitter compare to CS5 class of control signals, are recommended to use EF Diffserv class. These control signals are expected to be of the small packet size and short-lived flows. Specifically while sharing EF class with voice traffic, any big control packets can cause additional latency to voice packets and so care MUST be taken either to use a different Diffserv class for them or compress such packets to smaller size.

#### 4.3. Monitoring Data

##### 4.3.1. Video Data

RFC4594 has well documented recommendations for different types of Video traffic. If there is any Video traffic from/to LLNs to/from outside of LLNs, they should use same recommended dscp from RFC4594. For example, surveillance video feed is recommended to use dscp CS3.

##### 4.3.2. Query based data

Low latency data, like query based report and non-critical signals, is recommended to use AF2 assured forwarding service. Also, certain periodic reporting/logging data that are critical to be reported with regular interval with relatively low jitter is recommended to use AF2x service.

##### 4.3.3. Periodic reporting/logging, Software downloads

Non-critical periodic reporting/logging and rest all other data MAY use AF1x or BE service class.



## 4.4. Summary of Differentiated Code-points and QoS Mechanics for them

- Alert Signals CS5
- Control Signaling CS5
- Lower latency control-signals EF
- Video broadcast/feed CS3
- Query-based data AF2x
- Assured monitoring data AF1x  
high throughput
- Best Effort monitoring data BE  
Reporting (periodic reporting.certain types of periodic monitoring  
MAY require assured forwarding)

Service Class	DSCP	Conditioning at DS Edge	PHB Used	Queuing	AQM
lower latency control signals	EF	Police using sr+bs Police using sr+bs	RFC3246	Priority	No
Alert signals/ Control signals	CS5	Police using sr+bs	RFC2474	Rate	No
Video feed	CS3	Police using sr+bs	RFC2474	Rate	No
Query-based Data	AF21 AF22 AF23	Using single-rate, three-color marker (such as RFC 2697)	RFC2597	Rate	Yes per DSCP
Periodic Reporting/ logging	AF11 AF12 AF13	Using two-rate, three-color marker (such as RFC 2698)	RFC2597	Rate	Yes per DSCP

\* "sr+bs" represents a policing mechanism that provides single rate with burst size control [RFC4594]

## 5. Deployment Scenario

Industrial Automation, as described in [RFC5673] and [ISA100.11a], classifies different types of traffic in following six classes ranging in complexity from Class 5 to Class 0 where Class 0 is the most time sensitive class.

- o Safety

- \* Class 0: Emergency action - Always a critical function

- o Control

- \* Class 1: Closed-loop regulatory control - Often a critical function
  - \* Class 2: Closed-loop supervisory control - Usually a non-critical function
  - \* Class 3: Open-loop control - Operator takes action and controls the actuator (human in the loop)

- o Monitoring

- \* Class 4: Alerting - Short-term operational effect (for example, event-based maintenance)
  - \* Class 5: Logging and downloading / uploading - No immediate operational consequence (e.g., history collection, sequence-of-events, preventive maintenance)

It might not be appropriate to transport Class 0 traffic over a wireless network or a multihop network, unless tight mechanisms are put in place such as TDM and frequency hopping. Today this class of traffic is expected to use other tightly managed method outside of IP networks. Excluding class 0 traffic, following table maps Class 1 thru Class 5 service classes to Diffserv code-point.

Service Class	DSCP
Class 1*	EF
Class 2	CS5
Class 3	CS5
Class 4	AF2x
class 5	AF1x/BE

\* Any Class 1 traffic that requires very tight control over latency and jitter falls in the same category as Class 0 traffic.

## 6. Security Considerations

A typical trust model, as much is applicable in traditional networks, is applicable with LLN traffic as well. At the border of the LLN, a trust model needs to be established for any traffic coming out of LLN. Without appropriate trust model to accept/mark dscp code-point for LLN traffic, misbehaving flow may attack a specific Diffserv class disrupting expected service for other traffic from the same Diffserv class. Trust models are typically established at the border router by employing rate-limiting and even marking down dscp code-point to Best Effort for non-trusted flows or dropping them as required.

## 7. Acknowledgements

Thanks to Fred Baker, James Polk for their valuable comments and suggestions.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, August 2006.
- [RFC5127] Chan, K., Babiarz, J., and F. Baker, "Aggregation of Diffserv Service Classes", RFC 5127, February 2008.
- [RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.
- [RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.
- [RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, April 2010.
- [RFC5867] Martocci, J., De Mil, P., Riou, N., and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and

Lossy Networks", RFC 5867, June 2010.

## 8.2. Informative References

- [ISA100.11a]  
ISA, "ISA-100.11a-2011 - Wireless systems for industrial automation: Process control and related applications", May 2011.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.
- [RFC6272] Baker, F. and D. Meyer, "Internet Protocols for the Smart Grid", RFC 6272, June 2011.

## Authors' Addresses

Shitanshu Shah  
Cisco Systems  
170 W. Tasman Drive  
San Jose, CA 95134  
US

Email: [svshah@cisco.com](mailto:svshah@cisco.com)

Pascal Thubert  
Cisco Systems  
Village d'Entreprises Green Side  
400, Avenue de Roumanille  
Batiment T3  
Biot - Sophia Antipolis 06410  
FRANCE

Email: [pthubert@cisco.com](mailto:pthubert@cisco.com)



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 17, 2013

R. Jesup  
WorldGate Communications  
S. Loreto  
Ericsson  
R. Stewart  
Adara Networks  
M. Tuexen  
Muenster Univ. of Appl. Sciences  
July 16, 2012

DTLS Encapsulation of SCTP Packets for RTCWEB  
draft-tuexen-tsvwg-sctp-dtls-encaps-01.txt

Abstract

The Stream Control Transmission Protocol (SCTP) is a transport protocol originally defined to run on top of the network protocols IPv4 or IPv6. This memo document specifies how SCTP can be used on top of the Datagram Transport Layer Security (DTLS) protocol. SCTP over DTLS is used by the RTCWeb protocol suite for transporting non-media data between browsers.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Conventions . . . . .	3
3. Encapsulation and Decapsulation Procedure . . . . .	3
4. DTLS Considerations . . . . .	4
5. SCTP Considerations . . . . .	4
6. IANA Considerations . . . . .	6
7. Security Considerations . . . . .	6
8. Acknowledgments . . . . .	6
9. References . . . . .	6
9.1. Normative References . . . . .	6
9.2. Informative References . . . . .	7
Authors' Addresses . . . . .	7

## 1. Introduction

### 1.1. Overview

The Stream Control Transmission Protocol (SCTP) as defined in [RFC4960] is a transport protocol running on top of the network protocols IPv4 or IPv6. This memo document specifies how SCTP can be used on top of the Datagram Transport Layer Security (DTLS) protocol. SCTP over DTLS is used by the RTCWeb protocol suite (see [I-D.ietf-rtcweb-overview] for an overview) for transporting non-media data between browsers. The architecture of this stack is described in [I-D.jesup-rtcweb-data].

### 1.2. Terminology

This document uses the following terms:

Association: An SCTP association.

Stream: A unidirectional stream of an SCTP association. It is uniquely identified by a stream identifier.

### 1.3. Abbreviations

DTLS: Datagram Transport Layer Security.

MTU: Maximum Transmission Unit.

PPID: Payload Protocol Identifier.

SCTP: Stream Control Transmission Protocol.

TCP: Transmission Control Protocol.

TLS: Transport Layer Security.

## 2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. Encapsulation and Decapsulation Procedure

When an SCTP packet is sent down to the DTLS layer, the complete SCTP packet, consisting of the SCTP common header and a number of SCTP



chunks, MUST be handled as the payload of the application layer protocol of DTLS. When the DTLS layer has processed a DTLS record containing a message of the application layer protocol, the payload MUST be given up to the SCTP layer. The SCTP layer expects an SCTP common header followed by a number of SCTP chunks.

#### 4. DTLS Considerations

The DTLS implementation MUST be based on [RFC6347].

If path MTU discovery is performed by the DTLS layer, the method described in [RFC4821] MUST be used. For probe packets, the extension defined in [RFC6520] MUST be used.

