

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 24, 2014

S. Dhesikan  
C. Jennings  
Cisco  
D. Druta, Ed.  
ATT  
P. Jones  
J. Polk  
Cisco  
June 22, 2014

DSCP and other packet markings for RTCWeb QoS  
draft-dhesikan-tsvwg-rtcweb-qos-07

Abstract

Many networks, such as service provider and enterprise networks, can provide per packet treatments based on Differentiated Services Code Points (DSCP) on a per hop basis. This document provides the recommended DSCP values for browsers to use for various classes of traffic.

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 December 24, 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. Relation to Other Standards . . . . .	3
3. Terminology . . . . .	3
4. Inputs . . . . .	3
5. DSCP Mappings . . . . .	4
6. Security Considerations . . . . .	6
7. IANA Considerations . . . . .	6
8. Downward References . . . . .	6
9. Acknowledgements . . . . .	6
10. Document History . . . . .	6
11. References . . . . .	6
11.1. Normative References . . . . .	6
11.2. Informative References . . . . .	6
Authors' Addresses . . . . .	7

## 1. Introduction

Differentiated Services Code Points (DSCP)[RFC2474] style packet marking can help provide QoS in some environments. There are many use cases where such marking does not help, but it seldom makes things worse if packets are marked appropriately. In other words, if too many packets, say all audio or all audio and video, are marked for a given network condition then it can prevent desirable results. Either too much other traffic will be starved, or there is not enough capacity for the preferentially marked packets (i.e., audio and/or video).

This draft proposes how WebRTC applications can mark packets. This draft does not contradict or redefine any advice from previous IETF RFCs but simply provides a simple set of recommendations for implementers based on the previous RFCs.

There are some environments where priority markings frequently help. These include:

1. Private networks (Wide Area).
2. If the congested link is the broadband uplink in a Cable or DSL scenario, often residential routers/NAT support preferential treatment based on DSCP.

3. If the congested link is a local WiFi network, marking may help.

Traditionally DSCP values have been thought of as being site specific, with each site selecting its own code points for each QoS level. However in the RTCWeb use cases, the browsers need to set them to something when there is no site specific information. This document describes a reasonable default set of DSCP code point values drawn from existing RFCs and common usage. These code points are solely defaults. Future drafts may define mechanisms for site specific mappings to override the values provided in this draft.

This draft defines some inputs that the browser in an WebRTC application can look at to determine how to set the various packet markings and defines the mapping from abstract QoS policies (data type, priority level) to those packet markings.

## 2. Relation to Other Standards

This specification does not change or override the advice in any other standards about setting packet markings. It simply provides a summary of them and provides the context of how they relate into the RTCWeb context. In some cases, such as DSCP where the normative RFC leaves open multiple options to choose from, this clarifies which choice should be used in the RTCWeb context. This document also specifies the inputs that are needed by the browser to provide to the media engine.

The DSCP value set by the endpoint is not always trusted by the network. Therefore, the DSCP value may be remarked to any other DSCP, even to best effort at the network edge through policy. The mitigation for such action is through an authorization mechanism. Such authorization mechanism is outside the scope of this document.

## 3. Terminology

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

## 4. Inputs

The below uses the concept of a media flow, however these are commonly not equivalent to a transport flow, i.e. as defined by a 5-tuple (source address, destination address, source port, destination port, and protocol). Instead each media flow contains all the packets associated with an independent media entity within one 5-tuple. There may be multiple media flows within the same 5-tuple. These media flows might be consisting of different media types and have different priorities. The following are the inputs that the browser provides to the media engine:

- o Data Type: The browser provides this input as it knows if the flow is audio, interactive video with or without audio, non-interactive video with or without audio, or data.
- o Priority: Another input is the relative treatment of the flow within that data type. Many applications have multiple media flows of the same data type and often some are more important than others. Likewise, in a video conference where the flows in the conference is of the same data type but contains different media types, the flow for audio may be more important than the video flow. JavaScript applications can tell the browser whether a particular media flow is high, medium, low or very low importance to the application.

When it comes to data transmission, a media (data) flow is the SCTP stream under a common congestion control (currently within the same SCTP association).

[I-D.ietf-rtcweb-transports] defines in more detail what an individual media flow is within the WebRTC context.

## 5. DSCP Mappings

Below is a table of DSCP markings for each data type of interest to RTCWeb. These DSCPs for each data type listed are a reasonable default set of code point values taken from [RFC4594]. A web browser SHOULD use these values to mark the appropriate media packets. More information on EF can be found in [RFC3246]. More information on AF can be found in [RFC2597].

Data Type	Very Low	Low	Medium	High
Audio	CS1 (8)	BE (0)	EF (46)	EF (46)
Interactive Video with or without audio	CS1 (8)	BE (0)	AF42, AF43 (36, 38)	AF41, AF42 (34,

				36)
Non-Interactive Video with or without audio	CS1 (8)	BE (0)	AF32, AF33 (28, 30)	AF31, AF32 (26, 28)
Data	CS1 (8)	BE (0)	AF1x (10, 12, 14)	AF2x (18, 20, 22)

Table 1

The columns "very low", "low", "Medium" and "high" are the priority levels. The browser app SHOULD first select the data type of the media flow. Within the data type, the priority of the media flow SHOULD be selected. All packets within a media flow SHOULD have the same priority. In some cases, the selected cell may have multiple DSCP values, such as AF41 and AF42. These offer different drop precedences. One may select difference drop precedences for the different packets in the media flow. Therefore, all packets in the stream SHOULD be marked with the same priority but can have difference drop precedences.

The combination of data type and priority provides specificity and helps in selecting the right DSCP value for the media flow. In some cases, the different drop precedence values provides additional granularity in classifying packets within a media flow. For example: In a video conference, the video media flow may be medium priority. If so, either AF42 or AF43 may be selected. If the I frames in the stream are more important than the P frames then the I frames can be marked with AF42 and the P frames marked with AF43.

The above table assumes that packets marked with CS1 is treated as "less than best effort". However, the treatment of CS1 is implementation dependent. If an implementation treats CS1 as other than "less than best effort", then the priority of the packets may be changed from what is intended.

If a packet enters a QoS domain that has no support for the above defined Data Types/Application classes, then the network node at the edge will remark the DSCP value based on policies. Subsequently, if the packet enters a QoS domain that supports a larger number of Data types/Application (service) classes, there may not be sufficient information in the packet to restore the original markings. Mechanisms for restoring such original DSCP is outside the scope of this document.

## 6. Security Considerations

This draft does not add any additional security implication other than the normal application use of DSCP. For security implications on use of DSCP, please refer to Section 6 of RFC 4594 . Please also see work-in-progress draft draft-ietf-rtcweb-security-04 as an additional reference.

## 7. IANA Considerations

This specification does not require any actions from IANA.

## 8. Downward References

This specification contains a downwards reference to [RFC4594] however the parts of that RFC used by this specification are sufficiently stable for this downward reference.

## 9. Acknowledgements

Thanks To David Black, Magnus Westerland, Paolo Severini, Jim Hasselbrook, Joe Marcus, and Erik Nordmark for their help.

## 10. Document History

Note to RFC Editor: Please remove this section.

This document was originally an individual submission in RTCWeb WG. The RTCWeb working group selected it to be become a WG document. Later the transport ADs requested that this be moved to the TSVWG WG as that seemed to be a better match. This document is now being submitted as individual submission to the TSVWG with the hope that WG will select it as a WG draft and move it forward to an RFC.

## 11. References

### 11.1. 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.

### 11.2. Informative References

- [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.
- [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., 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.

## Authors' Addresses

Subha Dhesikan  
Cisco

Email: [sdhesika@cisco.com](mailto:sdhesika@cisco.com)

Cullen Jennings  
Cisco

Email: [fluffy@cisco.com](mailto:fluffy@cisco.com)

Dan Druta (editor)  
ATT

Email: [dd5826@att.com](mailto:dd5826@att.com)

Paul Jones  
Cisco

Email: [paulej@packetizer.com](mailto:paulej@packetizer.com)

James Polk  
Cisco

Email: [jmpolk@cisco.com](mailto:jmpolk@cisco.com)

TSVWG  
Internet-Draft  
Intended status: Informational  
Expires: August 18, 2014

R. Geib, Ed.  
Deutsche Telekom  
February 14, 2014

DiffServ interconnection classes and practice  
draft-geib-tsvwg-diffserv-intercon-05

Abstract

This document proposes a limited and well defined set of QoS PHBs and PHB groups to be applied at (inter)connections of two separately administered and operated networks. Many network providers operate Aggregated DiffServ classes. This draft contains DiffServ aggregation friendly interconnection concepts.

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 18, 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. Related work . . . . .	5
2. Terminology . . . . .	5
3. Aggregating PHBs of a class by a DSCP Precedence Prefix . . . .	6
4. An Interconnection class and codepoint scheme . . . . .	6
4.1. Treatment of Network Control traffic at carrier interconnection interfaces . . . . .	9
5. DiffServ Intercon relation to other QoS standards . . . . .	10
5.1. MPLS, Ethernet and DSCP Precedence Prefixes for aggregated classes . . . . .	11
5.2. Proposed GSMA IR.34 to DiffServ Intercon mapping . . . . .	11
5.3. Proposed MEF 23.1 to DiffServ Intercon mapping . . . . .	12
6. Contributors . . . . .	14
7. Acknowledgements . . . . .	14
8. IANA Considerations . . . . .	14
9. Security Considerations . . . . .	14
10. References . . . . .	14
10.1. Normative References . . . . .	14
10.2. Informative References . . . . .	15
Appendix A. Change log . . . . .	16
Author's Address . . . . .	16

## 1. Introduction

DiffServ has been deployed in many networks. As described by section 2.3.4.2 of RFC 2475, remarking of packets at domain boundaries is a DiffServ feature [RFC2475]. This draft proposes a set of standard QoS classes and code points at interconnection points to which and from which locally used classes and code points should be mapped.

IP precedence has been deprecated. MPLS and Ethernet support 3 bit code point fields to differentiate service quality (see MPLS TC / Traffic Class [RFC5462] and PCP, Priority Code Point [IEEE802.1Q]). The concept of classifying DiffServ traffic classes by the bits 0-2 of a DSCP has been part of Diffserv from start on. This is also reflected by the DiffServ codepoint definitions of AF and EF. It is common practice today also to copy these three DSCP bits into MPLS TC or Ethernet P-Bits. PHBs based on DSCP bit 0-2 classification may be applied in core network sections rather than then DSCP based PHBs. Network providers make use of this feature for their own IP QoS concepts. This draft suggests to expand it to interconnections between operators of different domains in a simple manner while each operator may maintain the own class and codepoint scheme within the own domain.

The scope of this draft is limited to 4 specified interconnection classes having four different 3 bit code points in DSCP bits 0-2. Using more than the 4 proposed IP precedences at interconnection could result in non-revertible IP Precedence or DSCP rewrites and avoid sustaining end-to-end QoS classes, if a receiving provider operates more than 4 MPLS TCs. Assume a provider operating 4 QoS classes available at interconnection and MPLS within his backbone. Further assume this carrier to support MPLS based ECN marking and assume this carrier to operate a newtork control class with an own MPLS TC. Two codepoints are left for future use. If 5 or more PHBs each with different DSCP bits 0-2 are offerd at an interconnection point and no more than a single MPLS label needs to be pushed, two (or more) PHBs will carry the same DSCP bits 0-2 after re-marking to the provider internal QoS scheme. Due to MPLS pen ultimate hop popping, DSCPs must be re-written in this case. That may work if bits 3-5 of the DSCP can be varied without introducing ambiguities. Should this traffic later pass another QoS interconnection point further downstream, the original sending domain may not be able to ensure proper class mapping for the PHBs merged into a single class by the receiving domain.

At first glance, the interconnection codepoint scheme looks like an additional effort. But there are some obvious benefits: each party sending or receiving traffic has to specify the mapping from or to the interconnection class and code point scheme only once. Without

it, this is to be negotiated per interconnection party individually. Further, end-to-end QoS in terms of traffic being classified for the same class in all passed domains is more likely to result if an interconnection code point scheme is used. It is not necessarily resulting from individual per provider mapping negotiations, as can be seen from the example given above.

The standards and deployments known to the author of this draft are limited to 4 DiffServ classes at interconnection points (or less). The example given in RFC 5127 on aggregation of DiffServ service classes picks 4 Treatment aggregates (note that this document prefers class instead of treatment aggregate). Reasons to favour working with 4 DiffServ interconnection classes:

- o There should be a coding reserve for interconnection classes. This leaves space for future standards, for private bilateral agreements and for provider internal classes.
- o The fields available for carrying QoS information (e.g., DiffServ PHB) in MPLS and Ethernet are only 3 bits in size, and are intended for more than just QoS purposes (see e.g. [RFC5129]).
- o Migrations from one code point scheme to another may require spare QoS code points.

IP Precedence has been deprecated when DiffServ was standardised. It is common practice today however to copy the DSCPs Bits 0-2 (called DSCP Precedence Prefix in the following) into MPLS TC or Ethernet P-Bits. This is also reflected by the DiffServ codepoint definitions of AF and EF. Class based PHBs may be applied in core network sections rather than then DSCP based PHBs.

The set of available router and traffic management tools to configure and operate DiffServ classes is limited. This should be reflected by class definitions. These may in the end be more related to transport properties (e.g., whether the traffic in a class is congestion controlled or not) than to application requirements.

RFC5127 provides recommendations on domain internal aggregation of DiffServ traffic and offers a deployment example [RFC5127]. This draft differs from the RFC5127 aggregation deployment example in the following points:

- o the basic concept of this draft is to maintain classes, while expecting DSCP remarking at provider edges.
- o This draft follows RFC4594 in the proposed marking of provider Network Control traffic and expands RFC4594 on treatment of CS6

marked traffic at interconnection points (see section 5.2).

The proposed Interconnection class and code point scheme tries to reflect and consolidate related DiffServ and QoS standardisation efforts outside of the IETF, namely MEF [MEF 23.1], GSMA [IR.34] and ITU [Y.1566]. GSMAs IR.34 specifies an inter provider VPN, but it is nevertheless specifying a kind of DiffServ aware IP based carrier interconnection.

### 1.1. Related work

In addition to the standardisation activities which triggered this work, other authors published RFCs or drafts which may benefit from an interconnection class- and codepoint scheme. RFC 5160 suggests Meta-QoS- Classes to enable deployment of standardised end to end QoS classes [RFC5160]. The authors agree that the proposed interconnection class- and codepoint scheme as well as the idea of standardised end to end classes would complement their own work. Work on signaling Class of Service at interconnection interfaces by BGP [I-D.knoll-idr-cos-interconnect], [ID.idr-sla] is beyond the scope of this draft. Should the basic transport and class properties be standardised as proposed here, signaled access to QoS classes may be of interest. The current BGP drafts focus on exchanging SLA and traffic conditioning parameters. They seem to assume that common interpretation of the PHB properties identified by DSCPs has been established prior to exchanging further details by BGP signaling.

## 2. Terminology

This draft re-uses existing terminology.

**DSCP Precedence Prefix** Bits 0-2 of the DSCP ("x" in this generic DSCP: xxxddd) are called the DSCP Precedence Prefix. Section 4.2 of [RFC2474] discusses the role of these bits in enabling use of DiffServ with network equipment that is not fully DiffServ- compliant; this term provides a formal for these bits that is preferable to referring to them as "the former IP Precedence field".

**DSCP Precedence Class** This is a set of one or more PHBs that utilize the same DSCP Precedence Prefix on an interconnection between two networks.

### 3. Aggregating PHBs of a class by a DSCP Precedence Prefix

Configuration and operation of MPLS networks is simplified, if a DSCP Precedence Class can be aggregated into a single PSC by classifying them on their DSCP Precedence Prefix. The DSCP Precedence Prefix of an interconnection DSCP Precedence Class may be rewritten at the ingress edge router and then simply be copied into the MPLS TC field of one or more labels to be pushed.

To allow for simple carrier interconnection agreements, carriers sending traffic belonging to the same class but marked by DSCPs with differing DSCP Precedence Prefixes should apply the interconnection marking and code point scheme specified below, if they interconnect to a carrier applying DSCP Precedence Prefix based traffic aggregation. An example where this may be applicable is the Interactive Class of GSMA IR.34 [IR.34]). Another option is to negotiate a customised interconnection agreement of course.

Classification by DSCP Precedence Prefix is useful for links aggregating DiffServ traffic. DSCP Precedence Prefix based classification is not recommended as a general mode of operation. Edge systems, QoS policy enforcement nodes, service areas and hosts benefit from fine grained DSCP based classification and should continue to do so.

RFC 2474 specifies the Class Selector Codepoints [RFC2474]. These offer a similar concept, but they are strictly limited to xxx000 DSCPs. The Class Selector Code points don't offer aggregation, they just simplify classification. This draft intends to aggregate several PHBs of a single class by a DSCP Precedence Prefix, which a different concept than that of the Class Selector Code points.

### 4. An Interconnection class and codepoint scheme

Interconnecting parties face the problem of matching classes to be interconnected and then to agree on code point mapping. As stated by the DiffServ Architecture [RFC2475], remarking is a standard behaviour at interconnection interfaces. This draft proposes a standard interconnection set of 4 DSCP precedence classes with well defined DSCP and DSCP Precedence Prefix values. A sending party remarks DSCPs from internal schemes to the Interconnection code points. The receiving party remarks DSCP Precedence Prefixes and / or DSCPs to her internal scheme. Thus the interconnection code point scheme fully complies with the DiffServ architecture.

This draft picks up the DiffServ interconnection class definitions proposed by ITU-T Y.1566 [Y.1566]. In addition to the classes

defined there, this draft proposes a complete set of PHBs and DSCPs. Like in the example given by RFC 5127 for domain internal class aggregation, Y.1566 specifies four PHB scheduling classes (for provider interconnection however). Their properties slightly from those of the RFC5127 example:

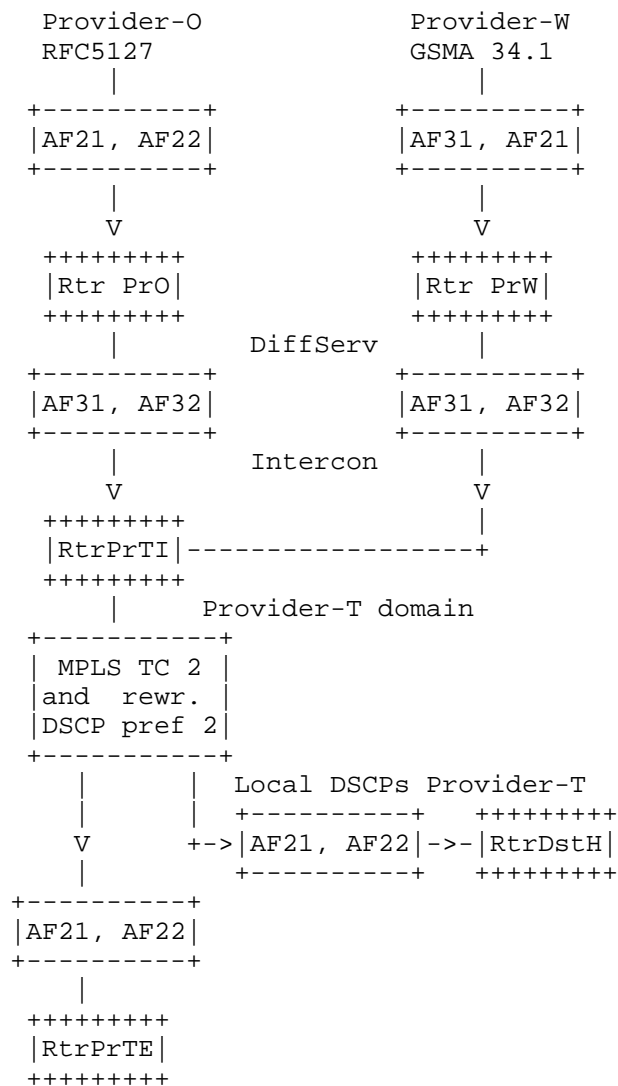
**Class Priority:** PHB EF, DSCP 101 110. The figures of merit describing the PHB to be in the range of low single digit milliseconds. See [RFC3246]. This class corresponds to RFC 5127's real time class, but it is limited to traffic for which node configuration "ensures that the service rate of EF packets on a given output interface exceeds their arrival rate at that interface over long and short time intervals" (see RFC 3246).

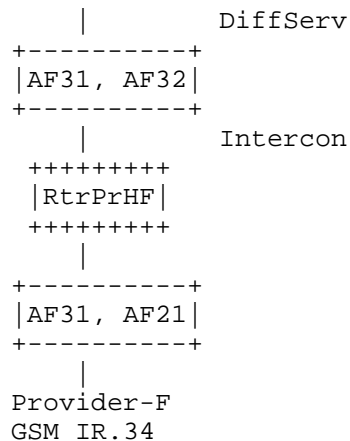
**Bulk inelastic:** PHB AF41, DSCP 100 010 (the other AF4 PHB group PHB's and DSCPs should be reserved for future extension of this DSCP scheduling class). Optimised for low loss, low delay, low jitter at high bandwidth. Traffic load in this class must be controlled, e.g. by application servers. One example could be flow admission control. There may be infrequent retransmissions requested by the application layer to mitigate low levels of packet losses. Discard of packets through active queue management should be avoided in this class. Congestion in this class may result in bursty packet loss. If used to carry multimedia traffic, it is recommended to carry audio and video traffic in a single PHB (note that video conferencing may require separate PHBs for audio and video traffic, which could also be realised by utilising two AF 4 PHBs). All of these properties influence the buffer design. This class is designed to transport those parts of RFC 5127's Real Time class, which consume considerable QoS bandwidth at the interconnection interface.

**Assured:** The complete PHB group AF3, DSCPs 011 010, 011 100 and 011 110. This class may be optimised to transport traffic without bandwidth requirements. It aims on very low loss at high bandwidths. Retransmissions after losses characterise the class and influence the buffer design. Active queue management with probabilistic dropping may be deployed. The RFC 5127 example calls this class Assured Elastic.

**Default:** Default PHB, CS0 with DSCP 000 000. This class may be optimised to transport traffic without bandwidth requirements. Retransmissions after losses characterise the class and influence the buffer design. Active queue management with probabilistic dropping may be deployed. The RFC 5127 example calls this class Elastic.

The idea is illustrated by an example. Provider O and provider W are peer with provider T. They have agreed upon a QoS interconnection. Traffic of provider O terminates within provider Ts network, while the GSMA IR.34 traffic transits through the network of provider T to provider F. Assume all providers to run their own internal codepoint schemes for a class with properties of the DiffServ Intercon Assured class. See section for a description of GSMA IR.34.





DiffServ Intercon example

Figure 1

It is easily visible that all providers only need to deploy internal DSCP to DiffServ Intercon DSCP mappings to exchange traffic in the desired classes.

RFC5127 specifies a separate PHB scheduling class for network control traffic. This class may be present at interconnection interfaces too, but depending on the agreement between providers, it may also be classified for another interconnection class. See section 4.2 for a detailed discussion.

The proposed class and code point scheme is designed for point to point IP layer interconnections. Other types of interconnections are out of scope of this document. The basic class and code point scheme is applicable on Ethernet layer too.

#### 4.1. Treatment of Network Control traffic at carrier interconnection interfaces

As specified by RFC4594, section 3.2, Network Control (NC) traffic marked by CS6 is to be expected at interconnection interfaces. This document does not change NC specifications of RFC4594. The latter specification is detailed on domain internal NC traffic and on traffic exchanged between peering points. Further, it recommends not to forward CS6 marked traffic originating from user-controlled end points by the NC class of a provider domain.



As a minor clarification to RFC4594, "peering" shouldn't be interpreted in a commercial sense. The NC PHB is applicable also in the case of a purchased network service based on a transit agreement with an upstream provider. RFC4594 recommendations on NC traffic are applicable for IP carrier interconnections in general.

Some CS6 traffic exchanged accross carrier interconnections will terminate at the domain ingress node (e.g., if BGP is running between the two routers on opposite ends of the interconnection link).

An IP carrier may limit access to the NC PHB for traffic which is recognised as network control traffic relevant to the own domain. Interconnecting carriers should specify treatment of CS6 marked traffic received at a carrier interconnection which is to be forwarded beyond the ingress node. An SLA covering the following cases is recommended, if a carrier wishes to send CS6 marked traffic accross an interconnection link which isn't terminating at the interconnected ingress node:

- o classification of traffic which is network control traffic for both domains. This traffic should be classified and marked for the NC PHB.
- o classification of traffic which is network control traffic for the sending domain only. This traffic should be classified for a PHB offering similar properties as the NC class (e.g. AF31 as specified by this document). As an example GSMA IR.34 proposes an Interactive class / AF31 to carry SIP and DIAMETER traffic. While this is service control traffic of high importance to the interconnected Mobile Network Operators, it is certainly no Network Control traffic for a fixed network providing transit. The example may not be perfect. It was picked nevertheless because it refers to an existing standard.
- o any other CS6 marked traffic should be remarked or dropped.

## 5. DiffServ Intercon relation to other QoS standards

This sections provides suggestions how to aggregate traffic by DSCP Precedence Prefexies to Ethernet and MPLS. Other Standardisation Organsiations deal with QoS aware provider interconnection. As IP is the state of the art realisation of provider interconnections, these standards bodies specify DiffServ aware interconnections. Some of these bodies are industry alliances chartered also to promote interconnection of wireless or Ethernet technology including the exchange of IP datagrams at interconnection points. Examples are the Metro Ethernet Forum (MEF) or the GSM Alliance (GSMA). The ITU was

mentioned earlier. ITU works across and between responsibilities of different Standardisation Organisations and liaising with them, if their responsibilities are touched, is traditional part of that process.

#### 5.1. MPLS, Ethernet and DSCP Precedence Prefixes for aggregated classes

The interconnection class and code point scheme respects properties and limits of a 3 bit PHB coding space in different ways:

- o it allows to classify four interconnection classes based on the DSCP Precedence Prefix.
- o it supports a single PHB group (AF3), whose DSCPs may be aggregated into a single MPLS TC (or Ethernet priority) based on their joint DSCP Precedence Prefix. This kind of aggregation will work for the AF4 PHB group, if the PHBs AF42 and AF43 need to be supported in addition to AF41.
- o Applying only 4 aggregated classes at interconnection allows for bilateral extensions, if desired. Should two carriers agree to map AF32 and AF33 to an additional individual MPLS TC and offer an Ordered Aggregate across the interconnecting domain, this proposal at leaves some MPLS TC coding space for such an extension (although this draft doesn't recommend interconnections of that type).

The above statement is no requirement to deprecate any DSCP to MPLS TC or Ethernet P-Bit mapping functionality. In the opposite, by limiting the interconnection scheme to 6 PHBs, each PHB may be mapped to an individual Traffic Class or Priority also within MPLS or Ethernet (if desired).

#### 5.2. Proposed GSMA IR.34 to DiffServ Intercon mapping

GSMA IR.34 provides guidelines on how to set up and interconnect Internet Protocol (IP) Packet eXchange (IPX) Networks [IR.34]. An IPX network is an inter-Service Provider IP backbone which comprises the interconnected networks of IPX Providers. IPX is a telecommunications interconnection model for the exchange of IP based traffic between customers of separate mobile and fixed operators as well as other types of service provider (such as ISP), via IP based network-to-network interface. Note that IPX is not a public interconnection model however, it is designed as a private IP Backbone of the interconnected parties. Two IPX partners may interconnect using transit offered by an Inter-Service Provider IP Backbone. This requires an IP QoS aware interconnection as described by this draft between IPX provider and Inter-Service Provider IP

Backbone.

GSMA IR.34 specifies 4 aggregated classes and 7 PHBs. Mapping to DiffServ Intercon is smooth apart from the GSMA aggregated class Interactive, which consists of 4 PHBs. The table below lists a suggested mapping to DiffServ Intercon.

GSMA IR.34		DiffServ-Intercon	
Agg. Class	PHB	Agg. Class	PHB
Conversational	EF	Priority	EF
Streaming	AF41	Bulk inelastic	AF41
Interactive	AF31	Assured	AF31
(ordered by	AF32		
priority,			AF32
AF3 highest)	AF21		
	AF11		AF33
Background	CS0	Default	CS0

Suggested mapping of GSMA IR.34 classes and PHBs to DiffServ Intercon

Figure 2

The motivation resulting in the design of the IR.34 Interactive class are unknown to the author of this draft. It is out of scope of this draft to decide how 4 PHBs can be mapped to a single aggregated class. The suggested mapping is pragmatic and tries to come as close as possible to other design criteria pursued by GSMA IR.34.

### 5.3. Proposed MEF 23.1 to DiffServ Intercon mapping

MEF 23.1 is an implementation agreement facilitating Ethernet service interoperability and consistency between Operators and the use of a common CoS Label set for Subscribers [MEF23.1]. It pursues the same aims and method on Ethernet layer as this draft does on IP layer (i.e. providing an interconnection class and codepoint scheme). MEF 23.1 addresses external network to network interfaces typically interconnecting metro ethernet providers. This may typically involve

Ethernet Network Sections associated with typical Operator domains that interconnect at external network to network interfaces. MEF 23.1 specifies three aggregated CoS classes. In addition, the usage of a subset of Ethernet PCP and IP DSCP values is specified thus facilitating synergies between Ethernet and IP services and networks. The main purpose is specifying operator virtual (Ethernet) connections. As an IP QoS model is added, this draft proposes an IP class and codepoint mapping.

MEF 23.1 QoS mapping requires some thought as 3 classes are supported of which two are Ordered Aggregates. MEF 23.1s section 8.5.1 example on IP DSCP mapping may suggest supporting classification based on the DSCP Precedence Prefix. MEF 23.1 section 8.5.2.1 allows the conclusion that MEF class M is best mapped to this drafts Bulk inelastic class. The suggested mapping MEF to DiffServ Intercon is limited to the the MEF color blind mode (3 classes, no sub-classes):

MEF 23.1		DiffServ-Intercon	
Agg. Class	PHB	Agg. Class	PHB
High	EF	Priority	EF
Medium	AF3	Bulk inelastic	AF41
Low	CS1	Default	CS0

Suggested mapping of MEF 23.1 color blind mode classes and PHBs to DiffServ Intercon

Figure 3

The MEF color aware mode supports Ordered Aggregates in the MEF classes M and L. The MEF L specification classifies AF1 and Best Effort for the same Ordered Aggregate. A Better than Best Effort service produced in the same queue as best effort traffic can be realized, but hasn't been standardized by IETF. This document doesn't suggest any mapping. Diffserv Intercon also doesn't suggest an Ordered Aggregate in the Bulk Inelastic class. Later versions of this document may do so and specify a solution. Both issues are left for bilateral negotiation.

## 6. Contributors

David Black provided many helpful comments and inputs to this work.

## 7. Acknowledgements

Al Morton and Sebastien Jobert provided feedback on many aspects during private discussions. Brian Carpenter, Mohamed Boucadair and Thomas Knoll helped adding awareness of related work.

## 8. IANA Considerations

This memo includes no request to IANA.

## 9. Security Considerations

This document does not introduce new features, it describes how to use existing ones. The security section of RFC 2475 [RFC2475] and RFC 4594 [RFC4594] apply.

## 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.
- [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.
- [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., 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.

- [RFC3260] Grossman, D., "New Terminology and Clarifications for Diffserv", RFC 3260, April 2002.
- [RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, May 2002.
- [RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, January 2008.
- [RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, February 2009.
- [min\_ref] authSurName, authInitials., "Minimal Reference", 2006.

## 10.2. Informative References

- [I-D.knoll-idr-cos-interconnect]  
Knoll, T., "BGP Class of Service Interconnection",  
draft-knoll-idr-cos-interconnect-11 (work in progress),  
November 2013.
- [ID.idr-sla]  
IETF, "Inter-domain SLA Exchange", IETF, [http://  
datatracker.ietf.org/doc/draft-ietf-idr-sla-exchange/](http://datatracker.ietf.org/doc/draft-ietf-idr-sla-exchange/),  
2013.
- [IEEE802.1Q]  
IEEE, "IEEE Standard for Local and Metropolitan Area  
Networks - Virtual Bridged Local Area Networks", 2005.
- [IR.34] GSMA Association, "IR.34 Inter-Service Provider IP  
Backbone Guidelines Version 7.0", GSMA, GSMA IR.34 [http://  
www.gsma.com/newsroom/wp-content/uploads/2012/03/  
ir.34.pdf](http://www.gsma.com/newsroom/wp-content/uploads/2012/03/ir.34.pdf), 2012.
- [MEF23.1] MEF, "Implementation Agreement MEF 23.1 Carrier Ethernet  
Class of Service Phase 2", MEF, MEF23.1 [http://  
metroethernetforum.org/PDF\\_Documents/  
technical-specifications/MEF\\_23.1.pdf](http://metroethernetforum.org/PDF_Documents/technical-specifications/MEF_23.1.pdf), 2012.
- [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.

- [RFC5160] Levis, P. and M. Boucadair, "Considerations of Provider-to-Provider Agreements for Internet-Scale Quality of Service (QoS)", RFC 5160, March 2008.
- [Y.1566] ITU-T, "Quality of service mapping and interconnection between Ethernet, IP and multiprotocol label switching networks", ITU, <http://www.itu.int/rec/T-REC-Y.1566-201207-I/en>, 2012.

#### Appendix A. Change log

- 00 to 01 Added terminology and references. Added details and information to interconnection class and codepoint scheme. Editorial changes.
- 01 to 02 Added some references regarding related work. Clarified class definitions. Further editorial improvements.
- 02 to 03 Consistent terminology. Discussion of Network Management PHB at interconnection interfaces. Editorial review.
- 03 to 04 Again improved terminology. Better wording of Network Control PHB at interconnection interfaces.

#### Author's Address

Ruediger Geib (editor)  
Deutsche Telekom  
Heinrich Hertz Str. 3-7  
Darmstadt, 64295  
Germany

Phone: +49 6151 5812747  
Email: [Ruediger.Geib@telekom.de](mailto:Ruediger.Geib@telekom.de)

TSVWG  
Internet-Draft  
Updates: 4787, 5382, 5508 (if approved)  
Intended status: Best Current Practice  
Expires: September 3, 2016

R. Penno  
Cisco  
S. Perreault  
Jive Communications  
M. Boucadair, Ed.  
Orange  
S. Sivakumar  
Cisco  
K. Naito  
NTT  
March 2, 2016

Network Address Translation (NAT) Behavioral Requirements Updates  
draft-ietf-tsvwg-behave-requirements-update-08

Abstract

This document clarifies and updates several requirements of RFC4787, RFC5382, and RFC5508 based on operational and development experience. The focus of this document is NAT44.

This document updates RFCs 4787, 5382, and 5508.

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 3, 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.

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 . . . . .	3
1.1. Scope . . . . .	3
1.2. Terminology . . . . .	3
2. TCP Session Tracking . . . . .	3
2.1. TCP Transitory Connection Idle-Timeout . . . . .	5
2.2. TCP RST . . . . .	5
3. Port Overlapping Behavior . . . . .	5
4. Address Pooling Paired (APP) . . . . .	6
5. Endpoint-Independent Mapping (EIM) Protocol Independence . .	7
6. Endpoint-Independent Filtering (EIF) Protocol Independence .	7
7. Endpoint-Independent Filtering (EIF) Mapping Refresh . . . .	7
7.1. Outbound Mapping Refresh and Error Packets . . . . .	8
8. Port Parity . . . . .	8
9. Port Randomization . . . . .	8
10. IP Identification (IP ID) . . . . .	9
11. ICMP Query Mappings Timeout . . . . .	9
12. Hairpinning Support for ICMP Packets . . . . .	9
13. IANA Considerations . . . . .	9
14. Security Considerations . . . . .	10
15. References . . . . .	11
15.1. Normative References . . . . .	11
15.2. Informative References . . . . .	11
Acknowledgements . . . . .	12
Contributors . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

[RFC4787], [RFC5382], and [RFC5508] contributed to enhance Network Address Translation (NAT) interoperability and conformance. Operational experience gained through widespread deployment and evolution of NAT indicates that some areas of the original documents need further clarification or updates. This document provides such clarifications and updates.

### 1.1. Scope

The goal of this document is to clarify and update the set of requirements listed in [RFC4787], [RFC5382], and [RFC5508]. The document focuses exclusively on NAT44.

The scope of this document has been set so that it does not create new requirements beyond those specified in the documents cited above.

Carrier-Grade NAT (CGN) related requirements are defined in [RFC6888].

### 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].

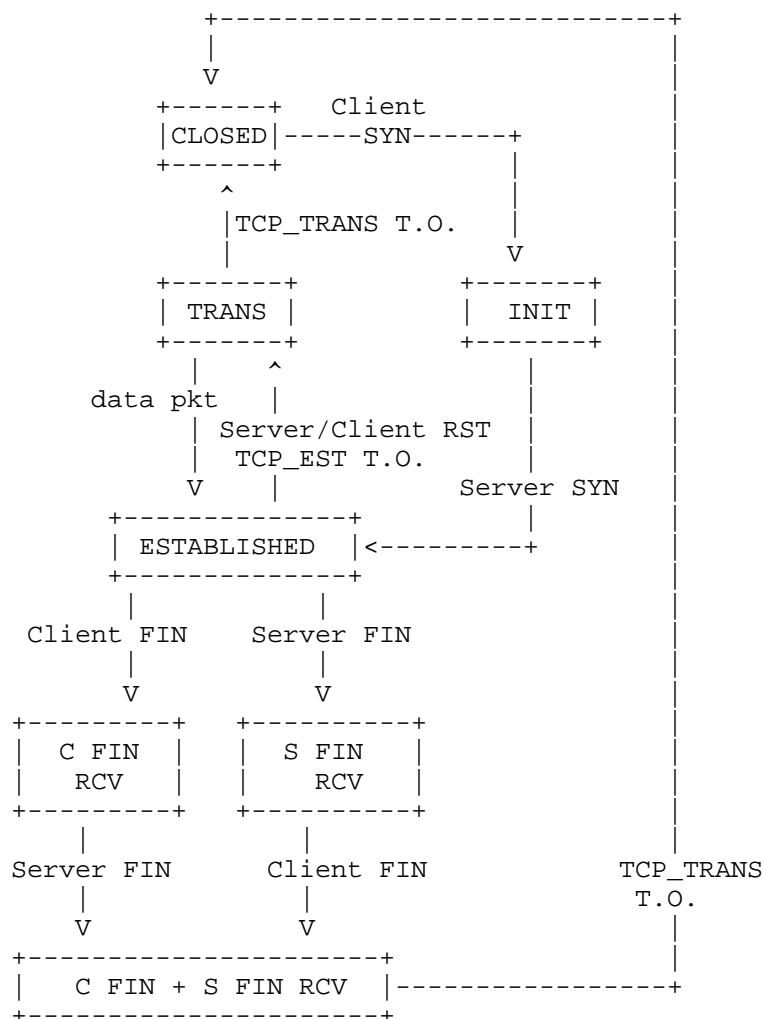
The reader is assumed to be familiar with the terminology defined in: [RFC2663],[RFC4787],[RFC5382], and [RFC5508].

In this document, the term "NAT" refers to both "Basic NAT" and "Network Address/Port Translator (NAPT)" (see Section 3 of [RFC4787]). As a reminder, Basic NAT and NAPT are two variations of traditional NAT, in that translation in Basic NAT is limited to IP addresses alone, whereas translation in NAPT is extended to include IP address and Transport identifier (such as TCP/UDP port or ICMP query ID) (refer to Section 2 of [RFC3022]).

## 2. TCP Session Tracking

[RFC5382] specifies TCP timers associated with various connection states but does not specify the TCP state machine a NAT44 should follow as a basis to apply such timers.

Update: The TCP state machine depicted in Figure 1, adapted from [RFC6146], SHOULD be implemented by a NAT for TCP session tracking purposes.

**Legend:**

- \* Messages sent or received from the server are prefixed with "Server".
- \* Messages sent or received from the client are prefixed with "Client".
- \* "C" means "Client-side"
- \* "S" means "Server-side".
- \* TCP\_EST T.O: refers to the established connection idle timeout as defined in [RFC5382].
- \* TCP\_TRANS T.O: refers to the transitory connection idle timeout as defined in [RFC5382].

Figure 1: Simplified version of the TCP State Machine

## 2.1. TCP Transitory Connection Idle-Timeout

The transitory connection idle-timeout is defined as the minimum time a TCP connection in the partially open or closing phases must remain idle before the NAT considers the associated session a candidate for removal (REQ-5 of [RFC5382]). But [RFC5382] does not clearly state whether these can be configured separately.

Clarification: This document clarifies that a NAT SHOULD provide different configurable parameters for configuring the open and closing idle timeouts.

To accommodate deployments that consider a partially open timeout of 4 minutes as being excessive from a security standpoint, a NAT MAY allow the configured timeout to be less than 4 minutes. However, a minimum default transitory connection idle-timeout of 4 minutes is RECOMMENDED.