If path MTU discovery is performed by the SCTP layer and IPv4 is used as the network layer protocol, the DTLS implementation MUST allow the DTLS user to enforce that the corresponding IPv4 packet is sent with the DF bit set.

SCTP performs segmentation and reassembly based on the path MTU. Therefore the DTLS layer MUST NOT use any compression algorithm.

#### 5. SCTP Considerations

##### 5.1. Base Protocol

SCTP as specified in [RFC4960] is used. However, the following restrictions are necessary to reflect that the lower layer is the connection oriented protocol DTLS instead of the connection less protocol IPv4 and IPv6:

- o A DTLS connection MUST be established before an SCTP association can be set up.
- o All associations MUST be single-homed.
- o The INIT and INIT-ACK chunk MUST NOT contain any IPv4 Address or IPv6 Address parameters. The INIT chunk MUST NOT contain the Supported Address Types parameter.
- o The implementation MUST NOT rely on processing ICMP or ICMPv6 packets. This applies in particular to path MTU discovery when performed by SCTP.
- o The DTLS implementation might not allow the setting of ECN bits for outgoing packets or provide the ECN bits for incoming packets.

In this case, SCTP MUST NOT use ECN.

- o The DTLS implementation might not allow the setting of DF bit for outgoing packets. In this case, SCTP can't perform path MTU discovery.

## 5.2. Padding Extension

The padding extension defined in [RFC4820] MUST be supported and used for probe packets when performing path MTU discovery as specified in [RFC4821].

## 5.3. Dynamic Address Reconfiguration Extension

The SCTP implementation MUST support the Supported Extensions Parameter defined in [RFC5061] to signal the support of the SCTP stream reset extension (see Section 5.6). The other functionality described in [RFC5061] MUST NOT be used.

## 5.4. SCTP Authentication Extension

The SCTP authentication extension defined in [RFC4895] is not required.

## 5.5. Partial Reliability Extension

The SCTP implementation MUST support the extension defined in [RFC3758].

The SCTP implementation SHOULD support the following PR-SCTP policies:

- o A user message is abandoned after a user specified lifetime.
- o A user message is abandoned if the number of retransmissions exceeds a user specified threshold.

## 5.6. Stream Reset Extension

The SCTP implementation MUST support the SCTP stream reset extension defined in [RFC6525]. It is used to reset streams and add streams during the lifetime of the SCTP association.

## 5.7. Large User Message Extension

SCTP as defined in [RFC4960] does not support the multiplexing of large user messages that need to be fragmented and reassembled by the SCTP layer. To overcome this limitation, the SCTP implementation

SHOULD support an extension, which has to be defined.

#### 5.8. Congestion Control

In addition to the TCP-like congestion control specified in [RFC4960], other congestion control algorithms MAY be provided. For example, it might be helpful to use a congestion control which does not increase the queueing delay substantially (see [I-D.ietf-ledbat-congestion] for an example).

#### 6. IANA Considerations

This document requires no actions from IANA.

#### 7. Security Considerations

TBD.

#### 8. Acknowledgments

The authors wish to thank XXX for their invaluable comments.

#### 9. References

##### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, May 2004.
- [RFC4820] Tuexen, M., Stewart, R., and P. Lei, "Padding Chunk and Parameter for the Stream Control Transmission Protocol (SCTP)", RFC 4820, March 2007.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.
- [RFC4895] Tuexen, M., Stewart, R., Lei, P., and E. Rescorla, "Authenticated Chunks for the Stream Control Transmission Protocol (SCTP)", RFC 4895, August 2007.

- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.
- [RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", RFC 5061, September 2007.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.
- [RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, February 2012.
- [RFC6525] Stewart, R., Tuexen, M., and P. Lei, "Stream Control Transmission Protocol (SCTP) Stream Reconfiguration", RFC 6525, February 2012.

## 9.2. Informative References

- [I-D.ietf-rtcweb-overview]  
Alvestrand, H., "Overview: Real Time Protocols for Brower-based Applications", draft-ietf-rtcweb-overview-04 (work in progress), June 2012.
- [I-D.jesup-rtcweb-data]  
Jesup, R., Loreto, S., and M. Tuexen, "RTCWeb Datagram Connection", draft-jesup-rtcweb-data-01 (work in progress), October 2011.
- [I-D.ietf-ledbat-congestion]  
Hazel, G., Iyengar, J., Kuehlewind, M., and S. Shalunov, "Low Extra Delay Background Transport (LEDBAT)", draft-ietf-ledbat-congestion-09 (work in progress), October 2011.

Authors' Addresses

Randell Jesup  
WorldGate Communications  
3800 Horizon Blvd, Suite #103  
Trevose, PA 19053-4947  
US

Phone: +1-215-354-5166  
Email: randell\_ietf@jesup.org

Salvatore Loreto  
Ericsson  
Hirsalantie 11  
Jorvas 02420  
FI

Email: Salvatore.Loreto@ericsson.com

Randall R. Stewart  
Adara Networks  
Chapin, SC 29036  
US

Email: randall@lakerest.net

Michael Tuexen  
Muenster University of Applied Sciences  
Stegerwaldstrasse 39  
48565 Steinfurt  
DE

Email: tuexen@fh-muenster.de



Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: 13 August 2022

P. D. Amer  
University of Delaware  
M. Becke  
HAW Hamburg  
T. Dreibholz  
SimulaMet  
N. Ekiz  
University of Delaware  
J. Iyengar  
Franklin and Marshall College  
P. Natarajan  
Cisco Systems  
R. R. Stewart  
Netflix  
M. Tuexen  
Muenster Univ. of Appl. Sciences  
9 February 2022

Load Sharing for the Stream Control Transmission Protocol (SCTP)  
draft-tuexen-tsvwg-sctp-multipath-23

Abstract

The Stream Control Transmission Protocol (SCTP) supports multi-homing for providing network fault tolerance. However, mainly one path is used for data transmission. Only timer-based retransmissions are carried over other paths as well.

This document describes how multiple paths can be used simultaneously for transmitting user messages.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 13 August 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Conventions . . . . .	3
3. Load Sharing . . . . .	3
3.1. Split Fast Retransmissions . . . . .	4
3.2. Appropriate Congestion Window Growth . . . . .	4
3.3. Appropriate Delayed Acknowledgements . . . . .	5
4. Non-Renegable SACK . . . . .	6
4.1. Negotiation . . . . .	6
4.2. The New Chunk Type: Non-Renegable SACK (NR-SACK) . . . . .	6
4.3. An Illustrative Example . . . . .	11
4.4. Procedures . . . . .	15
4.4.1. Sending an NR-SACK chunk . . . . .	15
4.4.2. Receiving an NR-SACK Chunk . . . . .	17
5. Buffer Blocking Mitigation . . . . .	18
5.1. Sender Buffer Splitting . . . . .	18
5.2. Receiver Buffer Splitting . . . . .	18
5.3. Chunk Rescheduling . . . . .	18
5.4. Problems during Path Failure . . . . .	18
5.4.1. Problem Description . . . . .	18
5.4.2. Solution: Potentially-failed Destination State . . . . .	19
5.5. Non-Renegable SACK . . . . .	19
5.5.1. Problem Description . . . . .	19
5.5.2. Solution: Non-Renegable SACKs . . . . .	20
6. Handling of Shared Bottlenecks . . . . .	20
6.1. Introduction . . . . .	20
6.2. Initial Values . . . . .	20
6.3. Congestion Window Growth . . . . .	21
6.4. Congestion Window Decrease . . . . .	21
7. Chunk Scheduling and Rescheduling . . . . .	21
8. Socket API Considerations . . . . .	21
9. Testbed Platforms . . . . .	21
10. IANA Considerations . . . . .	21



10.1. A New Chunk Type . . . . .	22
11. Security Considerations . . . . .	22
12. Acknowledgments . . . . .	22
13. References . . . . .	22
13.1. Normative References . . . . .	22
13.2. Informative References . . . . .	23
Authors' Addresses . . . . .	27

## 1. Introduction

One of the important features of the Stream Control Transmission Protocol (SCTP), which is currently specified in [RFC4960], is network fault tolerance. This feature is for example required for Reliable Server Pooling (RSerPool, [RFC5351]). Therefore, transmitting messages over multiple paths is supported, but only for redundancy. So [RFC4960] does not specify how to use multiple paths simultaneously.