## 2.2. TCP RST

[RFC5382] leaves the handling of TCP RST packets unspecified.

Update: This document adopts a similar default behavior as in [RFC6146]. Concretely, when the NAT receives a TCP RST matching an existing mapping, it MUST translate the packet according to the NAT mapping entry. Moreover, the NAT SHOULD wait for 4 minutes before deleting the session and removing any state associated with it if no packets are received during that 4 minutes timeout.

Notes:

- \* Admittedly, the NAT has to verify whether received TCP RST packets belong to a connection. This verification check is required to avoid off-path attacks.
- \* If the NAT removes immediately the NAT mapping upon receipt of a TCP RST message, stale connections may be maintained by endpoints if the first RST message is lost between the NAT and the recipient.

## 3. Port Overlapping Behavior

REQ-1 from [RFC4787] and REQ-1 from [RFC5382] specify a specific port overlapping behavior; that is the external IP address and port can be reused for connections originating from the same internal source IP address and port irrespective of the destination. This is known as endpoint-independent mapping (EIM).

Update: This document clarifies that this port overlapping behavior may be extended to connections originating from different internal source IP addresses and ports as long as their destinations are different.

The following mechanism MAY be implemented by a NAT:

If destination addresses and ports are different for outgoing connections started by local clients, a NAT MAY assign the same external port as the source ports for the connections. The port overlapping mechanism manages mappings between external packets and internal packets by looking at and storing their 5-tuple (protocol, source address, source port, destination address, destination port).

This enables concurrent use of a single NAT external port for multiple transport sessions, which allows a NAT to successfully process packets in an IP address resource limited network (e.g., deployment with high address space multiplicative factor (refer to Appendix B. of [RFC6269])).

#### 4. Address Pooling Paired (APP)

The "IP address pooling" behavior of "Paired" (APP) was recommended in REQ-2 from [RFC4787], but the behavior when an external IPv4 runs out of ports was left undefined.

Clarification: This document clarifies that if APP is enabled, new sessions from a host that already has a mapping associated with an external IP that ran out of ports SHOULD be dropped. A configuration parameter MAY be provided to allow a NAT to start using ports from another external IP address when the one that anchored the APP mapping ran out of ports. Tweaking this configuration parameter is a trade-off between service continuity and APP strict enforcement. Note, this behavior is sometimes referred to as 'soft-APP'.

As a reminder, the recommendation for the particular case of a CGN is that an implementation must use the same external IP address mapping for all sessions associated with the same internal IP address, be they TCP, UDP, ICMP, something else, or a mix of different protocols [RFC6888].

Update: This behavior SHOULD apply also for TCP.

## 5. Endpoint-Independent Mapping (EIM) Protocol Independence

REQ-1 from [RFC4787] and REQ-1 from [RFC5382] do not specify whether EIM are protocol-dependent or protocol-independent. For example, if an outbound TCP SYN creates a mapping, it is left undefined whether outbound UDP packets can reuse such mapping.

Update: EIM mappings SHOULD be protocol-dependent. A configuration parameter MAY be provided to allow protocols that multiplex TCP and UDP over the same source IP address and port number to use a single mapping. The default value of this configuration parameter MUST be protocol-dependent EIM.

This update is consistent with the stateful NAT64 [RFC6146] that clearly specifies three binding information bases (TCP, UDP, ICMP).

## 6. Endpoint-Independent Filtering (EIF) Protocol Independence

REQ-8 from [RFC4787] and REQ-3 from [RFC5382] do not specify whether mappings with endpoint-independent filtering (EIF) are protocol-independent or protocol-dependent. For example, if an outbound TCP SYN creates a mapping, it is left undefined whether inbound UDP packets matching that mapping should be accepted or rejected.

Update: EIF filtering SHOULD be protocol-dependent. A configuration parameter MAY be provided to make it protocol-independent. The default value of this configuration parameter MUST be protocol-dependent EIF.

This behavior is aligned with the update in Section 5.

Applications that can be transported over a variety of transport protocols and/or support transport fall back schemes won't experience connectivity failures if the NAT is configured with protocol-independent EIM and protocol-independent EIF.

## 7. Endpoint-Independent Filtering (EIF) Mapping Refresh

The NAT mapping Refresh direction may have a "NAT Inbound refresh behavior" of "True" according to REQ-6 from [RFC4787], but [RFC4787] does not clarify how this behavior applies to EIF mappings. The issue in question is whether inbound packets that match an EIF mapping but do not create a new session due to a security policy should refresh the mapping timer.

Clarification: This document clarifies that even when a NAT has an inbound refresh behavior set to 'TRUE', such packets SHOULD NOT

refresh the mapping. Otherwise a simple attack of a packet every 2 minutes can keep the mapping indefinitely.

Update: This behavior SHOULD apply also for TCP.

#### 7.1. Outbound Mapping Refresh and Error Packets

Update: In the case of NAT outbound refresh behavior, ICMP Errors or TCP RST outbound packets, sent as response to inbound packets, SHOULD NOT refresh the mapping. Other packets which indicate the host is not interested in receiving packets MAY be configurable to also not refresh state, such as STUN error response [RFC5389] or IKE INVALID\_SYNTAX [RFC7296].

#### 8. Port Parity

Update: A NAT MAY disable port parity preservation for all dynamic mappings. Nevertheless, A NAT SHOULD support means to explicitly request to preserve port parity (e.g., [RFC7753]).

Note: According to [RFC6887], dynamic mappings are said to be dynamic in the sense that they are created on demand, either implicitly or explicitly:

1. Implicit dynamic mappings refer to mappings that are created as a side effect of traffic such as an outgoing TCP SYN or outgoing UDP packet. Implicit dynamic mappings usually have a finite lifetime, though this lifetime is generally not known to the client using them.
2. Explicit dynamic mappings refer to mappings that are created as a result, for example, of explicit Port Control Protocol (PCP) MAP and PEER requests. Explicit dynamic mappings have a finite lifetime, and this lifetime is communicated to the client.

#### 9. Port Randomization

Update: A NAT SHOULD follow the recommendations specified in Section 4 of [RFC6056], especially:

"A NAT that does not implement port preservation [RFC4787] [RFC5382] SHOULD obfuscate selection of the ephemeral port of a packet when it is changed during translation of that packet. A NAT that does implement port preservation SHOULD obfuscate the ephemeral port of a packet only if the port must be changed as a result of the port being already in use for some other session. A NAT that performs parity preservation and that

must change the ephemeral port during translation of a packet SHOULD obfuscate the ephemeral ports. The algorithms described in this document could be easily adapted such that the parity is preserved (i.e., force the lowest order bit of the resulting port number to 0 or 1 according to whether even or odd parity is desired)."

#### 10. IP Identification (IP ID)

Update: A NAT SHOULD handle the Identification field of translated IPv4 packets as specified in Section 5.3.1 of [RFC6864].

#### 11. ICMP Query Mappings Timeout

Section 3.1 of [RFC5508] specifies that ICMP Query Mappings are to be maintained by a NAT. However, the specification doesn't discuss Query Mapping timeout values. Section 3.2 of [RFC5508] only discusses ICMP Query Session Timeouts.

Update: ICMP Query Mappings MAY be deleted once the last session using the mapping is deleted.

#### 12. Hairpinning Support for ICMP Packets

REQ-7 from [RFC5508] specifies that a NAT enforcing 'Basic NAT' must support traversal of hairpinned ICMP Query sessions.

Clarification: This implicitly means that address mappings from external address to internal address (similar to Endpoint Independent Filters) must be maintained to allow inbound ICMP Query sessions. If an ICMP Query is received on an external address, a NAT can then translate to an internal IP.

REQ-7 from [RFC5508] specifies that all NATs must support the traversal of hairpinned ICMP Error messages.

Clarification: This behavior requires a NAT to maintain address mappings from external IP address to internal IP address in addition to the ICMP Query Mappings described in Section 3.1 of [RFC5508].

#### 13. IANA Considerations

This document does not require any IANA action.



#### 14. Security Considerations

NAT behavioral considerations are discussed in [RFC4787], [RFC5382], and [RFC5508].

Because some of the clarifications and updates (e.g., Section 2) are inspired from NAT64, the security considerations discussed in Section 5 of [RFC6146] apply also for this specification.

The update in Section 3 allows for an optimized NAT resource usage. In order to avoid service disruption, the NAT must not invoke this functionality unless the packets are to be sent to distinct destination addresses.

Some of the updates (e.g., Section 7, Section 9, and Section 11) allow for an increased security compared to [RFC4787], [RFC5382], and [RFC5508]. Particularly:

- o The updates in Section 7 and Section 11 prevent an illegitimate node to maintain mappings activated in the NAT while these mappings should be cleared.
- o Port randomization (Section 9) complicates tracking hosts located behind a NAT.

Section 4 and Section 12 propose updates that increase the serviceability of a host located behind a NAT. These updates do not introduce any additional security concerns to [RFC4787], [RFC5382], and [RFC5508].

The updates in Section 5 and Section 6 allow for a better NAT transparency from an application standpoint. Hosts that require a restricted filtering behavior should enable specific policies (e.g., access control list (ACL)) either locally or by soliciting a dedicated security device (e.g., firewall). How a host updates its filtering policies is out of scope of this document.

The update in Section 8 induces security concerns that are specific to the protocol used to interact with the NAT. For example, if PCP is used to explicitly request parity preservation for a given mapping, the security considerations discussed in [RFC6887] should be taken into account.

The update in Section 10 may have undesired effects on the performance of the NAT in environments in which fragmentation is massively experienced. Such issue may be used as an attack vector against NATs.

## 15. References

### 15.1. 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [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, <<http://www.rfc-editor.org/info/rfc4787>>.
- [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, <<http://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, <<http://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, <<http://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, <<http://www.rfc-editor.org/info/rfc6146>>.
- [RFC6864] Touch, J., "Updated Specification of the IPv4 ID Field", RFC 6864, DOI 10.17487/RFC6864, February 2013, <<http://www.rfc-editor.org/info/rfc6864>>.

### 15.2. Informative References

- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <<http://www.rfc-editor.org/info/rfc2663>>.

- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<http://www.rfc-editor.org/info/rfc3022>>.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <<http://www.rfc-editor.org/info/rfc5389>>.
- [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, <<http://www.rfc-editor.org/info/rfc6269>>.
- [RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <<http://www.rfc-editor.org/info/rfc6887>>.
- [RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <<http://www.rfc-editor.org/info/rfc6888>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<http://www.rfc-editor.org/info/rfc7296>>.
- [RFC7753] Sun, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perreault, "Port Control Protocol (PCP) Extension for Port-Set Allocation", RFC 7753, DOI 10.17487/RFC7753, February 2016, <<http://www.rfc-editor.org/info/rfc7753>>.

#### Acknowledgements

Thanks to Dan Wing, Suresh Kumar, Mayuresh Bakshi, Rajesh Mohan, Lars Eggert, Gorrry Fairhurst, Brandon Williams, and David Black for their review and discussion.

Many thanks to Ben Laurie for the secdir review, and Dan Romascanu for the Gen-ART review.

Dan Wing proposed some text for the configurable errors in Section 7.1.

## Contributors

The following individual contributed text to the document:

Sarat Kamiset, Insieme Networks, United States

## Authors' Addresses

Reinaldo Penno  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, California 95134  
USA

Email: repenno@cisco.com

Simon Perreault  
Jive Communications  
Canada

Email: sperreault@jive.com

Mohamed Boucadair (editor)  
Orange  
Rennes 35000  
France

Email: mohamed.boucadair@orange.com

Senthil Sivakumar  
Cisco Systems, Inc.  
United States

Email: ssenthil@cisco.com

Kengo Naito  
NTT  
Tokyo  
Japan

Email: k.naito@nttv6.jp

Transport Area Working Group  
Internet-Draft  
Updates: 2309 (if approved)  
Intended status: BCP  
Expires: May 11, 2014

B. Briscoe  
BT  
J. Manner  
Aalto University  
November 07, 2013

Byte and Packet Congestion Notification  
draft-ietf-tsvwg-byte-pkt-congest-12

Abstract

This document provides recommendations of best current practice for dropping or marking packets using any active queue management (AQM) algorithm, including random early detection (RED), BLUE, pre-congestion notification (PCN) and newer schemes such as CoDel (Controlled Delay) and PIE (Proportional Integral controller Enhanced). We give three strong recommendations: (1) packet size should be taken into account when transports detect and respond to congestion indications, (2) packet size should not be taken into account when network equipment creates congestion signals (marking, dropping), and therefore (3) in the specific case of RED, the byte-mode packet drop variant that drops fewer small packets should not be used. This memo updates RFC 2309 to deprecate deliberate preferential treatment of small packets in AQM algorithms.

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 May 11, 2014.

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.	Terminology and Scoping . . . . .	6
1.2.	Example Comparing Packet-Mode Drop and Byte-Mode Drop . .	7
2.	Recommendations . . . . .	9
2.1.	Recommendation on Queue Measurement . . . . .	9
2.2.	Recommendation on Encoding Congestion Notification . . . .	10
2.3.	Recommendation on Responding to Congestion . . . . .	11
2.4.	Recommendation on Handling Congestion Indications when Splitting or Merging Packets . . . . .	12
3.	Motivating Arguments . . . . .	12
3.1.	Avoiding Perverse Incentives to (Ab)use Smaller Packets .	12
3.2.	Small != Control . . . . .	14
3.3.	Transport-Independent Network . . . . .	14
3.4.	Partial Deployment of AQM . . . . .	15
3.5.	Implementation Efficiency . . . . .	17
4.	A Survey and Critique of Past Advice . . . . .	17
4.1.	Congestion Measurement Advice . . . . .	18
4.1.1.	Fixed Size Packet Buffers . . . . .	18
4.1.2.	Congestion Measurement without a Queue . . . . .	19
4.2.	Congestion Notification Advice . . . . .	20
4.2.1.	Network Bias when Encoding . . . . .	20
4.2.2.	Transport Bias when Decoding . . . . .	22
4.2.3.	Making Transports Robust against Control Packet Losses . . . . .	23
4.2.4.	Congestion Notification: Summary of Conflicting Advice . . . . .	24
5.	Outstanding Issues and Next Steps . . . . .	25
5.1.	Bit-congestible Network . . . . .	25
5.2.	Bit- & Packet-congestible Network . . . . .	25
6.	Security Considerations . . . . .	26
7.	IANA Considerations . . . . .	26
8.	Conclusions . . . . .	26
9.	Acknowledgements . . . . .	28
10.	Comments Solicited . . . . .	28
11.	References . . . . .	28
11.1.	Normative References . . . . .	28
11.2.	Informative References . . . . .	28
	Appendix A. Survey of RED Implementation Status . . . . .	32
	Appendix B. Sufficiency of Packet-Mode Drop . . . . .	34
	B.1. Packet-Size (In)Dependence in Transports . . . . .	35
	B.2. Bit-Congestible and Packet-Congestible Indications . . . .	38
	Appendix C. Byte-mode Drop Complicates Policing Congestion Response . . . . .	39
	Appendix D. Changes from Previous Versions . . . . .	40

## 1. Introduction

This document provides recommendations of best current practice for how we should correctly scale congestion control functions with respect to packet size for the long term. It also recognises that expediency may be necessary to deal with existing widely deployed protocols that don't live up to the long term goal.

When signalling congestion, the problem of how (and whether) to take packet sizes into account has exercised the minds of researchers and practitioners for as long as active queue management (AQM) has been discussed. Indeed, one reason AQM was originally introduced was to reduce the lock-out effects that small packets can have on large packets in drop-tail queues. This memo aims to state the principles we should be using and to outline how these principles will affect future protocol design, taking into account the existing deployments we have already.

The question of whether to take into account packet size arises at three stages in the congestion notification process:

Measuring congestion: When a congested resource measures locally how congested it is, should it measure its queue length in time, bytes or packets?

Encoding congestion notification into the wire protocol: When a congested network resource signals its level of congestion, should it drop / mark each packet dependent on the size of the particular packet in question?

Decoding congestion notification from the wire protocol: When a transport interprets the notification in order to decide how much to respond to congestion, should it take into account the size of each missing or marked packet?

Consensus has emerged over the years concerning the first stage, which Section 2.1 records in the RFC Series. In summary: If possible it is best to measure congestion by time in the queue, but otherwise the choice between bytes and packets solely depends on whether the resource is congested by bytes or packets.

The controversy is mainly around the last two stages: whether to allow for the size of the specific packet notifying congestion i) when the network encodes or ii) when the transport decodes the congestion notification.

Currently, the RFC series is silent on this matter other than a paper trail of advice referenced from [RFC2309], which conditionally



recommends byte-mode (packet-size dependent) drop [pktByteEmail]. Reducing drop of small packets certainly has some tempting advantages: i) it drops less control packets, which tend to be small and ii) it makes TCP's bit-rate less dependent on packet size. However, there are ways of addressing these issues at the transport layer, rather than reverse engineering network forwarding to fix the problems.

This memo updates [RFC2309] to deprecate deliberate preferential treatment of packets in AQM algorithms solely because of their size. It recommends that (1) packet size should be taken into account when transports detect and respond to congestion indications, (2) not when network equipment creates them. This memo also adds to the congestion control principles enumerated in BCP 41 [RFC2914].

In the particular case of Random early Detection (RED), this means that the byte-mode packet drop variant should not be used to drop fewer small packets, because that creates a perverse incentive for transports to use tiny segments, consequently also opening up a DoS vulnerability. Fortunately all the RED implementers who responded to our admittedly limited survey (Section 4.2.4) have not followed the earlier advice to use byte-mode drop, so the position this memo argues for seems to already exist in implementations.

However, at the transport layer, TCP congestion control is a widely deployed protocol that doesn't scale with packet size (i.e. its reduction in rate does not take into account the size of a lost packet). To date this hasn't been a significant problem because most TCP implementations have been used with similar packet sizes. But, as we design new congestion control mechanisms, this memo recommends that we should build in scaling with packet size rather than assuming we should follow TCP's example.

This memo continues as follows. First it discusses terminology and scoping. Section 2 gives the concrete formal recommendations, followed by motivating arguments in Section 3. We then critically survey the advice given previously in the RFC series and the research literature (Section 4), referring to an assessment of whether or not this advice has been followed in production networks (Appendix A). To wrap up, outstanding issues are discussed that will need resolution both to inform future protocol designs and to handle legacy (Section 5). Then security issues are collected together in Section 6 before conclusions are drawn in Section 8. The interested reader can find discussion of more detailed issues on the theme of byte vs. packet in the appendices.

This memo intentionally includes a non-negligible amount of material on the subject. For the busy reader Section 2 summarises the

recommendations for the Internet community.

### 1.1. Terminology and Scoping

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].

This memo applies to the design of all AQM algorithms, for example, Random Early Detection (RED) [RFC2309], BLUE [BLUE02], Pre-Congestion Notification (PCN) [RFC5670], Controlled Delay (CoDel) [I-D.nichols-tsvwg-codel] and the Proportional Integral controller Enhanced (PIE) [I-D.pan-tsvwg-pie]. Throughout, RED is used as a concrete example because it is a widely known and deployed AQM algorithm. There is no intention to imply that the advice is any less applicable to the other algorithms, nor that RED is preferred.

**Congestion Notification:** Congestion notification is a changing signal that aims to communicate the probability that the network resource(s) will not be able to forward the level of traffic load offered (or that there is an impending risk that they will not be able to).

The 'impending risk' qualifier is added, because AQM systems set a virtual limit smaller than the actual limit to the resource, then notify when this virtual limit is exceeded in order to avoid uncontrolled congestion of the actual capacity.

Congestion notification communicates a real number bounded by the range [ 0 , 1 ]. This ties in with the most well-understood measure of congestion notification: drop probability.

**Explicit and Implicit Notification:** The byte vs. packet dilemma concerns congestion notification irrespective of whether it is signalled implicitly by drop or using Explicit Congestion Notification (ECN [RFC3168] or PCN [RFC5670]). Throughout this document, unless clear from the context, the term marking will be used to mean notifying congestion explicitly, while congestion notification will be used to mean notifying congestion either implicitly by drop or explicitly by marking.

**Bit-congestible vs. Packet-congestible:** If the load on a resource depends on the rate at which packets arrive, it is called packet-congestible. If the load depends on the rate at which bits arrive it is called bit-congestible.

Examples of packet-congestible resources are route look-up engines and firewalls, because load depends on how many packet headers

they have to process. Examples of bit-congestible resources are transmission links, radio power and most buffer memory, because the load depends on how many bits they have to transmit or store. Some machine architectures use fixed size packet buffers, so buffer memory in these cases is packet-congestible (see Section 4.1.1).

The path through a machine will typically encounter both packet-congestible and bit-congestible resources. However, currently, a design goal of network processing equipment such as routers and firewalls is to size the packet-processing engine(s) relative to the lines in order to keep packet processing uncongested even under worst case packet rates with runs of minimum size packets. Therefore, packet-congestion is currently rare [RFC6077; S.3.3], but there is no guarantee that it will not become more common in future.

Note that information is generally processed or transmitted with a minimum granularity greater than a bit (e.g. octets). The appropriate granularity for the resource in question should be used, but for the sake of brevity we will talk in terms of bytes in this memo.

Coarser Granularity: Resources may be congestible at higher levels of granularity than bits or packets, for instance stateful firewalls are flow-congestible and call-servers are session-congestible. This memo focuses on congestion of connectionless resources, but the same principles may be applicable for congestion notification protocols controlling per-flow and per-session processing or state.

RED Terminology: In RED whether to use packets or bytes when measuring queues is called respectively "packet-mode queue measurement" or "byte-mode queue measurement". And whether the probability of dropping a particular packet is independent or dependent on its size is called respectively "packet-mode drop" or "byte-mode drop". The terms byte-mode and packet-mode should not be used without specifying whether they apply to queue measurement or to drop.

## 1.2. Example Comparing Packet-Mode Drop and Byte-Mode Drop

Taking RED as a well-known example algorithm, a central question addressed by this document is whether to recommend RED's packet-mode drop variant and to deprecate byte-mode drop. Table 1 compares how packet-mode and byte-mode drop affect two flows of different size packets. For each it gives the expected number of packets and of bits dropped in one second. Each example flow runs at the same bit-

rate of 48Mb/s, but one is broken up into small 60 byte packets and the other into large 1500 byte packets.

To keep up the same bit-rate, in one second there are about 25 times more small packets because they are 25 times smaller. As can be seen from the table, the packet rate is 100,000 small packets versus 4,000 large packets per second (pps).

Parameter	Formula	Small packets	Large packets
Packet size	$s/8$	60B	1,500B
Packet size	$s$	480b	12,000b
Bit-rate	$x$	48Mbps	48Mbps
Packet-rate	$u = x/s$	100kpps	4kpps
Packet-mode Drop			
Pkt loss probability	$p$	0.1%	0.1%
Pkt loss-rate	$p*u$	100pps	4pps
Bit loss-rate	$p*u*s$	48kbps	48kbps
Byte-mode Drop			
	MTU, $M=12,000b$		
Pkt loss probability	$b = p*s/M$	0.004%	0.1%
Pkt loss-rate	$b*u$	4pps	4pps
Bit loss-rate	$b*u*s$	1.92kbps	48kbps

Table 1: Example Comparing Packet-mode and Byte-mode Drop

For packet-mode drop, we illustrate the effect of a drop probability of 0.1%, which the algorithm applies to all packets irrespective of size. Because there are 25 times more small packets in one second, it naturally drops 25 times more small packets, that is 100 small packets but only 4 large packets. But if we count how many bits it drops, there are 48,000 bits in 100 small packets and 48,000 bits in 4 large packets--the same number of bits of small packets as large.

The packet-mode drop algorithm drops any bit with the same probability whether the bit is in a small or a large packet.

For byte-mode drop, again we use an example drop probability of 0.1%, but only for maximum size packets (assuming the link maximum transmission unit (MTU) is 1,500B or 12,000b). The byte-mode algorithm reduces the drop probability of smaller packets proportional to their size, making the probability that it drops a small packet 25 times smaller at 0.004%. But there are 25 times more small packets, so dropping them with 25 times lower probability results in dropping the same number of packets: 4 drops in both cases. The 4 small dropped packets contain 25 times less bits than the 4 large dropped packets: 1,920 compared to 48,000.

The byte-mode drop algorithm drops any bit with a probability proportionate to the size of the packet it is in.

## 2. Recommendations

This section gives recommendations related to network equipment in Sections 2.1 and 2.2, and in Sections 2.3 and 2.4 we discuss the implications on the transport protocols.

### 2.1. Recommendation on Queue Measurement

Ideally, an AQM would measure the service time of the queue to measure congestion of a resource. However service time can only be measured as packets leave the queue, where it is not always expedient to implement a full AQM algorithm. To predict the service time as packets join the queue, an AQM algorithm needs to measure the length of the queue.

In this case, if the resource is bit-congestible, the AQM implementation SHOULD measure the length of the queue in bytes and, if the resource is packet-congestible, the implementation SHOULD measure the length of the queue in packets. Subject to the exceptions below, no other choice makes sense, because the number of packets waiting in the queue isn't relevant if the resource gets congested by bytes and vice versa. For example, the length of the queue into a transmission line would be measured in bytes, while the length of the queue into a firewall would be measured in packets.

To avoid the pathological effects of drop tail, the AQM can then transform this service time or queue length into the probability of dropping or marking a packet (e.g. RED's piecewise linear function between thresholds).

What this advice means for RED as a specific example:

1. A RED implementation SHOULD use byte mode queue measurement for measuring the congestion of bit-congestible resources and packet mode queue measurement for packet-congestible resources.
2. An implementation SHOULD NOT make it possible to configure the way a queue measures itself, because whether a queue is bit-congestible or packet-congestible is an inherent property of the queue.

Exceptions to these recommendations might be necessary, for instance where a packet-congestible resource has to be configured as a proxy bottleneck for a bit-congestible resource in an adjacent box that does not support AQM.

The recommended approach in less straightforward scenarios, such as fixed size packet buffers, resources without a queue and buffers comprising a mix of packet and bit-congestible resources, is discussed in Section 4.1. For instance, Section 4.1.1 explains that the queue into a line should be measured in bytes even if the queue consists of fixed-size packet-buffers, because the root-cause of any congestion is bytes arriving too fast for the line--packets filling buffers are merely a symptom of the underlying congestion of the line.

## 2.2. Recommendation on Encoding Congestion Notification

When encoding congestion notification (e.g. by drop, ECN or PCN), the probability that network equipment drops or marks a particular packet to notify congestion SHOULD NOT depend on the size of the packet in question. As the example in Section 1.2 illustrates, to drop any bit with probability 0.1% it is only necessary to drop every packet with probability 0.1% without regard to the size of each packet.

This approach ensures the network layer offers sufficient congestion information for all known and future transport protocols and also ensures no perverse incentives are created that would encourage transports to use inappropriately small packet sizes.

What this advice means for RED as a specific example:

1. The RED AQM algorithm SHOULD NOT use byte-mode drop, i.e. it ought to use packet-mode drop. Byte-mode drop is more complex, it creates the perverse incentive to fragment segments into tiny pieces and it is vulnerable to floods of small packets.
2. If a vendor has implemented byte-mode drop, and an operator has turned it on, it is RECOMMENDED to switch it to packet-mode drop, after establishing if there are any implications on the relative performance of applications using different packet sizes. The unlikely possibility of some application-specific legacy use of byte-mode drop is the only reason that all the above recommendations on encoding congestion notification are not phrased more strongly.

RED as a whole SHOULD NOT be switched off. Without RED, a drop tail queue biases against large packets and is vulnerable to floods of small packets.

Note well that RED's byte-mode queue drop is completely orthogonal to byte-mode queue measurement and should not be confused with it. If a RED implementation has a byte-mode but does not specify what sort of byte-mode, it is most probably byte-mode queue measurement, which is

fine. However, if in doubt, the vendor should be consulted.

A survey (Appendix A) showed that there appears to be little, if any, installed base of the byte-mode drop variant of RED. This suggests that deprecating byte-mode drop will have little, if any, incremental deployment impact.

### 2.3. Recommendation on Responding to Congestion

When a transport detects that a packet has been lost or congestion marked, it SHOULD consider the strength of the congestion indication as proportionate to the size in octets (bytes) of the missing or marked packet.

In other words, when a packet indicates congestion (by being lost or marked) it can be considered conceptually as if there is a congestion indication on every octet of the packet, not just one indication per packet.

To be clear, the above recommendation solely describes how a transport should interpret the meaning of a congestion indication, as a long term goal. It makes no recommendation on whether a transport should act differently based on this interpretation. It merely aids interoperability between transports, if they choose to make their actions depend on the strength of congestion indications.

This definition will be useful as the IETF transport area continues its programme of;

- o updating host-based congestion control protocols to take account of packet size
- o making transports less sensitive to losing control packets like SYNs and pure ACKs.

What this advice means for the case of TCP:

1. If two TCP flows with different packet sizes are required to run at equal bit rates under the same path conditions, this SHOULD be done by altering TCP (Section 4.2.2), not network equipment (the latter affects other transports besides TCP).
2. If it is desired to improve TCP performance by reducing the chance that a SYN or a pure ACK will be dropped, this SHOULD be done by modifying TCP (Section 4.2.3), not network equipment.

To be clear, we are not recommending at all that TCPs under equivalent conditions should aim for equal bit-rates. We are merely

saying that anyone trying to do such a thing should modify their TCP algorithm, not the network.

These recommendations are phrased as 'SHOULD' rather than 'MUST', because there may be cases where expediency dictates that compatibility with pre-existing versions of a transport protocol make the recommendations impractical.

#### 2.4. Recommendation on Handling Congestion Indications when Splitting or Merging Packets

Packets carrying congestion indications may be split or merged in some circumstances (e.g. at a RTP/RTCP transcoder or during IP fragment reassembly). Splitting and merging only make sense in the context of ECN, not loss.

The general rule to follow is that the number of octets in packets with congestion indications SHOULD be equivalent before and after merging or splitting. This is based on the principle used above; that an indication of congestion on a packet can be considered as an indication of congestion on each octet of the packet.

The above rule is not phrased with the word "MUST" to allow the following exception. There are cases where pre-existing protocols were not designed to conserve congestion marked octets (e.g. IP fragment reassembly [RFC3168] or loss statistics in RTCP receiver reports [RFC3550] before ECN was added [RFC6679]). When any such protocol is updated, it SHOULD comply with the above rule to conserve marked octets. However, the rule may be relaxed if it would otherwise become too complex to interoperate with pre-existing implementations of the protocol.

One can think of a splitting or merging process as if all the incoming congestion-marked octets increment a counter and all the outgoing marked octets decrement the same counter. In order to ensure that congestion indications remain timely, even the smallest positive remainder in the conceptual counter should trigger the next outgoing packet to be marked (causing the counter to go negative).

### 3. Motivating Arguments

This section is informative. It justifies the recommendations given in the previous section.

#### 3.1. Avoiding Perverse Incentives to (Ab)use Smaller Packets

Increasingly, it is being recognised that a protocol design must take care not to cause unintended consequences by giving the parties in



the protocol exchange perverse incentives [Evol\_cc][RFC3426]. Given there are many good reasons why larger path maximum transmission units (PMTUs) would help solve a number of scaling issues, we do not want to create any bias against large packets that is greater than their true cost.

Imagine a scenario where the same bit rate of packets will contribute the same to bit-congestion of a link irrespective of whether it is sent as fewer larger packets or more smaller packets. A protocol design that caused larger packets to be more likely to be dropped than smaller ones would be dangerous in both the following cases:

Malicious transports: A queue that gives an advantage to small packets can be used to amplify the force of a flooding attack. By sending a flood of small packets, the attacker can get the queue to discard more traffic in large packets, allowing more attack traffic to get through to cause further damage. Such a queue allows attack traffic to have a disproportionately large effect on regular traffic without the attacker having to do much work.

Non-malicious transports: Even if an application designer is not actually malicious, if over time it is noticed that small packets tend to go faster, designers will act in their own interest and use smaller packets. Queues that give advantage to small packets create an evolutionary pressure for applications or transports to send at the same bit-rate but break their data stream down into tiny segments to reduce their drop rate. Encouraging a high volume of tiny packets might in turn unnecessarily overload a completely unrelated part of the system, perhaps more limited by header-processing than bandwidth.

Imagine two unresponsive flows arrive at a bit-congestible transmission link each with the same bit rate, say 1Mbps, but one consists of 1500B and the other 60B packets, which are 25x smaller. Consider a scenario where gentle RED [gentle\_RED] is used, along with the variant of RED we advise against, i.e. where the RED algorithm is configured to adjust the drop probability of packets in proportion to each packet's size (byte mode packet drop). In this case, RED aims to drop 25x more of the larger packets than the smaller ones. Thus, for example if RED drops 25% of the larger packets, it will aim to drop 1% of the smaller packets (but in practice it may drop more as congestion increases [RFC4828; Appx B.4]). Even though both flows arrive with the same bit rate, the bit rate the RED queue aims to pass to the line will be 750kbps for the flow of larger packets but 990kbps for the smaller packets (because of rate variations it will actually be a little less than this target).

Note that, although the byte-mode drop variant of RED amplifies small

packet attacks, drop-tail queues amplify small packet attacks even more (see Security Considerations in Section 6). Wherever possible neither should be used.

### 3.2. Small != Control

Dropping fewer control packets considerably improves performance. It is tempting to drop small packets with lower probability in order to improve performance, because many control packets tend to be smaller (TCP SYNs & ACKs, DNS queries & responses, SIP messages, HTTP GETs, etc). However, we must not give control packets preference purely by virtue of their smallness, otherwise it is too easy for any data source to get the same preferential treatment simply by sending data in smaller packets. Again we should not create perverse incentives to favour small packets rather than to favour control packets, which is what we intend.

Just because many control packets are small does not mean all small packets are control packets.

So, rather than fix these problems in the network, we argue that the transport should be made more robust against losses of control packets (see 'Making Transports Robust against Control Packet Losses' in Section 4.2.3).

### 3.3. Transport-Independent Network

TCP congestion control ensures that flows competing for the same resource each maintain the same number of segments in flight, irrespective of segment size. So under similar conditions, flows with different segment sizes will get different bit-rates.

To counter this effect it seems tempting not to follow our recommendation, and instead for the network to bias congestion notification by packet size in order to equalise the bit-rates of flows with different packet sizes. However, in order to do this, the queuing algorithm has to make assumptions about the transport, which become embedded in the network. Specifically:

- o The queuing algorithm has to assume how aggressively the transport will respond to congestion (see Section 4.2.4). If the network assumes the transport responds as aggressively as TCP NewReno, it will be wrong for Compound TCP and differently wrong for Cubic TCP, etc. To achieve equal bit-rates, each transport then has to guess what assumption the network made, and work out how to replace this assumed aggressiveness with its own aggressiveness.

- o Also, if the network biases congestion notification by packet size it has to assume a baseline packet size--all proposed algorithms use the local MTU (for example see the byte-mode loss probability formula in Table 1). Then if the non-Reno transports mentioned above are trying to reverse engineer what the network assumed, they also have to guess the MTU of the congested link.

Even though reducing the drop probability of small packets (e.g. RED's byte-mode drop) helps ensure TCP flows with different packet sizes will achieve similar bit rates, we argue this correction should be made to any future transport protocols based on TCP, not to the network in order to fix one transport, no matter how predominant it is. Effectively, favouring small packets is reverse engineering of network equipment around one particular transport protocol (TCP), contrary to the excellent advice in [RFC3426], which asks designers to question "Why are you proposing a solution at this layer of the protocol stack, rather than at another layer?"

In contrast, if the network never takes account of packet size, the transport can be certain it will never need to guess any assumptions the network has made. And the network passes two pieces of information to the transport that are sufficient in all cases: i) congestion notification on the packet and ii) the size of the packet. Both are available for the transport to combine (by taking account of packet size when responding to congestion) or not. Appendix B checks that these two pieces of information are sufficient for all relevant scenarios.

When the network does not take account of packet size, it allows transport protocols to choose whether to take account of packet size or not. However, if the network were to bias congestion notification by packet size, transport protocols would have no choice; those that did not take account of packet size themselves would unwittingly become dependent on packet size, and those that already took account of packet size would end up taking account of it twice.

### 3.4. Partial Deployment of AQM

In overview, the argument in this section runs as follows:

- o Because the network does not and cannot always drop packets in proportion to their size, it shouldn't be given the task of making drop signals depend on packet size at all.
- o Transports on the other hand don't always want to make their rate response proportional to the size of dropped packets, but if they want to, they always can.

The argument is similar to the end-to-end argument that says "Don't do X in the network if end-systems can do X by themselves, and they want to be able to choose whether to do X anyway." Actually the following argument is stronger; in addition it says "Don't give the network task X that could be done by the end-systems, if X is not deployed on all network nodes, and end-systems won't be able to tell whether their network is doing X, or whether they need to do X themselves." In this case, the X in question is "making the response to congestion depend on packet size".

We will now re-run this argument taking each step in more depth. The argument applies solely to drop, not to ECN marking.

A queue drops packets for either of two reasons: a) to signal to host congestion controls that they should reduce the load and b) because there is no buffer left to store the packets. Active queue management tries to use drops as a signal for hosts to slow down (case a) so that drop due to buffer exhaustion (case b) should not be necessary.

AQM is not universally deployed in every queue in the Internet; many cheap Ethernet bridges, software firewalls, NATs on consumer devices, etc implement simple tail-drop buffers. Even if AQM were universal, it has to be able to cope with buffer exhaustion (by switching to a behaviour like tail-drop), in order to cope with unresponsive or excessive transports. For these reasons networks will sometimes be dropping packets as a last resort (case b) rather than under AQM control (case a).

When buffers are exhausted (case b), they don't naturally drop packets in proportion to their size. The network can only reduce the probability of dropping smaller packets if it has enough space to store them somewhere while it waits for a larger packet that it can drop. If the buffer is exhausted, it does not have this choice. Admittedly tail-drop does naturally drop somewhat fewer small packets, but exactly how few depends more on the mix of sizes than the size of the packet in question. Nonetheless, in general, if we wanted networks to do size-dependent drop, we would need universal deployment of (packet-size dependent) AQM code, which is currently unrealistic.

A host transport cannot know whether any particular drop was a deliberate signal from an AQM or a sign of a queue shedding packets due to buffer exhaustion. Therefore, because the network cannot universally do size-dependent drop, it should not do it all.

Whereas universality is desirable in the network, diversity is desirable between different transport layer protocols - some, like

NewReno TCP [RFC5681], may not choose to make their rate response proportionate to the size of each dropped packet, while others will (e.g. TFRC-SP [RFC4828]).

### 3.5. Implementation Efficiency

Biasing against large packets typically requires an extra multiply and divide in the network (see the example byte-mode drop formula in Table 1). Allowing for packet size at the transport rather than in the network ensures that neither the network nor the transport needs to do a multiply operation--multiplication by packet size is effectively achieved as a repeated add when the transport adds to its count of marked bytes as each congestion event is fed to it. Also the work to do the biasing is spread over many hosts, rather than concentrated in just the congested network element. These aren't principled reasons in themselves, but they are a happy consequence of the other principled reasons.

## 4. A Survey and Critique of Past Advice

This section is informative, not normative.

The original 1993 paper on RED [RED93] proposed two options for the RED active queue management algorithm: packet mode and byte mode. Packet mode measured the queue length in packets and dropped (or marked) individual packets with a probability independent of their size. Byte mode measured the queue length in bytes and marked an individual packet with probability in proportion to its size (relative to the maximum packet size). In the paper's outline of further work, it was stated that no recommendation had been made on whether the queue size should be measured in bytes or packets, but noted that the difference could be significant.

When RED was recommended for general deployment in 1998 [RFC2309], the two modes were mentioned implying the choice between them was a question of performance, referring to a 1997 email [pktByteEmail] for advice on tuning. A later addendum to this email introduced the insight that there are in fact two orthogonal choices:

- o whether to measure queue length in bytes or packets (Section 4.1)
- o whether the drop probability of an individual packet should depend on its own size (Section 4.2).

The rest of this section is structured accordingly.

#### 4.1. Congestion Measurement Advice

The choice of which metric to use to measure queue length was left open in RFC2309. It is now well understood that queues for bit-congestible resources should be measured in bytes, and queues for packet-congestible resources should be measured in packets [pktByteEmail].

Congestion in some legacy bit-congestible buffers is only measured in packets not bytes. In such cases, the operator has to set the thresholds mindful of a typical mix of packets sizes. Any AQM algorithm on such a buffer will be oversensitive to high proportions of small packets, e.g. a DoS attack, and under-sensitive to high proportions of large packets. However, there is no need to make allowances for the possibility of such legacy in future protocol design. This is safe because any under-sensitivity during unusual traffic mixes cannot lead to congestion collapse given the buffer will eventually revert to tail drop, discarding proportionately more large packets.

##### 4.1.1. Fixed Size Packet Buffers

The question of whether to measure queues in bytes or packets seems to be well understood. However, measuring congestion is confusing when the resource is bit congestible but the queue into the resource is packet congestible. This section outlines the approach to take.