This document overcomes this limitation by specifying how multiple paths can be used simultaneously. This has several benefits:

- \* Improved bandwidth usage.
- \* Better availability check with real user messages compared to HEARTBEAT-based information.

## 2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. Load Sharing

Basic requirement for applying SCTP load sharing is the Concurrent Multipath Transfer (CMT) extension of SCTP, which utilises multiple paths simultaneously. We denote CMT-enabled SCTP as CMT-SCTP throughout this document. CMT-SCTP is introduced in [IAS06] and in more detail in [I06], some illustrative examples of chunk handling are provided in [OMNeTWorkshop2010-SCTP]. CMT-SCTP provides three modifications to standard SCTP (split Fast Retransmissions, appropriate congestion window growth and delayed SACKs), which are described in the following subsections.

### 3.1. Split Fast Retransmissions

Paths with different latencies lead to overtaking of DATA chunks. This leads to gap reports, which are handled by Fast Retransmissions. However, due to the fact that multiple paths are used simultaneously, these Fast Retransmissions are usually useless and furthermore lead to a decreased congestion window size.

To avoid unnecessary Fast Retransmissions, the sender has to keep track of the path each DATA chunk has been sent on and consider transmission paths before performing Fast Retransmissions. That is, on reception of a SACK, the sender MUST identify the highest acknowledged TSN on each path. A chunk SHOULD only be considered as missing if its TSN is smaller than the highest acknowledged TSN on its path. Section 3.1 of [OMNeTWorkshop2010-SCTP] contains an illustrated example.

### 3.2. Appropriate Congestion Window Growth

The congestion window adaptation algorithm for SCTP [RFC4960] allows increasing the congestion window only when a new cumulative ack (CumAck) is received by a sender. When SACKs with unchanged CumAcks are generated (due to reordering) and later arrive at a sender, the sender does not modify its congestion window. Since a CMT-SCTP receiver naturally observes reordering, many SACKs are sent containing new gap reports but not new CumAcks. When these gaps are later acked by a new CumAck, congestion window growth occurs, but only for the data newly acked in the most recent SACK. Data previously acked through gap reports will not contribute to congestion window growth, in order to prevent sudden increases in the congestion window resulting in bursts of data being sent.

To overcome the problems described above, the congestion window growth has to be handled as follows [IAS06]:

- \* The sender SHOULD keep track of the earliest non-retransmitted outstanding TSN per path.
- \* The sender SHOULD keep track of the earliest retransmitted outstanding TSN per path.
- \* The in-order delivery per path SHOULD be deduced.
- \* The congestion window of a path SHOULD be increased when the earliest non-retransmitted outstanding TSN of this path is advanced ('Pseudo CumAck') OR when the earliest retransmitted outstanding TSN of this path is advanced ('RTX Pseudo CumAck').

Section 3.2 of [OMNeTWorkshop2010-SCTP] contains an illustrated example of appropriate congestion window handling for CMT-SCTP.

### 3.3. Appropriate Delayed Acknowledgements

Standard SCTP [RFC4960] sends a SACK as soon as an out-of-sequence TSN has been received. Delayed Acknowledgements are only allowed if the received TSNs are in sequence. However, due to the load balancing of CMT-SCTP, DATA chunks may overtake each other. This leads to a high number of out-of-sequence TSNs, which have to be acknowledged immediately. Clearly, this behaviour increases the overhead traffic (usually nearly one SACK chunk for each received packet containing a DATA chunk).

Delayed Acknowledgements for CMT-SCTP are handled as follows:

- \* In addition to [RFC4960], delaying of SACKs SHOULD \*also\* be applied for out-of-sequence TSNs.
- \* A receiver MUST maintain a counter for the number of DATA chunks received before sending a SACK. The value of the counter is stored into each SACK chunk (FIXME: add details; needs reservation of flags bits by IANA). After transmitting a SACK, the counter MUST be reset to 0. Its initial value MUST be 0.
- \* The SACK handling procedure for a missing TSN M is extended as follows:
  - If all newly acknowledged TSNs have been transmitted over the same path:
    - o If there are newly acknowledged TSNs L and H so that  $L \leq M \leq H$ , the missing count of TSN M SHOULD be incremented by one (like for standard SCTP according to [RFC4960]).
    - o Else if all newly acknowledged TSNs N satisfy the condition  $M \leq N$ , the missing count of TSN M SHOULD be incremented by the number of TSNs reported in the SACK chunk.
  - Otherwise (that is, there are newly acknowledged TSNs on different paths), the missing count of TSN M SHOULD be incremented by one (like for standard SCTP according to [RFC4960]).

Section 3.3 of [OMNeTWorkshop2010-SCTP] contains an illustrated example of Delayed Acknowledgements for CMT-SCTP.

## 4. Non-Renegable SACK

### 4.1. Negotiation

Before sending/receiving NR-SACKs (see [YEN10]), both peer endpoints MUST agree on using NR-SACKs. This agreement MUST be negotiated during association establishment. NR-SACK is an extension to the core SCTP, and SCTP extensions that an endpoint supports are reported to the peer endpoint in Supported Extensions Parameter during association establishment (see Section 4.2.7 of [RFC5061].) The Supported Extensions Parameter consists of a list of non-standard Chunk Types that are supported by the sender.

An endpoint supporting the NR-SACK extension MUST list the NR-SACK chunk in the Supported Extensions Parameter carried in the INIT or INIT-ACK chunk, depending on whether the endpoint initiates or responds to the initiation of the association. If the NR-SACK chunk type ID is listed in the Chunk Types List of the Supported Extensions Parameter, then the receiving endpoint MUST assume that the NR-SACK chunk is supported by the sending endpoint.

Both endpoints MUST support NR-SACKs for either endpoint to send an NR-SACK. If an endpoint establishes an association with a remote endpoint that does not list NR-SACK in the Supported Extensions Parameter carried in INIT chunk, then both endpoints of the association MUST NOT use NR-SACKs. After association establishment, an endpoint MUST NOT renegotiate the use of NR-SACKs.

Once both endpoints indicate during association establishment that they support the NR-SACK extension, each endpoint SHOULD acknowledge received DATA chunks with NR-SACK chunks, and not SACK chunks. That is, throughout an SCTP association, both endpoints SHOULD send either SACK chunks or NR-SACK chunks, never a mixture of the two.

### 4.2. The New Chunk Type: Non-Renegable SACK (NR-SACK)

Table 1 illustrates a new chunk type that will be used to transfer NR-SACK information.