Some, mostly older, queuing hardware allocates fixed sized buffers in which to store each packet in the queue. This hardware forwards to the line in one of two ways:

- o With some hardware, any fixed sized buffers not completely filled by a packet are padded when transmitted to the wire. This case, should clearly be treated as packet-congestible, because both queuing and transmission are in fixed MTU-sized units. Therefore the queue length in packets is a good model of congestion of the link.
- o More commonly, hardware with fixed size packet buffers transmits packets to line without padding. This implies a hybrid forwarding system with transmission congestion dependent on the size of packets but queue congestion dependent on the number of packets, irrespective of their size.

Nonetheless, there would be no queue at all unless the line had become congested--the root-cause of any congestion is too many bytes arriving for the line. Therefore, the AQM should measure the queue length as the sum of all the packet sizes in bytes that

are queued up waiting to be serviced by the line, irrespective of whether each packet is held in a fixed size buffer.

In the (unlikely) first case where use of padding means the queue should be measured in packets, further confusion is likely because the fixed buffers are rarely all one size. Typically pools of different sized buffers are provided (Cisco uses the term 'buffer carving' for the process of dividing up memory into these pools [IOSArch]). Usually, if the pool of small buffers is exhausted, arriving small packets can borrow space in the pool of large buffers, but not vice versa. However, there is no need to consider all this complexity, because the root-cause of any congestion is still line overload--buffer consumption is only the symptom. Therefore, the length of the queue should be measured as the sum of the bytes in the queue that will be transmitted to line, including any padding. In the (unusual) case of transmission with padding this means the sum of the sizes of the small buffers queued plus the sum of the sizes of the large buffers queued.

We will return to borrowing of fixed sized buffers when we discuss biasing the drop/marketing probability of a specific packet because of its size in Section 4.2.1. But here we can repeat the simple rule for how to measure the length of queues of fixed buffers: no matter how complicated the buffering scheme is, ultimately a transmission line is nearly always bit-congestible so the number of bytes queued up waiting for the line measures how congested the line is, and it is rarely important to measure how congested the buffering system is.

#### 4.1.1.2. Congestion Measurement without a Queue

AQM algorithms are nearly always described assuming there is a queue for a congested resource and the algorithm can use the queue length to determine the probability that it will drop or mark each packet. But not all congested resources lead to queues. For instance, power limited resources are usually bit-congestible if energy is primarily required for transmission rather than header processing, but it is rare for a link protocol to build a queue as it approaches maximum power.

Nonetheless, AQM algorithms do not require a queue in order to work. For instance spectrum congestion can be modelled by signal quality using target bit-energy-to-noise-density ratio. And, to model radio power exhaustion, transmission power levels can be measured and compared to the maximum power available. [ECNFixedWireless] proposes a practical and theoretically sound way to combine congestion notification for different bit-congestible resources at different layers along an end to end path, whether wireless or wired, and whether with or without queues.

In wireless protocols that use request to send / clear to send (RTS / CTS) control, such as some variants of IEEE802.11, it is reasonable to base an AQM on the time spent waiting for transmission opportunities (TXOPs) even though wireless spectrum is usually regarded as congested by bits (for a given coding scheme). This is because requests for TXOPs queue up as the spectrum gets congested by all the bits being transferred. So the time that TXOPs are queued directly reflects bit congestion of the spectrum.

## 4.2. Congestion Notification Advice

### 4.2.1. Network Bias when Encoding

#### 4.2.1.1. Advice on Packet Size Bias in RED

The previously mentioned email [pktByteEmail] referred to by [RFC2309] advised that most scarce resources in the Internet were bit-congestible, which is still believed to be true (Section 1.1). But it went on to offer advice that is updated by this memo. It said that drop probability should depend on the size of the packet being considered for drop if the resource is bit-congestible, but not if it is packet-congestible. The argument continued that if packet drops were inflated by packet size (byte-mode dropping), "a flow's fraction of the packet drops is then a good indication of that flow's fraction of the link bandwidth in bits per second". This was consistent with a referenced policing mechanism being worked on at the time for detecting unusually high bandwidth flows, eventually published in 1999 [pBox]. However, the problem could and should have been solved by making the policing mechanism count the volume of bytes randomly dropped, not the number of packets.

A few months before RFC2309 was published, an addendum was added to the above archived email referenced from the RFC, in which the final paragraph seemed to partially retract what had previously been said. It clarified that the question of whether the probability of dropping/markings a packet should depend on its size was not related to whether the resource itself was bit congestible, but a completely orthogonal question. However the only example given had the queue measured in packets but packet drop depended on the size of the packet in question. No example was given the other way round.

In 2000, Cnodder et al [REDbyte] pointed out that there was an error in the part of the original 1993 RED algorithm that aimed to distribute drops uniformly, because it didn't correctly take into account the adjustment for packet size. They recommended an algorithm called RED\_4 to fix this. But they also recommended a further change, RED\_5, to adjust drop rate dependent on the square of relative packet size. This was indeed consistent with one implied



motivation behind RED's byte mode drop--that we should reverse engineer the network to improve the performance of dominant end-to-end congestion control mechanisms. This memo makes a different recommendations in Section 2.

By 2003, a further change had been made to the adjustment for packet size, this time in the RED algorithm of the ns2 simulator. Instead of taking each packet's size relative to a 'maximum packet size' it was taken relative to a 'mean packet size', intended to be a static value representative of the 'typical' packet size on the link. We have not been able to find a justification in the literature for this change, however Eddy and Allman conducted experiments [REDbias] that assessed how sensitive RED was to this parameter, amongst other things. However, this changed algorithm can often lead to drop probabilities of greater than 1 (which gives a hint that there is probably a mistake in the theory somewhere).

On 10-Nov-2004, this variant of byte-mode packet drop was made the default in the ns2 simulator. It seems unlikely that byte-mode drop has ever been implemented in production networks (Appendix A), therefore any conclusions based on ns2 simulations that use RED without disabling byte-mode drop are likely to behave very differently from RED in production networks.

#### 4.2.1.2. Packet Size Bias Regardless of AQM

The byte-mode drop variant of RED (or a similar variant of other AQM algorithms) is not the only possible bias towards small packets in queueing systems. We have already mentioned that tail-drop queues naturally tend to lock-out large packets once they are full.

But also queues with fixed sized buffers reduce the probability that small packets will be dropped if (and only if) they allow small packets to borrow buffers from the pools for larger packets (see Section 4.1.1). Borrowing effectively makes the maximum queue size for small packets greater than that for large packets, because more buffers can be used by small packets while less will fit large packets. Incidentally, the bias towards small packets from buffer borrowing is nothing like as large as that of RED's byte-mode drop.

Nonetheless, fixed-buffer memory with tail drop is still prone to lock-out large packets, purely because of the tail-drop aspect. So, fixed size packet-buffers should be augmented with a good AQM algorithm and packet-mode drop. If an AQM is too complicated to implement with multiple fixed buffer pools, the minimum necessary to prevent large packet lock-out is to ensure smaller packets never use the last available buffer in any of the pools for larger packets.

#### 4.2.2. Transport Bias when Decoding

The above proposals to alter the network equipment to bias towards smaller packets have largely carried on outside the IETF process. Whereas, within the IETF, there are many different proposals to alter transport protocols to achieve the same goals, i.e. either to make the flow bit-rate take account of packet size, or to protect control packets from loss. This memo argues that altering transport protocols is the more principled approach.

A recently approved experimental RFC adapts its transport layer protocol to take account of packet sizes relative to typical TCP packet sizes. This proposes a new small-packet variant of TCP-friendly rate control [RFC5348] called TFRC-SP [RFC4828]. Essentially, it proposes a rate equation that inflates the flow rate by the ratio of a typical TCP segment size (1500B including TCP header) over the actual segment size [PktSizeEquCC]. (There are also other important differences of detail relative to TFRC, such as using virtual packets [CCvarPktSize] to avoid responding to multiple losses per round trip and using a minimum inter-packet interval.)

Section 4.5.1 of this TFRC-SP spec discusses the implications of operating in an environment where queues have been configured to drop smaller packets with proportionately lower probability than larger ones. But it only discusses TCP operating in such an environment, only mentioning TFRC-SP briefly when discussing how to define fairness with TCP. And it only discusses the byte-mode dropping version of RED as it was before Cnoddler et al pointed out it didn't sufficiently bias towards small packets to make TCP independent of packet size.

So the TFRC-SP spec doesn't address the issue of which of the network or the transport should handle fairness between different packet sizes. In its Appendix B.4 it discusses the possibility of both TFRC-SP and some network buffers duplicating each other's attempts to deliberately bias towards small packets. But the discussion is not conclusive, instead reporting simulations of many of the possibilities in order to assess performance but not recommending any particular course of action.

The paper originally proposing TFRC with virtual packets (VP-TFRC) [CCvarPktSize] proposed that there should perhaps be two variants to cater for the different variants of RED. However, as the TFRC-SP authors point out, there is no way for a transport to know whether some queues on its path have deployed RED with byte-mode packet drop (except if an exhaustive survey found that no-one has deployed it!--see Appendix A). Incidentally, VP-TFRC also proposed that byte-mode RED dropping should really square the packet-size compensation-factor

(like that of Cnoder's RED\_5, but apparently unaware of it).

Pre-congestion notification [RFC5670] is an IETF technology to use a virtual queue for AQM marking for packets within one Diffserv class in order to give early warning prior to any real queuing. The PCN marking algorithms have been designed not to take account of packet size when forwarding through queues. Instead the general principle has been to take account of the sizes of marked packets when monitoring the fraction of marking at the edge of the network, as recommended here.

#### 4.2.3. Making Transports Robust against Control Packet Losses

Recently, two RFCs have defined changes to TCP that make it more robust against losing small control packets [RFC5562] [RFC5690]. In both cases they note that the case for these two TCP changes would be weaker if RED were biased against dropping small packets. We argue here that these two proposals are a safer and more principled way to achieve TCP performance improvements than reverse engineering RED to benefit TCP.

Although there are no known proposals, it would also be possible and perfectly valid to make control packets robust against drop by requesting a scheduling class with lower drop probability, by re-marking to a Diffserv code point [RFC2474] within the same behaviour aggregate.

Although not brought to the IETF, a simple proposal from Wischik [DupTCP] suggests that the first three packets of every TCP flow should be routinely duplicated after a short delay. It shows that this would greatly improve the chances of short flows completing quickly, but it would hardly increase traffic levels on the Internet, because Internet bytes have always been concentrated in the large flows. It further shows that the performance of many typical applications depends on completion of long serial chains of short messages. It argues that, given most of the value people get from the Internet is concentrated within short flows, this simple expedient would greatly increase the value of the best efforts Internet at minimal cost. A similar but more extensive approach has been evaluated on Google servers [GentleAggro].

The proposals discussed in this sub-section are experimental approaches that are not yet in wide operational use, but they are existence proofs that transports can make themselves robust against loss of control packets. The examples are all TCP-based, but applications over non-TCP transports could mitigate loss of control packets by making similar use of Diffserv, data duplication, FEC etc.

## 4.2.4. Congestion Notification: Summary of Conflicting Advice

transport cc	RED_1 (packet mode drop)	RED_4 (linear byte mode drop)	RED_5 (square byte mode drop)
TCP or TFRC	$s/\sqrt{p}$	$\sqrt{s/p}$	$1/\sqrt{p}$
TFRC-SP	$1/\sqrt{p}$	$1/\sqrt{sp}$	$1/(s.\sqrt{p})$

Table 2: Dependence of flow bit-rate per RTT on packet size,  $s$ , and drop probability,  $p$ , when network and/or transport bias towards small packets to varying degrees

Table 2 aims to summarise the potential effects of all the advice from different sources. Each column shows a different possible AQM behaviour in different queues in the network, using the terminology of Cnoder et al outlined earlier (RED\_1 is basic RED with packet-mode drop). Each row shows a different transport behaviour: TCP [RFC5681] and TFRC [RFC5348] on the top row with TFRC-SP [RFC4828] below. Each cell shows how the bits per round trip of a flow depends on packet size,  $s$ , and drop probability,  $p$ . In order to declutter the formulae to focus on packet-size dependence they are all given per round trip, which removes any RTT term.

Let us assume that the goal is for the bit-rate of a flow to be independent of packet size. Suppressing all inessential details, the table shows that this should either be achievable by not altering the TCP transport in a RED\_5 network, or using the small packet TFRC-SP transport (or similar) in a network without any byte-mode dropping RED (top right and bottom left). Top left is the 'do nothing' scenario, while bottom right is the 'do-both' scenario in which bit-rate would become far too biased towards small packets. Of course, if any form of byte-mode dropping RED has been deployed on a subset of queues that congest, each path through the network will present a different hybrid scenario to its transport.

Whatever, we can see that the linear byte-mode drop column in the middle would considerably complicate the Internet. It's a half-way house that doesn't bias enough towards small packets even if one believes the network should be doing the biasing. Section 2 recommends that all bias in network equipment towards small packets should be turned off--if indeed any equipment vendors have implemented it--leaving packet-size bias solely as the preserve of the transport layer (solely the leftmost, packet-mode drop column).

In practice it seems that no deliberate bias towards small packets

has been implemented for production networks. Of the 19% of vendors who responded to a survey of 84 equipment vendors, none had implemented byte-mode drop in RED (see Appendix A for details).

## 5. Outstanding Issues and Next Steps

### 5.1. Bit-congestible Network

For a connectionless network with nearly all resources being bit-congestible the recommended position is clear--that the network should not make allowance for packet sizes and the transport should. This leaves two outstanding issues:

- o How to handle any legacy of AQM with byte-mode drop already deployed;
- o The need to start a programme to update transport congestion control protocol standards to take account of packet size.

A survey of equipment vendors (Section 4.2.4) found no evidence that byte-mode packet drop had been implemented, so deployment will be sparse at best. A migration strategy is not really needed to remove an algorithm that may not even be deployed.

A programme of experimental updates to take account of packet size in transport congestion control protocols has already started with TFRC-SP [RFC4828].

### 5.2. Bit- & Packet-congestible Network

The position is much less clear-cut if the Internet becomes populated by a more even mix of both packet-congestible and bit-congestible resources (see Appendix B.2). This problem is not pressing, because most Internet resources are designed to be bit-congestible before packet processing starts to congest (see Section 1.1).

The IRTF Internet congestion control research group (ICCRG) has set itself the task of reaching consensus on generic forwarding mechanisms that are necessary and sufficient to support the Internet's future congestion control requirements (the first challenge in [RFC6077]). The research question of whether packet congestion might become common and what to do if it does may in the future be explored in the IRTF (the "Challenge 3: Packet Size" in [RFC6077]).

Note that sometimes it seems that resources might be congested by neither bits nor packets, e.g. where the queue for access to a wireless medium is in units of transmission opportunities. However,

the root cause of congestion of the underlying spectrum is overload of bits (see Section 4.1.2).

## 6. Security Considerations

This memo recommends that queues do not bias drop probability due to packets size. For instance dropping small packets less often than large creates a perverse incentive for transports to break down their flows into tiny segments. One of the benefits of implementing AQM was meant to be to remove this perverse incentive that drop-tail queues gave to small packets.

In practice, transports cannot all be trusted to respond to congestion. So another reason for recommending that queues do not bias drop probability towards small packets is to avoid the vulnerability to small packet DDoS attacks that would otherwise result. One of the benefits of implementing AQM was meant to be to remove drop-tail's DoS vulnerability to small packets, so we shouldn't add it back again.

If most queues implemented AQM with byte-mode drop, the resulting network would amplify the potency of a small packet DDoS attack. At the first queue the stream of packets would push aside a greater proportion of large packets, so more of the small packets would survive to attack the next queue. Thus a flood of small packets would continue on towards the destination, pushing regular traffic with large packets out of the way in one queue after the next, but suffering much less drop itself.

Appendix C explains why the ability of networks to police the response of \_any\_ transport to congestion depends on bit-congestible network resources only doing packet-mode not byte-mode drop. In summary, it says that making drop probability depend on the size of the packets that bits happen to be divided into simply encourages the bits to be divided into smaller packets. Byte-mode drop would therefore irreversibly complicate any attempt to fix the Internet's incentive structures.

## 7. IANA Considerations

This document has no actions for IANA.

## 8. Conclusions

This memo identifies the three distinct stages of the congestion notification process where implementations need to decide whether to take packet size into account. The recommendations provided in Section 2 of this memo are different in each case:

- o When network equipment measures the length of a queue, if it is not feasible to use time it is recommended to count in bytes if the network resource is congested by bytes, or to count in packets if is congested by packets.
- o When network equipment decides whether to drop (or mark) a packet, it is recommended that the size of the particular packet should not be taken into account
- o However, when a transport algorithm responds to a dropped or marked packet, the size of the rate reduction should be proportionate to the size of the packet.

In summary, the answers are 'it depends', 'no' and 'yes' respectively

For the specific case of RED, this means that byte-mode queue measurement will often be appropriate but the use of byte-mode drop is very strongly discouraged.

At the transport layer the IETF should continue updating congestion control protocols to take account of the size of each packet that indicates congestion. Also the IETF should continue to make protocols less sensitive to losing control packets like SYN's, pure ACKs and DNS exchanges. Although many control packets happen to be small, the alternative of network equipment favouring all small packets would be dangerous. That would create perverse incentives to split data transfers into smaller packets.

The memo develops these recommendations from principled arguments concerning scaling, layering, incentives, inherent efficiency, security and policeability. But it also addresses practical issues such as specific buffer architectures and incremental deployment. Indeed a limited survey of RED implementations is discussed, which shows there appears to be little, if any, installed base of RED's byte-mode drop. Therefore it can be deprecated with little, if any, incremental deployment complications.

The recommendations have been developed on the well-founded basis that most Internet resources are bit-congestible not packet-congestible. We need to know the likelihood that this assumption will prevail longer term and, if it might not, what protocol changes will be needed to cater for a mix of the two. The IRTF Internet Congestion Control Research Group (ICCRG) is currently working on these problems [RFC6077].

## 9. Acknowledgements

Thank you to Sally Floyd, who gave extensive and useful review comments. Also thanks for the reviews from Philip Eardley, David Black, Fred Baker, David Taht, Toby Moncaster, Arnaud Jacquet and Mirja Kuehlewind as well as helpful explanations of different hardware approaches from Larry Dunn and Fred Baker. We are grateful to Bruce Davie and his colleagues for providing a timely and efficient survey of RED implementation in Cisco's product range. Also grateful thanks to Toby Moncaster, Will Dormann, John Regnault, Simon Carter and Stefaan De Cnodder who further helped survey the current status of RED implementation and deployment and, finally, thanks to the anonymous individuals who responded.

Bob Briscoe and Jukka Manner were partly funded by Trilogy, a research project (ICT- 216372) supported by the European Community under its Seventh Framework Programme. The views expressed here are those of the authors only.

## 10. 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.

## 11. References

### 11.1. 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.

### 11.2. Informative References

- [BLUE02] Feng, W-c., Shin, K., Kandlur, D., and D. Saha, "The BLUE active queue management algorithms", IEEE/ACM Transactions on Networking 10(4) 513--528, August 2002, <<http://dx.doi.org/10.1109/TNET.2002.801399>>.
- [CCvarPktSize] Widmer, J., Boutremans, C., and J-Y. Le



- Boudec, "Congestion Control for Flows with Variable Packet Size", ACM CCR 34(2) 137--151, 2004, <<http://doi.acm.org/10.1145/997150.997162>>.
- [CHOke\_Var\_Pkt] Psounis, K., Pan, R., and B. Prabhaker, "Approximate Fair Dropping for Variable Length Packets", IEEE Micro 21(1):48--56, January-February 2001, <<http://www.stanford.edu/~balaji/papers/01approximatefair.pdf>>.
- [DRQ] Shin, M., Chong, S., and I. Rhee, "Dual-Resource TCP/AQM for Processing-Constrained Networks", IEEE/ACM Transactions on Networking Vol 16, issue 2, April 2008, <<http://dx.doi.org/10.1109/TNET.2007.900415>>.
- [DupTCP] Wischik, D., "Short messages", Philosophical Transactions of the Royal Society A 366(1872):1941-1953, June 2008, <<http://rsta.royalsocietypublishing.org/content/366/1872/1941.full.pdf+html>>.
- [ECNFixedWireless] Siris, V., "Resource Control for Elastic Traffic in CDMA Networks", Proc. ACM MOBICOM'02 , September 2002, <[http://www.ics.forth.gr/netlab/publications/resource\\_control\\_elastic\\_cdma.html](http://www.ics.forth.gr/netlab/publications/resource_control_elastic_cdma.html)>.
- [Evol\_cc] Gibbens, R. and F. Kelly, "Resource pricing and the evolution of congestion control", Automatica 35(12):1969--1985, December 1999, <<http://www.statslab.cam.ac.uk/~frank/evol.html>>.
- [GentleAggro] Flach, T., Dukkupati, N., Terzis, A., Raghavan, B., Cardwell, N., Cheng, Y., Jain, A., Hao, S., Katz-Bassett, E., and R. Govindan, "Reducing Web Latency: the Virtue of Gentle Aggression", ACM SIGCOMM CCR 43(4):159--170, August 2013, <<http://doi.acm.org/10.1145/2486001.2486014>>.
- [I-D.nichols-tsvwg-codel] Nichols, K. and V. Jacobson, "Controlled Delay Active Queue Management",

- draft-nichols-tsvwg-codel-01 (work in progress), February 2013.
- [I-D.pan-tsvwg-pie]    Pan, R., Natarajan, P., Piglione, C., and M. Prabhu, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", draft-pan-tsvwg-pie-00 (work in progress), December 2012.
- [IOSArch]    Bollapragada, V., White, R., and C. Murphy, "Inside Cisco IOS Software Architecture", Cisco Press: CCIE Professional Development ISBN13: 978-1-57870-181-0, July 2000.
- [PktSizeEquCC]    Vasallo, P., "Variable Packet Size Equation-Based Congestion Control", ICSI Technical Report tr-00-008, 2000, <<http://http.icsi.berkeley.edu/ftp/global/pub/techreports/2000/tr-00-008.pdf>>.
- [RED93]    Floyd, S. and V. Jacobson, "Random Early Detection (RED) gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking 1(4) 397--413, August 1993, <<http://www.icir.org/floyd/papers/red/red.html>>.
- [REDBias]    Eddy, W. and M. Allman, "A Comparison of RED's Byte and Packet Modes", Computer Networks 42(3) 261--280, June 2003, <<http://www.ir.bbn.com/documents/articles/redbias.ps>>.
- [REDbyte]    De Cnodder, S., Elloumi, O., and K. Pauwels, "RED behavior with different packet sizes", Proc. 5th IEEE Symposium on Computers and Communications (ISCC) 793--799, July 2000, <<http://www.icir.org/floyd/red/Elloumi99.pdf>>.
- [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.
- [RFC2914]                Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.
- [RFC3426]                Floyd, S., "General Architectural and Policy Considerations", RFC 3426, November 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.
- [RFC3714]                Floyd, S. and J. Kempf, "IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet", RFC 3714, March 2004.
- [RFC4828]                Floyd, S. and E. Kohler, "TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant", RFC 4828, April 2007.
- [RFC5348]                Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, September 2008.
- [RFC5562]                Kuzmanovic, A., Mondal, A., Floyd, S., and K. Ramakrishnan, "Adding Explicit Congestion Notification (ECN) Capability to TCP's SYN/ACK Packets", RFC 5562, June 2009.
- [RFC5670]                Eardley, P., "Metering and Marking Behaviour of PCN-Nodes", RFC 5670, November 2009.
- [RFC5681]                Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.

- [RFC5690]                    Floyd, S., Arcia, A., Ros, D., and J. Iyengar, "Adding Acknowledgement Congestion Control to TCP", RFC 5690, February 2010.
- [RFC6077]                    Papadimitriou, D., Welzl, M., Scharf, M., and B. Briscoe, "Open Research Issues in Internet Congestion Control", RFC 6077, February 2011.
- [RFC6679]                    Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, August 2012.
- [RFC6789]                    Briscoe, B., Woundy, R., and A. Cooper, "Congestion Exposure (ConEx) Concepts and Use Cases", RFC 6789, December 2012.
- [Rate\_fair\_Dis]              Briscoe, B., "Flow Rate Fairness: Dismantling a Religion", ACM CCR 37(2)63--74, April 2007, <<http://portal.acm.org/citation.cfm?id=1232926>>.
- [gentle\_RED]                Floyd, S., "Recommendation on using the "gentle\_" variant of RED", Web page , March 2000, <<http://www.icir.org/floyd/red/gentle.html>>.
- [pBox]                      Floyd, S. and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE/ACM Transactions on Networking 7(4) 458--472, August 1999, <<http://www.aciri.org/floyd/end2end-paper.html>>.
- [pktByteEmail]              Floyd, S., "RED: Discussions of Byte and Packet Modes", email , March 1997, <<http://www-nrg.ee.lbl.gov/floyd/REDaveraging.txt>>.

#### Appendix A.    Survey of RED Implementation Status

This Appendix is informative, not normative.

In May 2007 a survey was conducted of 84 vendors to assess how widely drop probability based on packet size has been implemented in RED Table 3. About 19% of those surveyed replied, giving a sample size

of 16. Although in most cases we do not have permission to identify the respondents, we can say that those that have responded include most of the larger equipment vendors, covering a large fraction of the market. The two who gave permission to be identified were Cisco and Alcatel-Lucent. The others range across the large network equipment vendors at L3 & L2, firewall vendors, wireless equipment vendors, as well as large software businesses with a small selection of networking products. All those who responded confirmed that they have not implemented the variant of RED with drop dependent on packet size (2 were fairly sure they had not but needed to check more thoroughly). At the time the survey was conducted, Linux did not implement RED with packet-size bias of drop, although we have not investigated a wider range of open source code.

Response	No. of vendors	%age of vendors
Not implemented	14	17%
Not implemented (probably)	2	2%
Implemented	0	0%
No response	68	81%
Total companies/orgs surveyed	84	100%

Table 3: Vendor Survey on byte-mode drop variant of RED (lower drop probability for small packets)

Where reasons have been given, the extra complexity of packet bias code has been most prevalent, though one vendor had a more principled reason for avoiding it--similar to the argument of this document.

Our survey was of vendor implementations, so we cannot be certain about operator deployment. But we believe many queues in the Internet are still tail-drop. The company of one of the co-authors (BT) has widely deployed RED, but many tail-drop queues are bound to still exist, particularly in access network equipment and on middleboxes like firewalls, where RED is not always available.

Routers using a memory architecture based on fixed size buffers with borrowing may also still be prevalent in the Internet. As explained in Section 4.2.1, these also provide a marginal (but legitimate) bias towards small packets. So even though RED byte-mode drop is not prevalent, it is likely there is still some bias towards small packets in the Internet due to tail drop and fixed buffer borrowing.

## Appendix B.    Sufficiency of Packet-Mode Drop

This Appendix is informative, not normative.

Here we check that packet-mode drop (or marking) in the network gives sufficiently generic information for the transport layer to use. We check against a 2x2 matrix of four scenarios that may occur now or in the future (Table 4). The horizontal and vertical dimensions have been chosen because each tests extremes of sensitivity to packet size in the transport and in the network respectively.

Note that this section does not consider byte-mode drop at all. Having deprecated byte-mode drop, the goal here is to check that packet-mode drop will be sufficient in all cases.

Network	Transport	a) Independent of packet size of congestion notifications	b) Dependent on packet size of congestion notifications
1) Predominantly bit-congestible network		Scenario a1)	Scenario b1)
2) Mix of bit-congestible and pkt-congestible network		Scenario a2)	Scenario b2)

Table 4: Four Possible Congestion Scenarios

Appendix B.1 focuses on the horizontal dimension of Table 4 checking that packet-mode drop (or marking) gives sufficient information, whether or not the transport uses it--scenarios b) and a) respectively.

Appendix B.2 focuses on the vertical dimension of Table 4, checking that packet-mode drop gives sufficient information to the transport whether resources in the network are bit-congestible or packet-congestible (these terms are defined in Section 1.1).

Notation: To be concrete, we will compare two flows with different packet sizes,  $s_1$  and  $s_2$ . As an example, we will take  $s_1 = 60B = 480b$  and  $s_2 = 1500B = 12,000b$ .

A flow's bit rate,  $x$  [bps], is related to its packet rate,  $u$  [pps], by

$$x(t) = s.u(t).$$

In the bit-congestible case, path congestion will be denoted by `p_b`, and in the packet-congestible case by `p_p`. When either case is implied, the letter `p` alone will denote path congestion.

#### B.1. Packet-Size (In)Dependence in Transports

In all cases we consider a packet-mode drop queue that indicates congestion by dropping (or marking) packets with probability `p` irrespective of packet size. We use an example value of loss (marking) probability, `p=0.1%`.

A transport like RFC5681 TCP treats a congestion notification on any packet whatever its size as one event. However, a network with just the packet-mode drop algorithm does give more information if the transport chooses to use it. We will use Table 5 to illustrate this.

We will set aside the last column until later. The columns labelled "Flow 1" and "Flow 2" compare two flows consisting of 60B and 1500B packets respectively. The body of the table considers two separate cases, one where the flows have equal bit-rate and the other with equal packet-rates. In both cases, the two flows fill a 96Mbps link. Therefore, in the equal bit-rate case they each have half the bit-rate (48Mbps). Whereas, with equal packet-rates, flow 1 uses 25 times smaller packets so it gets 25 times less bit-rate--it only gets  $1/(1+25)$  of the link capacity ( $96\text{Mbps}/26 = 4\text{Mbps}$  after rounding). In contrast flow 2 gets 25 times more bit-rate (92Mbps) in the equal packet rate case because its packets are 25 times larger. The packet rate shown for each flow could easily be derived once the bit-rate was known by dividing bit-rate by packet size, as shown in the column labelled "Formula".

Parameter	Formula	Flow 1	Flow 2	Combined
-----	-----	-----	-----	-----
Packet size	$s/8$	60B	1,500B	(Mix)
Packet size	$s$	480b	12,000b	(Mix)
Pkt loss probability	$p$	0.1%	0.1%	0.1%
EQUAL BIT-RATE CASE				
Bit-rate	$x$	48Mbps	48Mbps	96Mbps
Packet-rate	$u = x/s$	100kpps	4kpps	104kpps
Absolute pkt-loss-rate	$p*u$	100pps	4pps	104pps
Absolute bit-loss-rate	$p*u*s$	48kbps	48kbps	96kbps
Ratio of lost/sent pkts	$p*u/u$	0.1%	0.1%	0.1%
Ratio of lost/sent bits	$p*u*s/(u*s)$	0.1%	0.1%	0.1%
EQUAL PACKET-RATE CASE				
Bit-rate	$x$	4Mbps	92Mbps	96Mbps
Packet-rate	$u = x/s$	8kpps	8kpps	15kpps
Absolute pkt-loss-rate	$p*u$	8pps	8pps	15pps
Absolute bit-loss-rate	$p*u*s$	4kbps	92kbps	96kbps
Ratio of lost/sent pkts	$p*u/u$	0.1%	0.1%	0.1%
Ratio of lost/sent bits	$p*u*s/(u*s)$	0.1%	0.1%	0.1%

Table 5: Absolute Loss Rates and Loss Ratios for Flows of Small and Large Packets and Both Combined

So far we have merely set up the scenarios. We now consider congestion notification in the scenario. Two TCP flows with the same round trip time aim to equalise their packet-loss-rates over time. That is the number of packets lost in a second, which is the packets per second ( $u$ ) multiplied by the probability that each one is dropped ( $p$ ). Thus TCP converges on the "Equal packet-rate" case, where both flows aim for the same "Absolute packet-loss-rate" (both 8pps in the table).

Packet-mode drop actually gives flows sufficient information to measure their loss-rate in bits per second, if they choose, not just packets per second. Each flow can count the size of a lost or marked packet and scale its rate-response in proportion (as TFRC-SP does). The result is shown in the row entitled "Absolute bit-loss-rate", where the bits lost in a second is the packets per second ( $u$ ) multiplied by the probability of losing a packet ( $p$ ) multiplied by the packet size ( $s$ ). Such an algorithm would try to remove any imbalance in bit-loss-rate such as the wide disparity in the "Equal packet-rate" case (4kbps vs. 92kbps). Instead, a packet-size-dependent algorithm would aim for equal bit-loss-rates, which would drive both flows towards the "Equal bit-rate" case, by driving them to equal bit-loss-rates (both 48kbps in this example).



The explanation so far has assumed that each flow consists of packets of only one constant size. Nonetheless, it extends naturally to flows with mixed packet sizes. In the right-most column of Table 5 a flow of mixed size packets is created simply by considering flow 1 and flow 2 as a single aggregated flow. There is no need for a flow to maintain an average packet size. It is only necessary for the transport to scale its response to each congestion indication by the size of each individual lost (or marked) packet. Taking for example the "Equal packet-rate" case, in one second about 8 small packets and 8 large packets are lost (making closer to 15 than 16 losses per second due to rounding). If the transport multiplies each loss by its size, in one second it responds to  $8 \times 480\text{b}$  and  $8 \times 12,000\text{b}$  lost bits, adding up to 96,000 lost bits in a second. This double checks correctly, being the same as 0.1% of the total bit-rate of 96Mbps. For completeness, the formula for absolute bit-loss-rate is  $p(u_1 \cdot s_1 + u_2 \cdot s_2)$ .

Incidentally, a transport will always measure the loss probability the same irrespective of whether it measures in packets or in bytes. In other words, the ratio of lost to sent packets will be the same as the ratio of lost to sent bytes. (This is why TCP's bit rate is still proportional to packet size even when byte-counting is used, as recommended for TCP in [RFC5681], mainly for orthogonal security reasons.) This is intuitively obvious by comparing two example flows; one with 60B packets, the other with 1500B packets. If both flows pass through a queue with drop probability 0.1%, each flow will lose 1 in 1,000 packets. In the stream of 60B packets the ratio of bytes lost to sent will be 60B in every 60,000B; and in the stream of 1500B packets, the loss ratio will be 1,500B out of 1,500,000B. When the transport responds to the ratio of lost to sent packets, it will measure the same ratio whether it measures in packets or bytes: 0.1% in both cases. The fact that this ratio is the same whether measured in packets or bytes can be seen in Table 5, where the ratio of lost to sent packets and the ratio of lost to sent bytes is always 0.1% in all cases (recall that the scenario was set up with  $p=0.1\%$ ).

This discussion of how the ratio can be measured in packets or bytes is only raised here to highlight that it is irrelevant to this memo! Whether a transport depends on packet size or not depends on how this ratio is used within the congestion control algorithm.

So far we have shown that packet-mode drop passes sufficient information to the transport layer so that the transport can take account of bit-congestion, by using the sizes of the packets that indicate congestion. We have also shown that the transport can choose not to take packet size into account if it wishes. We will now consider whether the transport can know which to do.

## B.2. Bit-Congestible and Packet-Congestible Indications

As a thought-experiment, imagine an idealised congestion notification protocol that supports both bit-congestible and packet-congestible resources. It would require at least two ECN flags, one for each of bit-congestible and packet-congestible resources.

1. A packet-congestible resource trying to code congestion level  $p_p$  into a packet stream should mark the idealised 'packet congestion' field in each packet with probability  $p_p$  irrespective of the packet's size. The transport should then take a packet with the packet congestion field marked to mean just one mark, irrespective of the packet size.
2. A bit-congestible resource trying to code time-varying byte-congestion level  $p_b$  into a packet stream should mark the 'byte congestion' field in each packet with probability  $p_b$ , again irrespective of the packet's size. Unlike before, the transport should take a packet with the byte congestion field marked to count as a mark on each byte in the packet.

This hides a fundamental problem--much more fundamental than whether we can magically create header space for yet another ECN flag, or whether it would work while being deployed incrementally. Distinguishing drop from delivery naturally provides just one implicit bit of congestion indication information--the packet is either dropped or not. It is hard to drop a packet in two ways that are distinguishable remotely. This is a similar problem to that of distinguishing wireless transmission losses from congestive losses.

This problem would not be solved even if ECN were universally deployed. A congestion notification protocol must survive a transition from low levels of congestion to high. Marking two states is feasible with explicit marking, but much harder if packets are dropped. Also, it will not always be cost-effective to implement AQM at every low level resource, so drop will often have to suffice.

We are not saying two ECN fields will be needed (and we are not saying that somehow a resource should be able to drop a packet in one of two different ways so that the transport can distinguish which sort of drop it was!). These two congestion notification channels are a conceptual device to illustrate a dilemma we could face in the future. Section 3 gives four good reasons why it would be a bad idea to allow for packet size by biasing drop probability in favour of small packets within the network. The impracticality of our thought experiment shows that it will be hard to give transports a practical way to know whether to take account of the size of congestion indication packets or not.

Fortunately, this dilemma is not pressing because by design most equipment becomes bit-congested before its packet-processing becomes congested (as already outlined in Section 1.1). Therefore transports can be designed on the relatively sound assumption that a congestion indication will usually imply bit-congestion.

Nonetheless, although the above idealised protocol isn't intended for implementation, we do want to emphasise that research is needed to predict whether there are good reasons to believe that packet congestion might become more common, and if so, to find a way to somehow distinguish between bit and packet congestion [RFC3714].

Recently, the dual resource queue (DRQ) proposal [DRQ] has been made on the premise that, as network processors become more cost effective, per packet operations will become more complex (irrespective of whether more function in the network is desirable). Consequently the premise is that CPU congestion will become more common. DRQ is a proposed modification to the RED algorithm that folds both bit congestion and packet congestion into one signal (either loss or ECN).

Finally, we note one further complication. Strictly, packet-congestible resources are often cycle-congestible. For instance, for routing look-ups load depends on the complexity of each look-up and whether the pattern of arrivals is amenable to caching or not. This also reminds us that any solution must not require a forwarding engine to use excessive processor cycles in order to decide how to say it has no spare processor cycles.

#### Appendix C. Byte-mode Drop Complicates Policing Congestion Response

This section is informative, not normative.

There are two main classes of approach to policing congestion response: i) policing at each bottleneck link or ii) policing at the edges of networks. Packet-mode drop in RED is compatible with either, while byte-mode drop precludes edge policing.

The simplicity of an edge policer relies on one dropped or marked packet being equivalent to another of the same size without having to know which link the drop or mark occurred at. However, the byte-mode drop algorithm has to depend on the local MTU of the line--it needs to use some concept of a 'normal' packet size. Therefore, one dropped or marked packet from a byte-mode drop algorithm is not necessarily equivalent to another from a different link. A policing function local to the link can know the local MTU where the congestion occurred. However, a policer at the edge of the network cannot, at least not without a lot of complexity.

The early research proposals for type (i) policing at a bottleneck link [pBox] used byte-mode drop, then detected flows that contributed disproportionately to the number of packets dropped. However, with no extra complexity, later proposals used packet mode drop and looked for flows that contributed a disproportionate amount of dropped bytes [CHOKe\_Var\_Pkt].

Work is progressing on the congestion exposure protocol (ConEx [RFC6789]), which enables a type (ii) edge policer located at a user's attachment point. The idea is to be able to take an integrated view of the effect of all a user's traffic on any link in the internetwork. However, byte-mode drop would effectively preclude such edge policing because of the MTU issue above.

Indeed, making drop probability depend on the size of the packets that bits happen to be divided into would simply encourage the bits to be divided into smaller packets in order to confuse policing. In contrast, as long as a dropped/marked packet is taken to mean that all the bytes in the packet are dropped/marked, a policer can remain robust against bits being re-divided into different size packets or across different size flows [Rate\_fair\_Dis].

#### Appendix D. Changes from Previous Versions

To be removed by the RFC Editor on publication.

Full incremental diffs between each version are available at  
<<http://tools.ietf.org/wg/tsvwg/draft-ietf-tsvwg-byte-pkt-congest/>>  
(courtesy of the rfcdiff tool):

From -11 to -12: Following the second pass through the IESG:

- \* Section 2.1 [Barry Leiba]:
  - + s/No other choice makes sense,/Subject to the exceptions below, no other choice makes sense,/
  - + s/Exceptions to these recommendations MAY be necessary /Exceptions to these recommendations may be necessary /
- \* Sections 3.2 and 4.2.3 [Joel Jaeggli]:
  - + Added comment to section 4.2.3 that the examples given are not in widespread production use, but they give evidence that it is possible to follow the advice given.
  - + Section 4.2.3:

- OLD: Although there are no known proposals, it would also be possible and perfectly valid to make control packets robust against drop by explicitly requesting a lower drop probability using their Diffserv code point [RFC2474] to request a scheduling class with lower drop.  
NEW: Although there are no known proposals, it would also be possible and perfectly valid to make control packets robust against drop by requesting a scheduling class with lower drop probability, by re-marking to a Diffserv code point [RFC2474] within the same behaviour aggregate.
- appended "Similarly applications, over non-TCP transports could make any packets that are effectively control packets more robust by using Diffserv, data duplication, FEC etc."
- + Updated Wischik ref and added "Reducing Web Latency: the Virtue of Gentle Aggression" ref.
- \* Expanded more abbreviations (CoDel, PIE, MTU).
- \* Section 1. Intro [Stephen Farrell]:
  - + In the places where the doc describes the dichotomy between 'long-term goal' and 'expediency' the words long term goal and expedient have been introduced, to more explicitly refer back to this introductory para (S.2.1 & S.2.3).
  - + Added explanation of what scaling with packet size means.
- \* Conclusions [Benoit Claise]:
  - + OLD: For the specific case of RED, this means that byte-mode queue measurement will often be appropriate although byte-mode drop is strongly deprecated.  
NEW: For the specific case of RED, this means that byte-mode queue measurement will often be appropriate but the use of byte-mode drop is very strongly discouraged.