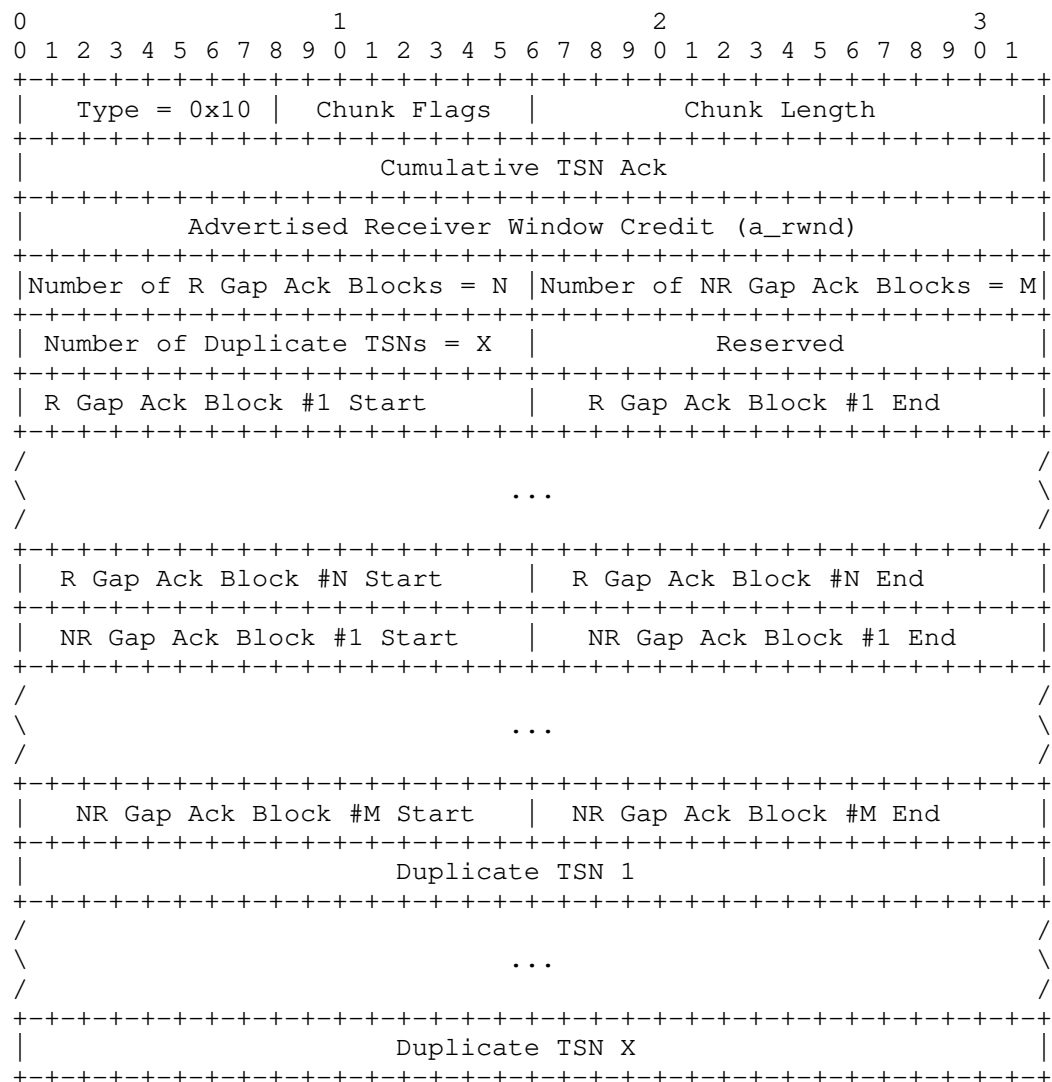
Chunk Type	Chunk Name
0x10	Non-Renegable Selective Acknowledgment (NR-SACK)

Table 1: NR-SACK Chunk

As the NR-SACK chunk replaces the SACK chunk, many SACK chunk fields are preserved in the NR-SACK chunk. These preserved fields have the same semantics with the corresponding SACK chunk fields, as defined

in [RFC4960], Section 3.3.4. The Gap Ack fields from RFC4960 have been renamed as R Gap Ack to emphasize their renegable nature. Their semantics are unchanged. For completeness, we describe all fields of the NR-SACK chunk, including those that are identical in the SACK chunk.

Similar to the SACK chunk, the NR-SACK chunk is sent to a peer endpoint to (1) acknowledge DATA chunks received in-order, (2) acknowledge DATA chunks received out-of-order, and (3) identify DATA chunks received more than once (i.e., duplicate.) In addition, the NR-SACK chunk (4) informs the peer endpoint of non-renegable out-of-order DATA chunks.



Type: 8 bits

This field holds the IANA defined chunk type for NR-SACK chunk. The suggested value of this field for IANA is 0x10.

Chunk Flags: 8 bits

Currently not used. It is recommended a sender set all bits to zero on transmit, and a receiver ignore this field.

Chunk Length: 16 bits (unsigned integer) [Same as SACK chunk]

This value represents the size of the chunk in bytes including the Chunk Type, Chunk Flags, Chunk Length, and Chunk Value fields.

Cumulative TSN Ack: 32 bits (unsigned integer) [Same as SACK chunk]

The value of the Cumulative TSN Ack is the last TSN received before a break in the sequence of received TSNs occurs. The next TSN value following the Cumulative TSN Ack has not yet been received at the endpoint sending the NR-SACK.

Advertised Receiver Window Credit (a\_rwnd): 32 bits (unsigned integer) [Same as SACK chunk]

Indicates the updated receive buffer space in bytes of the sender of this NR-SACK, see Section 6.2.1 of [RFC4960] for details.

Number of (R)enegable Gap Ack Blocks (N): 16 bits (unsigned integer)

Indicates the number of Renegable Gap Ack Blocks included in this NR-SACK.

Number of (N)on(R)enegable Gap Ack Blocks (M): 16 bits (unsigned integer)

Indicates the number of Non-Renegable Gap Ack Blocks included in this NR-SACK.

Number of Duplicate TSNs (X): 16 bits [Same as SACK chunk]

Contains the number of duplicate TSNs the endpoint has received. Each duplicate TSN is listed following the NR Gap Ack Block list.

Reserved : 16 bits

Currently not used. It is recommended a sender set all bits to zero on transmit, and a receiver ignore this field.

(R)enegable Gap Ack Blocks:

The NR-SACK contains zero or more R Gap Ack Blocks. Each R Gap Ack Block acknowledges a subsequence of renegable out-of-order TSNs. By definition, all TSNs acknowledged by R Gap Ack Blocks are "greater than" the value of the Cumulative TSN Ack.

Because of TSN numbering wraparound, comparisons and all arithmetic

operations discussed in this document are based on "Serial Number Arithmetic" as described in Section 1.6 of [RFC4960].

R Gap Ack Blocks are repeated for each R Gap Ack Block up to 'N' defined in the Number of R Gap Ack Blocks field. All DATA chunks with TSNs  $\geq$  (Cumulative TSN Ack + R Gap Ack Block Start) and  $\leq$  (Cumulative TSN Ack + R Gap Ack Block End) of each R Gap Ack Block are assumed to have been received correctly, and are renegable.

R Gap Ack Block Start: 16 bits (unsigned integer)

Indicates the Start offset TSN for this R Gap Ack Block. This number is set relative to the cumulative TSN number defined in Cumulative TSN Ack field. To calculate the actual start TSN number, the Cumulative TSN Ack is added to this offset number. The calculated TSN identifies the first TSN in this R Gap Ack Block that has been received.

R Gap Ack Block End: 16 bits (unsigned integer)

Indicates the End offset TSN for this R Gap Ack Block. This number is set relative to the cumulative TSN number defined in the Cumulative TSN Ack field. To calculate the actual TSN number, the Cumulative TSN Ack is added to this offset number. The calculated TSN identifies the TSN of the last DATA chunk received in this R Gap Ack Block.

N(on)R(enegable) Gap Ack Blocks:

The NR-SACK contains zero or more NR Gap Ack Blocks. Each NR Gap Ack Block acknowledges a continuous subsequence of non-renegable out-of-order DATA chunks. If a TSN is nr-gap-acked in any NR-SACK chunk, then all subsequently transmitted NR-SACKs with a smaller cum-ack value than that TSN SHOULD also nr-gap-ack that TSN.

NR Gap Ack Blocks are repeated for each NR Gap Ack Block up to 'M' defined in the Number of NR Gap Ack Blocks field. All DATA chunks with TSNs  $\geq$  (Cumulative TSN Ack + NR Gap Ack Block Start) and  $\leq$  (Cumulative TSN Ack + NR Gap Ack Block End) of each NR Gap Ack Block are assumed to be received correctly, and are Non-Renegable.

NR Gap Ack Block Start: 16 bits (unsigned integer)

Indicates the Start offset TSN for this NR Gap Ack Block. This number is set relative to the cumulative TSN number defined in Cumulative TSN Ack field. To calculate the actual TSN number, the Cumulative TSN Ack is added to this offset number. The calculated



TSN identifies the first TSN in this NR Gap Ack Block that has been received.

NR Gap Ack Block End: 16 bits (unsigned integer)

Indicates the End offset TSN for this NR Gap Ack Block. This number is set relative to the cumulative TSN number defined in Cumulative TSN Ack field. To calculate the actual TSN number, the Cumulative TSN Ack is added to this offset number. The calculated TSN identifies the TSN of the last DATA chunk received in this NR Gap Ack Block.