From -10 to -11: Following a further WGLC:

- \* Abstract: clarified that advice applies to all AQMs including newer ones
- \* Abstract & Intro: changed 'read' to 'detect', because you don't read losses, you detect them.

- \* S.1. Introduction: Disambiguated summary of advice on queue measurement.
- \* Clarified that the doc deprecates any preference based solely on packet size, it's not only against preferring smaller packets.
- \* S.4.1.2. Congestion Measurement without a Queue: Explained that a queue of TXOPs represents a queue into spectrum congested by too many bits.
- \* S.5.2: Bit- & Packet-congestible Network: Referred to explanation in S.4.1.2 to make the point that TXOPs are not a primary unit of workload like bits and packets are, even though you get queues of TXOPs.
- \* 6. Security: Disambiguated 'bias towards'.
- \* 8. Conclusions: Made consistent with recommendation to use time if possible for queue measurement.

From -09 to -10: Following IESG review:

- \* Updates 2309: Left header unchanged reflecting eventual IESG consensus [Sean Turner, Pete Resnick].
- \* S.1 Intro: This memo adds to the congestion control principles enumerated in BCP 41 [Pete Resnick]
- \* Abstract, S.1, S.1.1, s.1.2 Intro, Scoping and Example: Made applicability to all AQMs clearer listing some more example AQMs and explained that we always use RED for examples, but this doesn't mean it's not applicable to other AQMs. [A number of reviewers have described the draft as "about RED"]
- \* S.1 & S.2.1 Queue measurement: Explained that the choice between measuring the queue in packets or bytes is only relevant if measuring it in time units is infeasible [So as not to imply that we haven't noticed the advances made by PDPC & CoDel]
- \* S.1.1. Terminology: Better explained why hybrid systems congested by both packets and bytes are often designed to be treated as bit-congestible [Richard Barnes].
- \* S.2.1. Queue measurement advice: Added examples. Added a counter-example to justify SHOULDs rather than MUSTs. Pointed to S.4.1 for a list of more complicated scenarios. [Benson]

Schliesser, OpsDir]

- \* S2.2. Recommendation on Encoding Congestion Notification: Removed SHOULD treat packets equally, leaving only SHOULD NOT drop dependent on packet size, to avoid it sounding like we're saying QoS is not allowed. Pointed to possible app-specific legacy use of byte-mode as a counter-example that prevents us saying MUST NOT. [Pete Resnick]
- \* S.2.3. Recommendation on Responding to Congestion: capitalised the two SHOULDs in recommendations for TCP, and gave possible counter-examples. [noticed while dealing with Pete Resnick's point]
- \* S2.4. Splitting & Merging: RTCP -> RTP/RTCP [Pete McCann, Gen-ART]
- \* S.3.2 Small != Control: many control packets are small -> ...tend to be small [Stephen Farrell]
- \* S.3.1 Perverse incentives: Changed transport designers to app developers [Stephen Farrell]
- \* S.4.1.1. Fixed Size Packet Buffers: Nearly completely re-written to simplify and to reverse the advice when the underlying resource is bit-congestible, irrespective of whether the buffer consists of fixed-size packet buffers. [Richard Barnes & Benson Schliesser]
- \* S.4.2.1.2. Packet Size Bias Regardless of AQM: Largely re-written to reflect the earlier change in advice about fixed-size packet buffers, and to primarily focus on getting rid of tail-drop, not various nuances of tail-drop. [Richard Barnes & Benson Schliesser]
- \* Editorial corrections [Tim Bray, AppsDir, Pete McCann, Gen-ART and others]
- \* Updated refs (two I-Ds have become RFCs). [Pete McCann]

From -08 to -09: Following WG last call:

- \* S.2.1: Made RED-related queue measurement recommendations clearer
- \* S.2.3: Added to "Recommendation on Responding to Congestion" to make it clear that we are definitely not saying transports have to equalise bit-rates, just how to do it and not do it, if you

want to.

- \* S.3: Clarified motivation sections S.3.3 "Transport-Independent Network" and S.3.5 "Implementation Efficiency"
- \* S.3.4: Completely changed motivating argument from "Scaling Congestion Control with Packet Size" to "Partial Deployment of AQM".

From -07 to -08:

- \* Altered abstract to say it provides best current practice and highlight that it updates RFC2309
- \* Added null IANA section
- \* Updated refs

From -06 to -07:

- \* A mix-up with the corollaries and their naming in 2.1 to 2.3 fixed.

From -05 to -06:

- \* Primarily editorial fixes.

From -04 to -05:

- \* Changed from Informational to BCP and highlighted non-normative sections and appendices
- \* Removed language about consensus
- \* Added "Example Comparing Packet-Mode Drop and Byte-Mode Drop"
- \* Arranged "Motivating Arguments" into a more logical order and completely rewrote "Transport-Independent Network" & "Scaling Congestion Control with Packet Size" arguments. Removed "Why Now?"
- \* Clarified applicability of certain recommendations
- \* Shifted vendor survey to an Appendix
- \* Cut down "Outstanding Issues and Next Steps"



- \* Re-drafted the start of the conclusions to highlight the three distinct areas of concern
- \* Completely re-wrote appendices
- \* Editorial corrections throughout.

From -03 to -04:

- \* Reordered Sections 2 and 3, and some clarifications here and there based on feedback from Colin Perkins and Mirja Kuehlewind.

From -02 to -03 (this version)

- \* Structural changes:
  - + Split off text at end of "Scaling Congestion Control with Packet Size" into new section "Transport-Independent Network"
  - + Shifted "Recommendations" straight after "Motivating Arguments" and added "Conclusions" at end to reinforce Recommendations
  - + Added more internal structure to Recommendations, so that recommendations specific to RED or to TCP are just corollaries of a more general recommendation, rather than being listed as a separate recommendation.
  - + Renamed "State of the Art" as "Critical Survey of Existing Advice" and retitled a number of subsections with more descriptive titles.
  - + Split end of "Congestion Coding: Summary of Status" into a new subsection called "RED Implementation Status".
  - + Removed text that had been in the Appendix "Congestion Notification Definition: Further Justification".
- \* Reordered the intro text a little.
- \* Made it clearer when advice being reported is deprecated and when it is not.
- \* Described AQM as in network equipment, rather than saying "at the network layer" (to side-step controversy over whether functions like AQM are in the transport layer but in network

equipment).

- \* Minor improvements to clarity throughout

From -01 to -02:

- \* Restructured the whole document for (hopefully) easier reading and clarity. The concrete recommendation, in RFC2119 language, is now in Section 8.

From -00 to -01:

- \* Minor clarifications throughout and updated references

From briscoe-byte-pkt-mark-02 to ietf-byte-pkt-congest-00:

- \* Added note on relationship to existing RFCs
- \* Posed the question of whether packet-congestion could become common and deferred it to the IRTF ICCRG. Added ref to the dual-resource queue (DRQ) proposal.
- \* Changed PCN references from the PCN charter & architecture to the PCN marking behaviour draft most likely to imminently become the standards track WG item.

From -01 to -02:

- \* Abstract reorganised to align with clearer separation of issue in the memo.
- \* Introduction reorganised with motivating arguments removed to new Section 3.
- \* Clarified avoiding lock-out of large packets is not the main or only motivation for RED.
- \* Mentioned choice of drop or marking explicitly throughout, rather than trying to coin a word to mean either.
- \* Generalised the discussion throughout to any packet forwarding function on any network equipment, not just routers.
- \* Clarified the last point about why this is a good time to sort out this issue: because it will be hard / impossible to design new transports unless we decide whether the network or the transport is allowing for packet size.

- \* Added statement explaining the horizon of the memo is long term, but with short term expediency in mind.
- \* Added material on scaling congestion control with packet size (Section 3.4).
- \* Separated out issue of normalising TCP's bit rate from issue of preference to control packets (Section 3.2).
- \* Divided up Congestion Measurement section for clarity, including new material on fixed size packet buffers and buffer carving (Section 4.1.1 & Section 4.2.1) and on congestion measurement in wireless link technologies without queues (Section 4.1.2).
- \* Added section on 'Making Transports Robust against Control Packet Losses' (Section 4.2.3) with existing & new material included.
- \* Added tabulated results of vendor survey on byte-mode drop variant of RED (Table 3).

From -00 to -01:

- \* Clarified applicability to drop as well as ECN.
- \* Highlighted DoS vulnerability.
- \* Emphasised that drop-tail suffers from similar problems to byte-mode drop, so only byte-mode drop should be turned off, not RED itself.
- \* Clarified the original apparent motivations for recommending byte-mode drop included protecting SYN's and pure ACK's more than equalising the bit rates of TCP's with different segment sizes. Removed some conjectured motivations.
- \* Added support for updates to TCP in progress (ackcc & ecn-syn-ack).
- \* Updated survey results with newly arrived data.
- \* Pulled all recommendations together into the conclusions.
- \* Moved some detailed points into two additional appendices and a note.

\* Considerable clarifications throughout.

\* 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://bobbriscoe.net/>

Jukka Manner  
Aalto University  
Department of Communications and Networking (Comnet)  
P.O. Box 13000  
FIN-00076 Aalto  
Finland

Phone: +358 9 470 22481  
EMail: jukka.manner@aalto.fi  
URI: <http://www.netlab.tkk.fi/~jmanner/>



Network Working Group  
Internet-Draft  
Intended status: Standard Track

E. Crabbe, Ed.  
Google  
L. Yong, Ed.  
Huawei USA  
X. Xu, Ed.  
Huawei Technologies

Expires: September 2014

February 13, 2014

Generic UDP Encapsulation for IP Tunneling  
draft-ietf-tsvwg-gre-in-udp-encap-01

Abstract

This document describes a method of encapsulating arbitrary protocols within GRE and UDP headers. In this encapsulation, the source UDP port may be used as an entropy field for purposes of load balancing while the payload protocol may be identified by the GRE Protocol Type.

Status of This Document

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 13, 2014.

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.....	3
1.1. Applicability Statements.....	3
2. Terminology.....	4
2.1. Requirements Language.....	4
3. Procedures.....	4
4. Encapsulation Considerations.....	8
5. Backward Compatibility.....	9
6. IANA Considerations.....	9
7. Security Considerations.....	10
7.1. Vulnerability.....	10
8. Acknowledgements.....	10
9. Contributors.....	10
10. References.....	11
10.1. Normative References.....	11
10.2. Informative References.....	12
11. Authors' Addresses.....	13

## 1. Introduction

Load balancing, or more specifically, statistical multiplexing of traffic using Equal Cost Multi-Path (ECMP) and/or Link Aggregation Groups (LAGs) in IP networks is a widely used technique for creating higher capacity networks out of lower capacity links. Most existing routers in IP networks are already capable of distributing IP traffic flows over ECMP paths and/or LAGs on the basis of a hash function performed on flow invariant fields in IP packet headers and their payload protocol headers. Specifically, when the IP payload is a User Datagram Protocol (UDP) [RFC0768] or Transmission Control Protocol (TCP) packet, router hash functions frequently operate on the five-tuple of the source IP address, the destination IP address, the source port, the destination port, and the protocol/next-header

Several tunneling techniques are in common use in IP networks, such as Generic Routing Encapsulation (GRE) [RFC2784], MPLS [RFC4023] and L2TPv3 [RFC3931]. GRE is an increasingly popular encapsulation choice, especially in environments where MPLS is unavailable or unnecessary. Unfortunately, use of common GRE endpoints may reduce the entropy available for use in load balancing, especially in environments where the GRE Key field [RFC2890] is not readily available for use as entropy in forwarding decisions.

This document defines a generic GRE-in-UDP encapsulation for tunneling arbitrary network protocol payloads across an IP network environment where ECMP or LAGs are used. The GRE header provides payload protocol de-multiplexing by way of its protocol type field [RFC2784] while the UDP header provides additional entropy by way of its source port.

This encapsulation method requires no changes to the transit IP network. Hash functions in most existing IP routers may utilize and benefit from the use of a GRE-in-UDP tunnel is without needing any change or upgrade to their ECMP implementation. The encapsulation mechanism is applicable to a variety of IP networks including Data Center and wide area networks.

### 1.1. Applicability Statements

It is recommended to use the GRE-in-UDP encapsulation technology in a Service Provider (SP) network and/or DC network where the congestion control is not a concern, rather than over the Internet where the congestion control is a must. Furthermore, packet filters should be added so as to prevent GRE-in-UDP packets from escaping



from the service provider networks due to mis-configuration or packet errors.

## 2. Terminology

The terms defined in [RFC768] are used in this document.

### 2.1. 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 [RFC2119].

## 3. Procedures

When a tunnel ingress device conforming to this document receives a packet, the ingress MUST encapsulate the packet in UDP and GRE headers and set the destination port of the UDP header to [TBD] Section 6. The ingress device must also insert the payload protocol type in the GRE Protocol Type field. The ingress device SHOULD set the UDP source port based on flow invariant fields from the payload header, otherwise it should be set to a randomly selected constant value, e.g. zero, to avoid packet flow reordering. How a tunnel ingress generates entropy from the payload is outside the scope of this document. The tunnel ingress MUST encode its own IP address as the source IP address and the egress tunnel endpoint IP address. The TTL field in the IP header must be set to a value appropriate for delivery of the encapsulated packet to the tunnel egress endpoint.

When the tunnel egress receives a packet, it must remove the outer UDP and GRE headers. Section 5 describes the error handling when this entity is not instantiated at the tunnel egress.

To simplify packet processing at the tunnel egress, packets destined to this assigned UDP destination port [TBD] MAY have their UDP checksum set to zero. In the environment where the UDP packets may be mis-delivered [RFC5405], UDP checksum SHOULD be used. Upon receiving a packet with a non-zero checksum, tunnel egress MUST perform the UDP checksum verification. For an IPv6 network, UDP checksum SHOULD be used; if the checksum needs to be disabled for performance or implementation concerns, the considerations described in [RFC6935][RFC6936] MUST be examined. The Sequence flags MUST set to zero.

The tunnel ingress may set the GRE Key Present, Sequence Number Present, and Checksum Present bits and associated fields in the GRE header defined by [RFC2784] and [RFC2890].

In addition IPv6 nodes MUST conform to the following:

1. the IPv6 tunnel ingress and egress SHOULD follow the node requirements specified in Section 4 of [RFC6936] and the usage requirements specified in Section 5 of [RFC6936].
2. IPv6 transit nodes SHOULD follow the requirements 9, 10, 11 specified in Section 5 of [RFC6936].

The tunnel ingress may set the GRE Key Present, Sequence Number Present, and Checksum Present bits and associated fields in the GRE header defined by [RFC2784] and [RFC2890].

The format of the GRE-in-UDP encapsulation for both IPv4 and IPv6 outer headers is shown in the following figures:

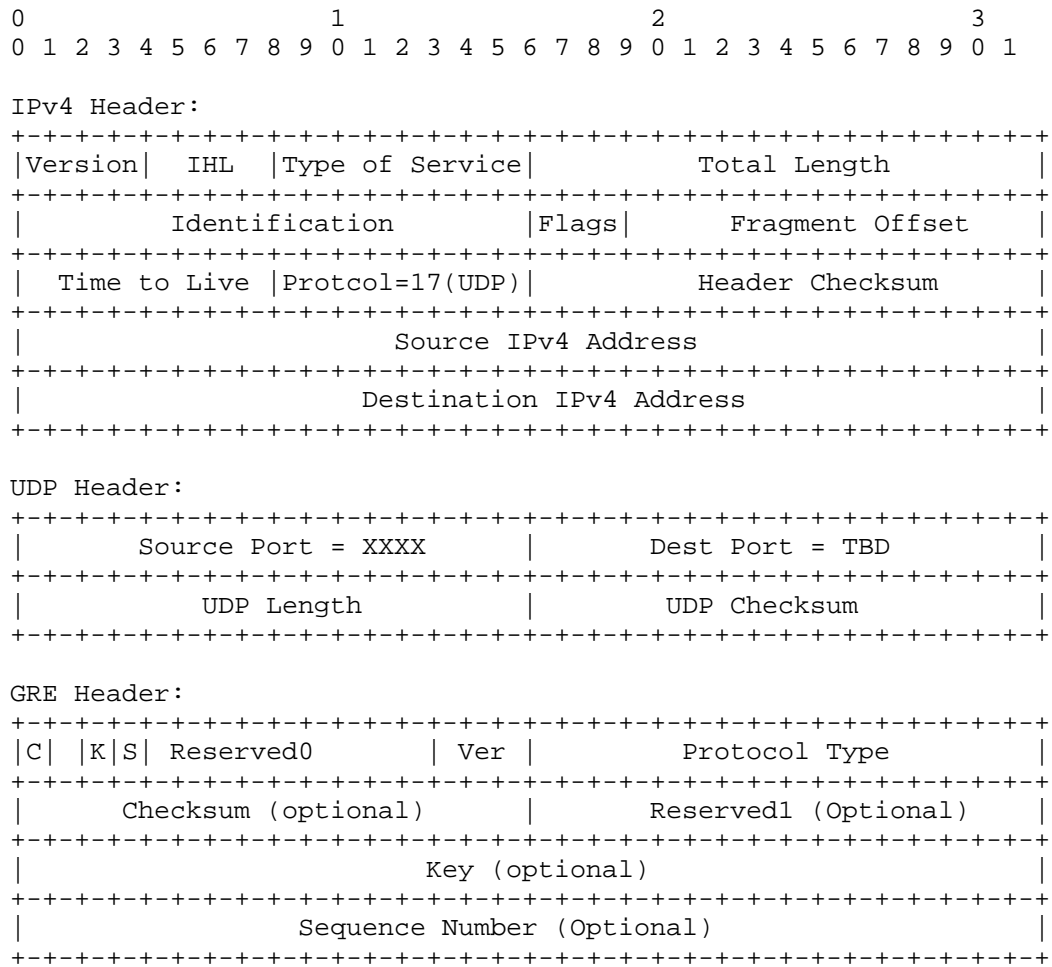


Figure 1 UDP+GRE IPv4 headers

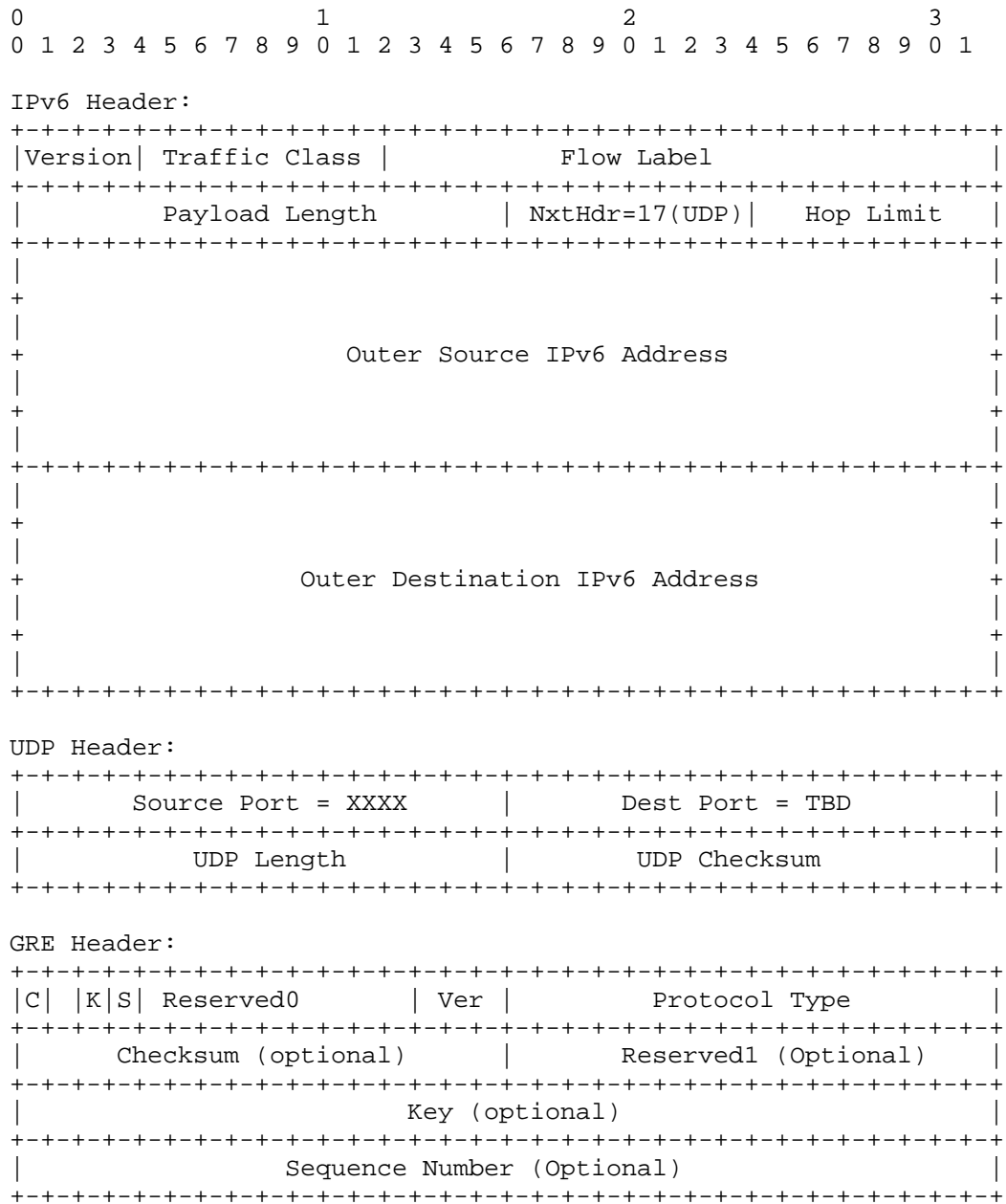


Figure 2 UDP+GRE IPv6 headers

The total overhead increase for a UDP+GRE tunnel without use of optional GRE fields, representing the lowest total overhead increase, is 32 bytes in the case of IPv4 and 52 bytes in the case of IPv6. The total overhead increase for a UDP+GRE tunnel with use of GRE Key, Sequence and Checksum Fields, representing the highest total overhead increase, is 44 bytes in the case of IPv4 and 64 bytes in the case of IPv6.