Note:

NR Gap Ack Blocks and R Gap Ack Blocks in an NR-SACK chunk SHOULD acknowledge disjoint sets of TSNs. That is, an out-of-order TSN SHOULD be listed in either an R Gap Ack Block or an NR Gap Ack Block, but not the both. R Gap Ack Blocks and NR Gap Ack Blocks together provide the information as do the Gap Ack Block of a SACK chunk, plus additional information about non-renegability.

If all out-of-order data acked by an NR-SACK are renegable, then the Number of NR Gap Ack Blocks MUST be set to 0. If all out-of-order data acked by an NR-SACK are non-renegable, then the Number of R Gap Ack Blocks SHOULD be set to 0. TSNs listed in R Gap Ack Block will be referred as r-gap-acked.

Duplicate TSN: 32 bits (unsigned integer) [Same as SACK chunk]

Indicates a duplicate TSN received since the last NR-SACK was sent. Exactly 'X' duplicate TSNs SHOULD be reported where 'X' was defined in Number of Duplicate TSNs field.

Each duplicate TSN is listed in this field as many times as the TSN was received since the previous NR-SACK was sent. For example, if a data receiver were to get the TSN 19 three times, the data receiver would list 19 twice in the outbound NR-SACK. After sending the NR-SACK if the receiver received one more TSN 19, the receiver would list 19 as a duplicate once in the next outgoing NR-SACK.

#### 4.3. An Illustrative Example

Assume the following DATA chunks have arrived at the receiver.

	TSN=16		SID=2		SSN=N/A		U=1	
	TSN=15		SID=1		SSN= 4		U=0	
	TSN=14		SID=0		SSN= 4		U=0	
	TSN=13		SID=2		SSN=N/A		U=1	
	TSN=11		SID=0		SSN= 3		U=0	
	TSN=8		SID=2		SSN=N/A		U=1	
	TSN=7		SID=1		SSN= 2		U=0	
	TSN=6		SID=1		SSN= 1		U=0	
	TSN=5		SID=0		SSN= 1		U=0	
	TSN=3		SID=1		SSN= 0		U=0	
	TSN=2		SID=0		SSN= 0		U=0	

The above figure shows the list of DATA chunks at the receiver. TSN denotes the transmission sequence number of the DATA chunk, SID denotes the stream id to which the DATA chunk belongs, SSN denotes the sequence number of the DATA chunk within its stream, and the U bit denotes whether the DATA chunk requires ordered(=0) or unordered(=1) delivery [RFC4960]. Note that TSNs 4,9,10, and 12 have not arrived.

This data can be viewed as three separate streams as follows (assume each stream begins with SSN=0.) Note that in this example, the application uses stream 2 for unordered data transfer. By definition, SSN fields of unordered DATA chunks are ignored.

Stream-0:

SSN:	0	1	2	3	4
TSN:	2	5		11	14
U-Bit:	0	0		0	0

Stream-1:

SSN:	0	1	2	3	4
TSN:	3	6	7		15
U-Bit:	0	0	0		0

Stream-2:

SSN:	N/A	N/A	N/A
TSN:	8	13	16
U-Bit:	1	1	1

The NR-SACK to acknowledge the above data SHOULD be constructed as follows for each of the three cases described below (the a\_rwnd is arbitrarily set to 4000):

CASE-1: Minimal Data Receiver Responsibility - no out-of-order deliverable data yet delivered

None of the deliverable out-of-order DATA chunks have been delivered, and the receiver of the above data does not take responsibility for any of the received out-of-order DATA chunks. The receiver reserves the right to renege any or all of the out-of-order DATA chunks.

Type = 0x10	00000000	Chunk Length = 32
Cumulative TSN Ack = 3		
a_rwnd = 4000		
Num of R Gap Ack Blocks = 3	Num of NR Gap Ack Blocks = 0	
Num of Duplicates = 0	0x00	
R Gap Ack Block #1 Start = 2	R Gap Ack Block #1 End = 5	
R Gap Ack Block #2 Start = 8	R Gap Ack Block #2 End = 8	
R Gap Ack Block #3 Start = 10	R Gap Ack Block #3 End = 13	

CASE-2: Minimal Data Receiver Responsibility - all out-of-order deliverable data delivered

In this case, the NR-SACK chunk is being sent after the data receiver has delivered all deliverable out-of-order DATA chunks to its receiving application(i.e., TSNs 5,6,7,8,13, and 16.) The receiver reserves the right to renege on all undelivered out-of-order DATA chunks(i.e., TSNs 11,14, and 15.)

Type = 0x10	0x00	Chunk Length = 40
Cumulative TSN Ack = 3		
a_rwnd = 4000		
Num of R Gap Ack Blocks = 2	Num of NR Gap Ack Blocks = 3	
Num of Duplicates = 0	0x00	
R Gap Ack Block #1 Start = 8	R Gap Ack Block #1 End = 8	
R Gap Ack Block #2 Start = 11	R Gap Ack Block #2 End = 12	
NR Gap Ack Block #1 Start = 2	NR Gap Ack Block #1 End = 5	
NR Gap Ack Block #2 Start = 10	NR Gap Ack Block #2 End = 10	
NR Gap Ack Block #3 Start = 13	NR Gap Ack Block #3 End = 13	

#### CASE-3: Maximal Data Receiver Responsibility

In this special case, all out-of-order data blocks acknowledged are non-renegable. This case would occur when the data receiver is programmed never to renege, and takes responsibility to deliver all DATA chunks that arrive out-of-order. In this case Num of R Gap Ack Blocks is zero indicating all reported out-of-order TSNs are nr-gap-acked.

Type = 0x10	0x00	Chunk Length = 32
Cumulative TSN Ack = 3		
a_rwnd = 4000		
Num of R Gap Ack Blocks = 0	Num of NR Gap Ack Blocks = 3	
Num of Duplicates = 0	0x00	
NR Gap Ack Block #1 Start = 2	NR Gap Ack Block #1 End = 5	
NR Gap Ack Block #2 Start = 8	NR Gap Ack Block #2 End = 8	
NR Gap Ack Block #3 Start = 10	NR Gap Ack Block #3 End = 13	

#### 4.4. Procedures

The procedures regarding "when" to send an NR-SACK chunk are identical to the procedures regarding when to send a SACK chunk, as outlined in Section 6.2 of [RFC4960].

##### 4.4.1. Sending an NR-SACK chunk

All of the NR-SACK chunk fields identical to the SACK chunk MUST be formed as described in Section 6.2 of [RFC4960].

It is up to the data receiver whether or not to take responsibility for delivery of each out-of-order DATA chunk. An out-of-order DATA chunk that has already been delivered, or that the receiver takes responsibility to deliver (i.e., guarantees not to renege) is Non Renegable(NR), and SHOULD be included in an NR Gap Ack Block field of the outgoing NR-SACK. All other out-of-order data is (R)enegable, and SHOULD be included in R Gap Ack Block field of the outgoing NR-SACK.

Consider three types of data receiver:

CASE-1: Data receiver takes no responsibility for delivery of any out-of-order DATA chunks

CASE-2: Data receiver takes responsibility for all out-of-order DATA chunks that are "deliverable" (i.e., DATA chunks in-sequence within the stream they belong to, or DATA chunks whose (U)nordered bit is 1)

CASE-3: Data receiver takes responsibility for delivery of all out-of-order DATA chunks, whether deliverable or not deliverable

The data receiver SHOULD follow the procedures outlined below for building the NR-SACK.

CASE-1:

- 1A) Identify the TSNS received out-of-order.
- 1B) For these out-of-order TSNS, identify the R Gap Ack Blocks. Fill the Number of R Gap Ack Blocks (N) field, R Gap Ack Block #i Start, and R Gap Ack Block #i End where i goes from 1 to N.
- 1C) Set the Number of NR Gap Ack Blocks (M) field to 0.

CASE-2:

- 2A) Identify the TSNS received out-of-order.
- 2B) For the received out-of-order TSNS, check the (U)nordered bit of each TSN. Tag unordered TSNS as NR.
- 2C) For each stream, also identify the TSNS received out-of-order but are in-sequence within that stream. Tag those in-sequence TSNS as NR.
- 2D) Tag all out-of-order data that is not NR as (R)enegable.
- 2E) For those TSNS tagged as (R)enegable, identify the (R)enegable Blocks. Fill the Number of R Gap Ack Blocks(N) field, R Gap Ack Block #i Start, and R Gap Ack Block #i End where i goes from 1 to N.
- 2F) For those TSNS tagged as NR, identify the NR Blocks. Fill the Number of NR Gap Ack Blocks(M) field, NR Gap Ack Block #i Start, and NR Gap Ack Block #i End where i goes from 1 to M.

CASE-3:

- 3A) Identify the TSNS received out-of-order. All of these TSNS SHOULD be nr-gap-acked.
- 3B) Set the Number of R Gap Ack Blocks (N) field to 0.
- 3C) For these out-of-order TSNS, identify the NR Gap Ack Blocks. Fill the Number of NR Gap Ack Blocks (M) field, NR Gap Ack Block #i Start, and NR Gap Ack Block #i End where i goes from 1 to M.

RFC4960 states that the SCTP endpoint MUST report as many Gap Ack Blocks as can fit in a single SACK chunk limited by the current path MTU. When using NR-SACKs, the SCTP endpoint SHOULD fill as many R Gap Ack Blocks and NR Gap Ack Blocks starting from the Cumulative TSN Ack value as can fit in a single NR-SACK chunk limited by the current path MTU. If space remains, the SCTP endpoint SHOULD fill as many Duplicate TSNs as possible starting from Cumulative TSN Ack value.

#### 4.4.2. Receiving an NR-SACK Chunk

When an NR-SACK chunk is received, all of the NR-SACK fields identical to a SACK chunk SHOULD be processed and handled as in SACK chunk handling outlined in Section 6.2.1 of [RFC4960].

The NR Gap Ack Block Start(s) and NR Gap Ack Block End(s) are offsets relative to the cum-ack. To calculate the actual range of nr-gap-acked TSNs, the cum-ack MUST be added to the Start and End.

For example, assume an incoming NR-SACK chunk's cum-ack is 12 and an NR Gap Ack Block defines the NR Gap Ack Block Start=5, and the NR Gap Ack Block End=7. This NR Gap Ack block nr-gap-acks TSNs 17 through 19 inclusive.

Upon reception of an NR-SACK chunk, all TSNs listed in either R Gap Ack Block(s) or NR Gap Ack Block(s) SHOULD be processed as would be TSNs included in Gap Ack Block(s) of a SACK chunk. All TSNs in all NR Gap Ack Blocks SHOULD be removed from the data sender's retransmission queue as their delivery to the receiving application has either already occurred, or is guaranteed by the data receiver.

Although R Gap Ack Blocks and NR Gap Ack Blocks SHOULD be disjoint sets, NR-SACK processing SHOULD work if an NR-SACK chunk has a TSN listed in both an R Gap Ack Block and an NR Gap Ack Block. In this case, the TSN SHOULD be treated as Non-Renegable.

#### Implementation Note:

Most of NR-SACK processing at the data sender can be implemented by using the same routines as in SACK that process the cum ack and the gap ack(s), followed by removal of nr-gap-acked DATA chunks from the retransmission queue. However, with NR-SACKs, as out-of-order DATA is sometimes removed from the retransmission queue, the gap ack processing routine should recognize that the data sender's retransmission queue has some transmitted data removed. For example, while calculating missing reports, the gap ack processing routine cannot assume that the highest TSN transmitted is always at the tail (right edge) of the retransmission queue.

## 5. Buffer Blocking Mitigation

TBD. See [Dre2012], [PAMS2011], [Globecom2010].

### 5.1. Sender Buffer Splitting

TBD. See [Dre2012], [PAMS2011], [Globecom2010].

### 5.2. Receiver Buffer Splitting

TBD. See [Dre2012], [PAMS2011], [Globecom2010].

### 5.3. Chunk Rescheduling

This algorithm ensures quick blocking resolution for ordered data.

TBD. See [Dre2012], [Globecom2010].

### 5.4. Problems during Path Failure

This section discusses CMT's receive buffer related problems during path failure, and proposes a solution for the same.

#### 5.4.1. Problem Description

Link failures arise when a router or a link connecting two routers fails due to link disconnection, hardware malfunction, or software error. Overloaded links caused by flash crowds and denial-of-service (DoS) attacks also degrade end-to-end communication between peer hosts. Ideally, the routing system detects link failures, and in response, reconfigures the routing tables and avoids routing traffic via the failed link. However, existing research highlights problems with Internet backbone routing that result in long route convergence times. The pervasiveness of path failures motivated us to study their impact on CMT, since CMT achieves better throughput via simultaneous data transmission over multiple end-to-end paths.

CMT is an extension to SCTP, and therefore retains SCTP's failure detection process. A CMT sender uses a tunable failure detection threshold called Path.Max.Retrans (PMR). When a sender experiences more than PMR consecutive timeouts while trying to reach an active destination, the destination is marked as failed. With PMR=5, the failure detection takes 6 consecutive timeouts or 63s. After every timeout, the CMT sender continues to transmit new data on the failed path increasing the chances of receive buffer (rbuf) blocking and degrading CMT performance during permanent and short-term path failures [NEA08].



#### 5.4.2. Solution: Potentially-failed Destination State

To mitigate the rbuf blocking, we introduce a new destination state called 'potentially-failed' state in SCTP (and CMT's) failure detection process [I-D.ietf-tsvwg-sctp-failover]. This solution is based on the rationale that loss detected by a timeout implies either severe congestion or failure en route. After a single timeout on a path, a sender is unsure, and marks the corresponding destination as 'potentially-failed' (PF). A PF destination is not used for data transmission or retransmission. CMT's retransmission policies are augmented to include the PF state. Performance evaluations prove that the PF state significantly reduces rbuf blocking during failure detection [NEA08].

#### 5.5. Non-Renegable SACK

This section discusses problems with SCTP's SACK mechanism and how it affects the send buffer and CMT performance.

##### 5.5.1. Problem Description

Gap-acks acknowledge DATA chunks that arrive out-of-order to a transport layer data receiver. A gap-ack in SCTP is advisory, in that, while it notifies a data sender about the reception of indicated DATA chunks, the data receiver is permitted to later discard DATA chunks that it previously had gap-acked. Discarding a previously gap-acked DATA chunk is known as 'reneging'. Because of the possibility of reneging in SCTP, any gap-acked DATA chunk MUST NOT be removed from the data sender's retransmission queue until the DATA chunk is later CumAacked.

Situations exist when a data receiver knows that reneging on a particular out-of-order DATA chunk will never take place, such as (but not limited to) after an out-of-order DATA chunk is delivered to the receiving application. With current SACKs in SCTP, it is not possible for a data receiver to inform a data sender if or when a particular out-of-order 'deliverable' DATA chunk has been 'delivered' to the receiving application. Thus the data sender MUST keep a copy of every gap-acked out-of-order DATA chunk(s) in the data sender's retransmission queue until the DATA chunk is CumAacked. This use of the data sender's retransmission queue is wasteful. The wasted buffer often degrades CMT performance; the degradation increases when a CMT flow traverses via paths with disparate end-to-end properties [NEY08].

### 5.5.2. Solution: Non-Renegable SACKs

Non-Renegable Selective Acknowledgments (NR-SACKs) Section 4 are a new kind of acknowledgements, extending SCTP's SACK chunk functionalities. The NR-SACK chunk is an extension of the existing SACK chunk. Several fields are identical, including the Cumulative TSN Ack, the Advertised Receiver Window Credit (*a\_rwnd*), and Duplicate TSNs. These fields have the same semantics as described in [RFC4960].

NR-SACKs also identify out-of-order DATA chunks that a receiver either: (1) has delivered to its receiving application, or (2) takes full responsibility to eventually deliver to its receiving application. These out-of-order DATA chunks are 'non-renegable.' Non-Renegable data are reported in the NR Gap Ack Block field of the NR-SACK chunk as described Section 4. We refer to non-renegable selective acknowledgements as 'nr-gap-acks.'

When an out-of-order DATA chunk is nr-gap-acked, the data sender no longer needs to keep that particular DATA chunk in its retransmission queue, thus allowing the data sender to free up its buffer space sooner than if the DATA chunk were only gap-acked. NR-SACKs improve send buffer utilization and throughput for CMT flows [NEY08].

## 6. Handling of Shared Bottlenecks

### 6.1. Introduction

CMT-SCTP assumes all paths to be disjoint. Since each path independently uses a TCP-like congestion control, an SCTP association using N paths over the same bottleneck acquires N times the bandwidth of a concurrent TCP flow. This is clearly unfair. A reliable detection of shared bottlenecks is impossible in arbitrary networks like the Internet. Therefore, [ICC2012] [ConTEL2011], [AINA2010] apply the idea of Resource Pooling to CMT-SCTP. Resource Pooling (RP) denotes 'making a collection of resources behave like a single pooled resource' [WHB09]. The modifications of RP-enabled CMT-SCTP, further denoted as CMT/RP-SCTP, are described in the following subsections. A detailed description of CMT/RP-SCTP, including congestion control examples, can be found in [ICC2012], [ConTEL2011], [AINA2010].

### 6.2. Initial Values

TDB.

### 6.3. Congestion Window Growth

TDB. See [Dre2012], [ICC2012], [ConTEL2011].

### 6.4. Congestion Window Decrease

TDB. See [Dre2012], [ICC2012], [ConTEL2011].

## 7. Chunk Scheduling and Rescheduling

TDB. See [Dre2012], [PFLDNeT2010].

## 8. Socket API Considerations

See [I-D.dreibholz-tsvwg-sctpsocket-multipath] and [I-D.dreibholz-tsvwg-sctpsocket-sqinfo].

## 9. Testbed Platforms

A large-scale and realistic Internet testbed platform with support for the multi-homing feature of the underlying SCTP protocol is NorNet. Particularly, it is also a platform for multi-path transport experiments with CMT-SCTP. A description of and introduction to NorNet is provided in [ComNets2013-Core], [PAMS2013-NorNet], [Haikou2017-2-MultiPath], [Haikou2017-2-NorNet-Tutorial]. Further information can be found on the project website [NorNet-Website] at <https://www.nntb.no>.

An Open Source simulation model of CMT-SCTP is available for OMNeT++ within the INET Framework. See [INET-Framework] for the Git repository. For documentation on the model, together with performance evaluations, see [Dre2012]. Some interesting performance evaluations for delay-sensitive traffic with CMT-SCTP can be found in [ComNets2016-MultipathSurvey].

## 10. IANA Considerations

NOTE to RFC-Editor:

"RFCXXXX" is to be replaced by the RFC number you assign this document.

NOTE to RFC-Editor:

The suggested values for the chunk type and the chunk parameter types are tentative and to be confirmed by IANA.

This document (RFCXXXX) is the reference for all registrations described in this section. The suggested changes are described below.

#### 10.1. A New Chunk Type

A chunk type has to be assigned by IANA. It is suggested to use the values given in Section 4. IANA should assign this value from the pool of chunks with the upper two bits set to '00'.

This requires an additional line in the "Chunk Types" registry for SCTP:

##### Chunk Types

ID Value	Chunk Type	Reference
-----	-----	-----
16	Non-Renegable SACK (NR-SACK)	[RFCXXXX]

The registration table as defined in [RFC6096] for the chunk flags of this chunk type is empty.

#### 11. Security Considerations

This document does not add any additional security considerations in addition to the ones given in [RFC4960].

#### 12. Acknowledgments

The authors wish to thank Hakim Adhari, Phillip Conrad, Jonathan Leighton, Ertugrul Yilmaz and Xing Zhou for their invaluable comments and support.

#### 13. References

##### 13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<https://www.rfc-editor.org/info/rfc4960>>.
- [RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP)

Dynamic Address Reconfiguration", RFC 5061,  
DOI 10.17487/RFC5061, September 2007,  
<<https://www.rfc-editor.org/info/rfc5061>>.

[RFC5351] Lei, P., Ong, L., Tuexen, M., and T. Dreibholz, "An Overview of Reliable Server Pooling Protocols", RFC 5351, DOI 10.17487/RFC5351, September 2008, <<https://www.rfc-editor.org/info/rfc5351>>.

[RFC6096] Tuexen, M. and R. Stewart, "Stream Control Transmission Protocol (SCTP) Chunk Flags Registration", RFC 6096, DOI 10.17487/RFC6096, January 2011, <<https://www.rfc-editor.org/info/rfc6096>>.

[I-D.ietf-tsvwg-sctp-failover]  
Nishida, Y., Natarajan, P., Caro, A., Amer, P. D., and K. E. E. Nielsen, "SCTP-PF: A Quick Failover Algorithm for the Stream Control Transmission Protocol", Work in Progress, Internet-Draft, draft-ietf-tsvwg-sctp-failover-16, 17 February 2016, <<https://www.ietf.org/archive/id/draft-ietf-tsvwg-sctp-failover-16.txt>>.

[I-D.dreibholz-tsvwg-sctpsocket-multipath]  
Dreibholz, T., Becke, M., and H. Adhari, "SCTP Socket API Extensions for Concurrent Multipath Transfer", Work in Progress, Internet-Draft, draft-dreibholz-tsvwg-sctpsocket-multipath-23, 6 September 2021, <<https://www.ietf.org/archive/id/draft-dreibholz-tsvwg-sctpsocket-multipath-23.txt>>.

[I-D.dreibholz-tsvwg-sctpsocket-sqinfo]  
Dreibholz, T., Seggelmann, R., and M. Becke, "Sender Queue Info Option for the SCTP Socket API", Work in Progress, Internet-Draft, draft-dreibholz-tsvwg-sctpsocket-sqinfo-23, 6 September 2021, <<https://www.ietf.org/archive/id/draft-dreibholz-tsvwg-sctpsocket-sqinfo-23.txt>>.

### 13.2. Informative References

[I06] Iyengar, J., "End-to-End Concurrent Multipath Transfer Using Transport Layer Multihoming", PhD Dissertation Computer Science Dept., University of Delaware, April 2006, <<https://www.eecis.udel.edu/~amer/PEL/poc/pdf/IyengarPhDdissertation.pdf>>.

[IAS06] Iyengar, J., Amer, P. D., and R. R. Stewart, "Concurrent Multipath Transfer Using SCTP Multihoming Over Independent

- End-to-End Paths", Journal IEEE/ACM Transactions on Networking, October 2006,  
<<https://www.eecis.udel.edu/~amer/PEL/poc/pdf/ToN2006-CMT-over-Independent-Paths-Iyengar.pdf>>.
- [NEA08] Natarajan, P., Ekiz, N., Iyengar, J., Amer, P., and R. Stewart, "Concurrent Multipath Transfer Using Transport Layer Multihoming: Introducing the Potentially-failed Destination State", Proceedings of the IFIP Networking, May 2008,  
<<http://dl.ifip.org/db/conf/networking/networking2008/NatarajanEAIS08.pdf>>.
- [NEY08] Natarajan, P., Ekiz, N., Yilmaz, E., Amer, P., Iyengar, J., and R. Stewart, "Non-Renegable Selective Acknowledgments (NR-SACKs) for SCTP", Proceedings of the 16th IEEE International Conference on Network Protocols (ICNP) , October 2008,  
<<http://www.ieee-icnp.org/2008/papers/Index19.pdf>>.
- [WHB09] Wischik, D., Handley, M., and M. B. Braun, "The Resource Pooling Principle", Journal ACM SIGCOMM Computer Communication Review, October 2009,  
<<http://haig.cs.ucl.ac.uk/staff/M.Handley/papers/respool-ccr.pdf>>.
- [OMNeTWorkshop2010-SCTP]  
Dreibholz, T., Becke, M., Pulinthanath, J., and E. P. Rathgeb, "Implementation and Evaluation of Concurrent Multipath Transfer for SCTP in the INET Framework", Proceedings of the 3rd ACM/ICST International Workshop on OMNeT++ ISBN 978-963-9799-87-5,  
DOI 10.4108/ICST.SIMUTOOLS2010.8673, 19 March 2010,  
<[https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/SCTP/Paper/OMNeT\\_Workshop2010-SCTP.pdf](https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/SCTP/Paper/OMNeT_Workshop2010-SCTP.pdf)>.
- [AINA2010] Dreibholz, T., Becke, M., Pulinthanath, J., and E. P. Rathgeb, "Applying TCP-Friendly Congestion Control to Concurrent Multipath Transfer", Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA) Pages 312-319,  
ISBN 978-0-7695-4018-4, DOI 10.1109/AINA.2010.117, 21 April 2010, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/SCTP/Paper/AINA2010.pdf>>.
- [YEN10] Yilmaz, E., Ekiz, N., Natarajan, P., Amer, P., Leighton, J., Baker, F., and R. Stewart, "Throughput Analysis of

Non-Renegable Selective Acknowledgments (NR-SACKs) for SCTP", Computer Communications, doi:10.1016/j.comcom.2010.06.028, 2010.

[PFLDNeT2010]

Dreibholz, T., Seggelmann, R., Tüxen, M., and E. P. Rathgeb, "Transmission Scheduling Optimizations for Concurrent Multipath Transfer", Proceedings of the 8th International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNeT) Volume 8, ISSN 2074-5168, 29 November 2010, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/SCTP/Paper/PFLDNeT2010.pdf>>.

[Globecom2010]

Dreibholz, T., Becke, M., Rathgeb, E. P., and M. Tüxen, "On the Use of Concurrent Multipath Transfer over Asymmetric Paths", Proceedings of the IEEE Global Communications Conference (GLOBECOM) ISBN 978-1-4244-5637-6, DOI 10.1109/GLOCOM.2010.5683579, 7 December 2010, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/SCTP/Paper/Globecom2010.pdf>>.

[PAMS2011] Adhari, H., Dreibholz, T., Becke, M., Rathgeb, E. P., and M. Tüxen, "Evaluation of Concurrent Multipath Transfer over Dissimilar Paths", Proceedings of the 1st International Workshop on Protocols and Applications with Multi-Homing Support (PAMS) Pages 708-714, ISBN 978-0-7695-4338-3, DOI 10.1109/WAINA.2011.92, 22 March 2011, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/SCTP/Paper/PAMS2011.pdf>>.

[ConTEL2011]

Dreibholz, T., Becke, M., Adhari, H., and E. P. Rathgeb, "On the Impact of Congestion Control for Concurrent Multipath Transfer on the Transport Layer", Proceedings of the 11th IEEE International Conference on Telecommunications (ConTEL) Pages 397-404, ISBN 978-953-184-152-8, 16 June 2011, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/SCTP/Paper/ConTEL2011.pdf>>.

[ICC2012]

Becke, M., Dreibholz, T., Adhari, H., and E. P. Rathgeb, "On the Fairness of Transport Protocols in a Multi-Path Environment", Proceedings of the IEEE International Conference on Communications (ICC) Pages 2666-2672,

ISBN 978-1-4577-2052-9, DOI 10.1109/ICC.2012.6363695, 12 June 2012, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/SCTP/Paper/ICC2012.pdf>>.

[ComNets2016-MultipathSurvey]

Yedugundla, K. V., Ferlin, S., Dreibholz, T., Alay, Ö., Kuhn, N., Hurtig, P., and A. Brunström, "Is Multi-Path Transport Suitable for Latency Sensitive Traffic?", Computer Networks Volume 105, Pages 1-21, ISSN 1389-1286, DOI 10.1016/j.comnet.2016.05.008, 4 August 2016, <<https://www.simula.no/file/comnets2016-multipathsurvey.pdf/download>>.

[Dre2012] Dreibholz, T., "Evaluation and Optimisation of Multi-Path Transport using the Stream Control Transmission Protocol", Habilitation Treatise, 13 March 2012, <[https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-29737/Dre2012\\_final.pdf](https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-29737/Dre2012_final.pdf)>.

[NorNet-Website]

Dreibholz, T., "NorNet -- A Real-World, Large-Scale Multi-Homing Testbed", Online: <https://www.nntb.no/>, 2016, <<https://www.nntb.no/>>.

[PAMS2013-NorNet]

Dreibholz, T. and E. G. Gran, "Design and Implementation of the NorNet Core Research Testbed for Multi-Homed Systems", Proceedings of the 3rd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS) Pages 1094-1100, ISBN 978-0-7695-4952-1, DOI 10.1109/WAINA.2013.71, 27 March 2013, <<https://www.simula.no/file/threfereedinproceedingsreference2012-12-207643198512pdf/download>>.

[ComNets2013-Core]

Gran, E. G., Dreibholz, T., and A. Kvalbein, "NorNet Core - A Multi-Homed Research Testbed", Computer Networks, Special Issue on Future Internet Testbeds Volume 61, Pages 75-87, ISSN 1389-1286, DOI 10.1016/j.bjp.2013.12.035, 14 March 2014, <<https://www.simula.no/file/simulasimula2236pdf/download>>.

[INET-Framework]

Hornig, R. and A. Varga, "INET Framework Git Repository", Online: <https://github.com/inet-framework/inet>, 2016, <<https://github.com/inet-framework/inet>>.



[Haikou2017-2-MultiPath]

Dreibholz, T., "An Introduction to Multi-Path Transport at Hainan University", Keynote Talk at Hainan University, College of Information Science and Technology (CIST), 14 December 2017, <<https://www.simula.no/file/haikou2017-multipath-presentationpdf-0/download>>.

[Haikou2017-2-NorNet-Tutorial]

Dreibholz, T., "NorNet Core Beginner Tutorial at Hainan University", Tutorial at Hainan University, College of Information Science and Technology (CIST), 15 December 2017, <<https://www.simula.no/file/haikou2017-nornet-tutorialpdf-0/download>>.

#### Authors' Addresses

Paul D. Amer  
University of Delaware, Computer and Information Sciences Department  
Newark, Delaware 19716  
United States of America

Phone: +1-302-831-1944  
Email: [amer@cis.udel.edu](mailto:amer@cis.udel.edu)  
URI: <https://www.eecis.udel.edu/~amer/>

Martin Becke  
HAW Hamburg, Informatics Department  
Berliner Tor 7  
20099 Hamburg  
Germany

Phone: +49-40-42875-8104  
Email: [martin.becke@haw-hamburg.de](mailto:martin.becke@haw-hamburg.de)  
URI: <http://www.scimbe.de/about.html>

Thomas Dreibholz  
Simula Metropolitan Centre for Digital Engineering  
Pilestredet 52  
0167 Oslo  
Norway

Phone: +47-6782-8200  
Email: [dreibh@simula.no](mailto:dreibh@simula.no)  
URI: <https://www.simula.no/people/dreibh>

Nasif Ekiz  
University of Delaware, Computer and Information Sciences Department  
Newark, Delaware 19716  
United States of America

Email: nekiz@udel.edu

Janardhan Iyengar  
Franklin and Marshall College, Mathematics and Computer Science  
PO Box 3003  
Lancaster, Pennsylvania 17604-3003  
United States of America

Phone: +1-717-358-4774  
Email: jiyengar@fandm.edu

Preethi Natarajan  
Cisco Systems  
425 East Tasman Drive  
San Jose, California 95134  
United States of America

Email: prenatar@cisco.com

Randall R. Stewart  
Netflix  
Chapin, South Carolina 29036  
United States of America

Email: randall@lakerest.net

Michael Tuexen  
Muenster University of Applied Sciences  
Stegerwaldstrasse 39  
48565 Steinfurt  
Germany

Email: tuexen@fh-muenster.de  
URI: <https://www.fh-muenster.de/fb2/personen/professoren/tuexen/>