#### 4. Encapsulation Considerations

GRE-in-UDP encapsulation allows the tunneled traffic to be unicast, broadcast, or multicast traffic. Entropy may be generated from the header of tunneled unicast or broadcast/multicast packets at tunnel ingress. The mapping mechanism between the tunneled multicast traffic and the multicast capability in the IP network is transparent and independent to the encapsulation and is outside the scope of this document.

If tunnel ingress must perform the fragmentation [GREMTU] on a packet before encapsulation, it MUST use the same source UDP port for all packet fragments. This ensures that the transit routers will forward the packet fragments on the same path. GRE-in-UDP encapsulation introduces some overhead as mentioned in section 3, which reduces the effective Maximum Transmission Unit (MTU) size. An operator should factor in this addition overhead bytes when considering an MTU size for the payload to reduce the likelihood of fragmentation.

To ensure the tunneled traffic gets the same treatment over the IP network, prior to the encapsulation process, tunnel ingress should process the payload to get the proper parameters to fill into the IP header such as DiffServ [RFC2983]. Tunnel end points that support ECN MUST use the method described in [RFC6040] for ECN marking propagation. This process is outside of the scope of this document.

Note that the IPv6 header [RFC2460] contains a flow label field that may be used for load balancing in an IPv6 network [RFC6438]. Thus in an IPv6 network, either GRE-in-UDP or flow labels may be used for improving load balancing performance. Use of GRE-in-UDP encapsulation provides a unified hardware implementation for load

balancing in an IP network independent of the IP version(s) in use. However, if UDP checksum has to be used in the environment, a flow label based load balancing is advantage in performance and implementation.

## 5. Backward Compatibility

It is assumed that tunnel ingress routers must be upgraded in order to support the encapsulations described in this document.

No change is required at transit routers to support forwarding of the encapsulation described in this document.

If a router that is intended for use as a tunnel egress does not support the GRE-in-UDP encapsulation described in this document, it will not be listening on destination port [TBD]. In these cases, the router will conform to normal UDP processing and respond to the tunnel ingress with an ICMP message indicating "port unreachable" according to [RFC792]. Upon receiving this ICMP message, the tunnel ingress MUST NOT continue to use GRE-in-UDP encapsulation toward this tunnel egress without management intervention.

## 6. IANA Considerations

IANA is requested to make the following allocation:

Service Name: GRE-in-UDP  
Transport Protocol(s): UDP  
Assignee: IESG <iesg@ietf.org>  
Contact: IETF Chair <chair@ietf.org>  
Description: GRE-in-UDP Encapsulation  
Reference: [This.I-D]  
Port Number: TBD  
Service Code: N/A  
Known Unauthorized Uses: N/A  
Assignment Notes: N/A

## 7. Security Considerations

### 7.1. Vulnerability

Neither UDP nor GRE encapsulation effects security for the payload protocol. When using GRE-in-UDP, Network Security in a network is the same as that of a network using GRE.

Use of ICMP for signaling of the GRE-in-UDP encapsulation capability adds a security concern. Tunnel ingress devices may want to validate the origin of ICMP Port Unreachable messages before taking action. The mechanism for performing this validation is out of the scope of this document.

In an instance where the UDP src port is not set based on the flow invariant fields from the payload header, a random port SHOULD be selected in order to minimize the vulnerability to off-path attacks. [RFC6056] How the src port randomization occurs is outside scope of this document.

Using one standardized value in UDP destination port for an encapsulation indication may increase the vulnerability of off-path attack. To overcome this, tunnel egress may request tunnel ingress using a different and specific value [RFC6056] in UDP destination port for the GRE-in-UDP encapsulation indication. How the tunnel end points communicate the value is outside scope of this document.

## 8. Acknowledgements

Authors like to thank Vivek Kumar, Ron Bonica, Joe Touch, Ruediger Geib, Gorrry Fairhurst, David Black, Lar Edds, Lloyd, and many others for their review and valuable input on this draft.

## 9. Contributors

The following people all contributed significantly to this document and are listed below in alphabetical order:

John E. Drake  
Juniper Networks

Email: jdrake@juniper.net

Adrian Farrel  
Juniper Networks

Email: adrian@olddog.co.uk

Vishwas Manral  
Hewlett-Packard Corp.  
3000 Hanover St, Palo Alto.

Email: vishwas.manral@hp.com

Carlos Pignataro  
Cisco Systems  
7200-12 Kit Creek Road  
Research Triangle Park, NC 27709 USA

EMail: cpignata@cisco.com

Yongbing Fan  
China Telecom  
Guangzhou, China.  
Phone: +86 20 38639121

## 10. References

### 10.1. Normative References

- [RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC791] DARPA, "Internet Protocol", RFC791, September 1981
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC2119, March 1997.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.
- [RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC2890, September 2000.



- [RFC2983] Black, D., "Differentiated Services and Tunnels", RFC2983, October 2000.
- [RFC5405] Eggert, L., "Unicast UDP Usage Guideline for Application Designers", RFC5405, November 2008.
- [RFC6040] Briscoe, B., "Tunneling of Explicit Congestion Notification", RFC6040, November 2010
- [RFC6438] Carpenter, B., Amante, S., "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in tunnels", RFC6438, November, 2011
- [RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, April 2013.
- [RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, April 2013.

#### 10.2. Informative References

- [RFC792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.
- [RFC4023] Worster, T., Rekhter, Y., and E. Rosen, "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, March 2005.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, April 2007.
- [RFC6056] Larsen, M. and Gont, F., "Recommendations for Transport-Protocol Port Randomization", RFC6056, January 2011

- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.
- [GREMTU] Bonica, R., "A Fragmentation Strategy for Generic Routing Encapsulation (GRE)", draft-bonica-intara-gre-mtu, work in progress

## 11. Authors' Addresses

Edward Crabbe (editor)  
Google  
1600 Amphitheatre Parkway  
Mountain View, CA 94102  
US

Lucy Yong (editor)  
Huawei Technologies, USA

Email: [lucy.yong@huawei.com](mailto:lucy.yong@huawei.com)

Xiaohu Xu (editor)  
Huawei Technologies,  
Beijing, China

Email: [xuxiaohu@huawei.com](mailto:xuxiaohu@huawei.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.

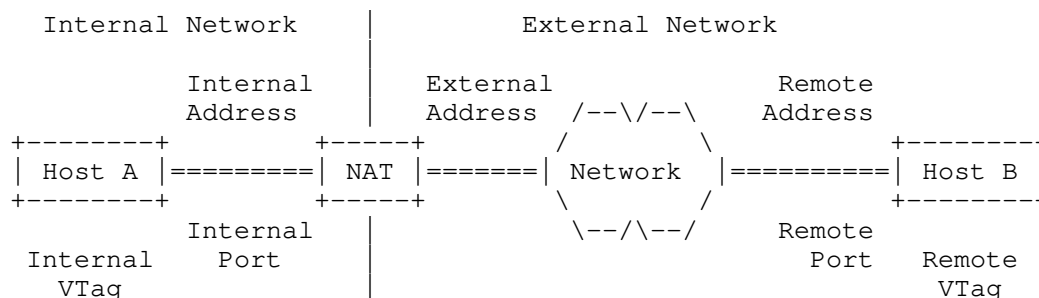


Figure 1: Basic Network Setup

## 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.

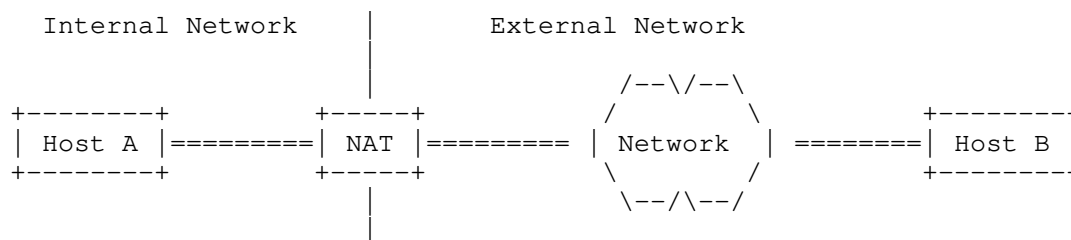


Figure 2: Single NAT Function Scenario

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

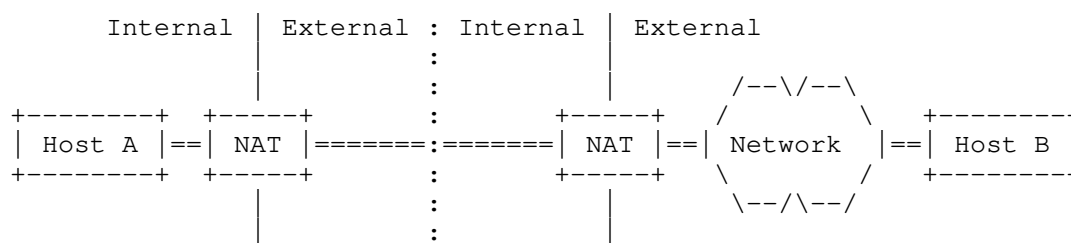


Figure 3: Serial NAT Functions Scenario

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.

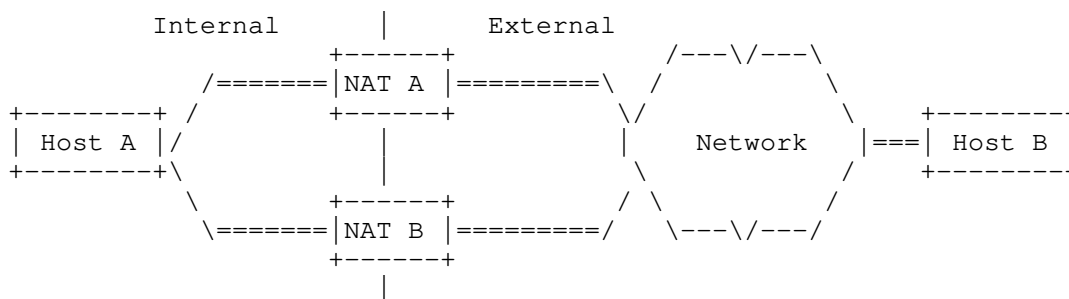


Figure 4: Parallel NAT Functions Scenario

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).

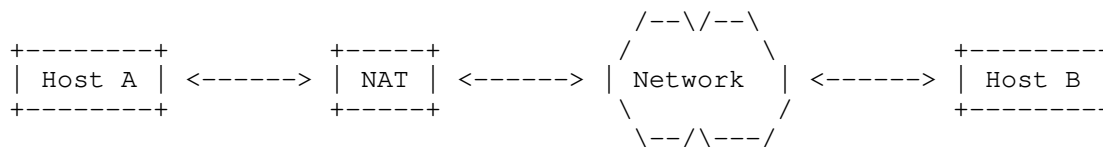
The modification of outgoing SCTP packets sent from an internal host is simple: the source address of the packets has to be replaced with the External-Address. It might also be necessary to establish some state in the NAT function to later handle incoming packets.

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

```

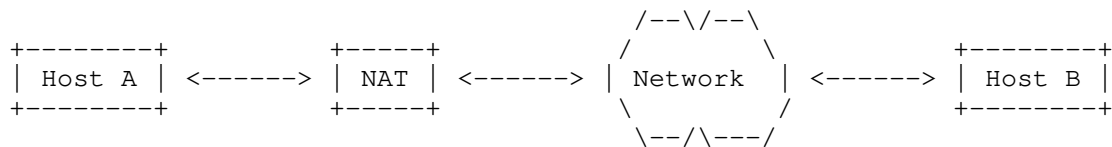
Normally a NAT binding table entry will be created.

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.

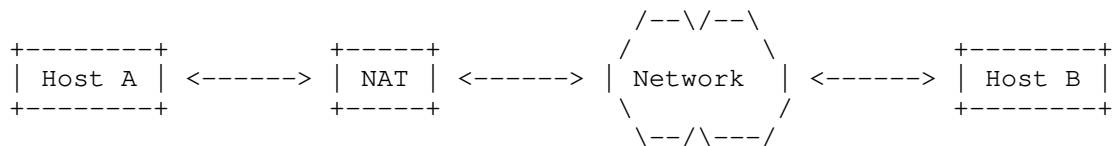


Int-Addr:Int-Port -----> Rem-Addr:Rem-Port  
                                   Rem-VTag

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

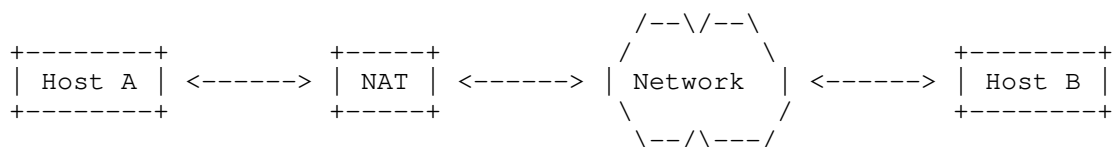
Lookup(Int-VTag, Int-Port, \*, Rem-Port)  
 Update(\*, \*, Initiate-Tag, \*)

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

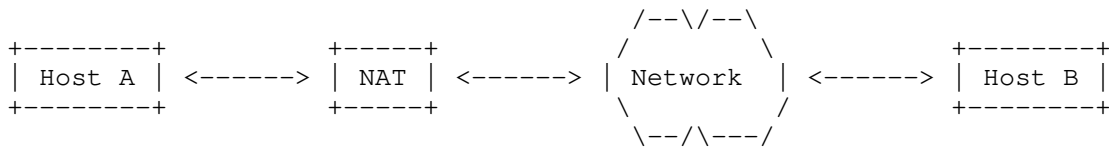
Lookup(\*, Int-Port, Rem-VTag, Rem-Port)

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

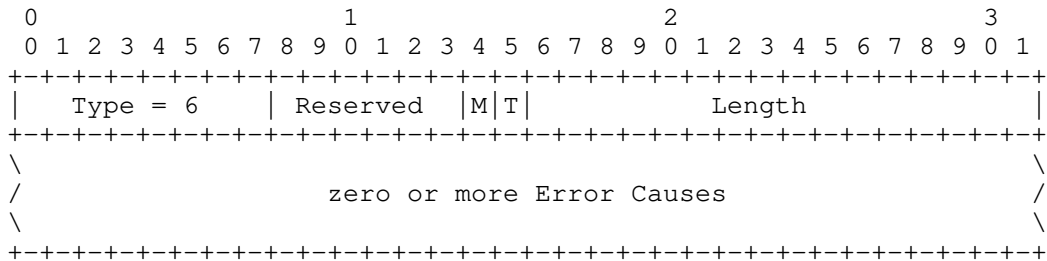
5. Data Formats

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

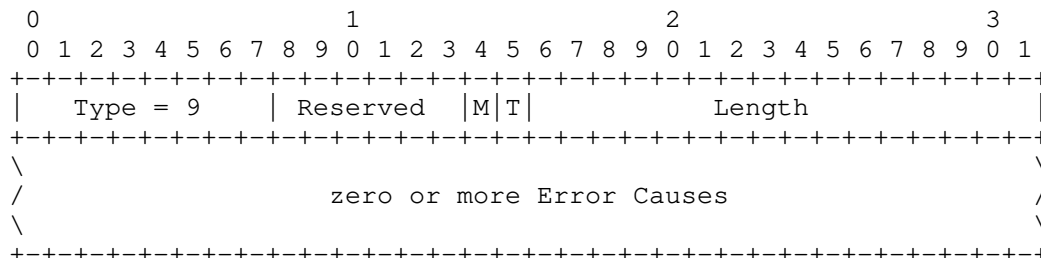


The ABORT chunk is extended to add the new 'M bit'. The M bit indicates to the receiver of the ABORT chunk that the chunk was not generated by the peer SCTP endpoint, but instead by a middle box (e.g., NAT).

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



## 5.1.2. Extended ERROR Chunk



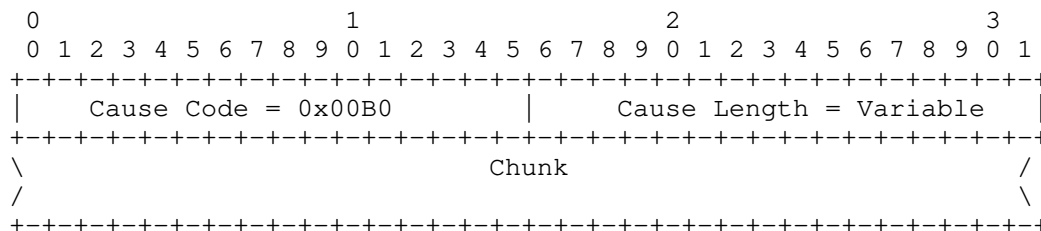
The ERROR chunk defined in [RFC4960] is extended to add the new 'M bit'. The M bit indicates to the receiver of the ERROR chunk that the chunk was not generated by the peer SCTP endpoint, but instead by a middle box.

[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)

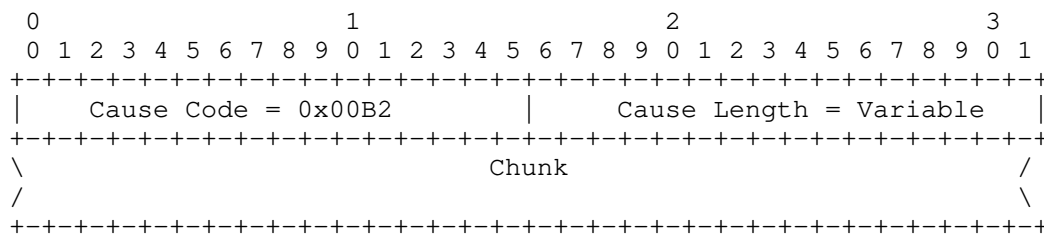
This field holds the length in bytes of the error cause. The value MUST be the length of the Cause-Specific Information plus 4.

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.

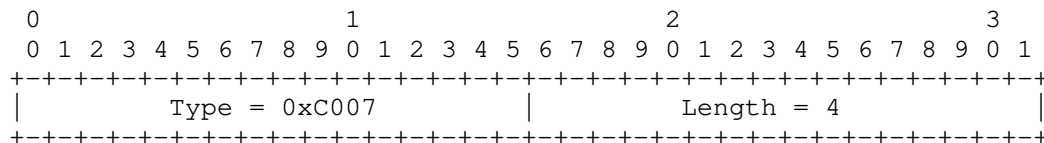
[NOTE to RFC-Editor: Assignment of cause code to be confirmed by IANA.]

### 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)

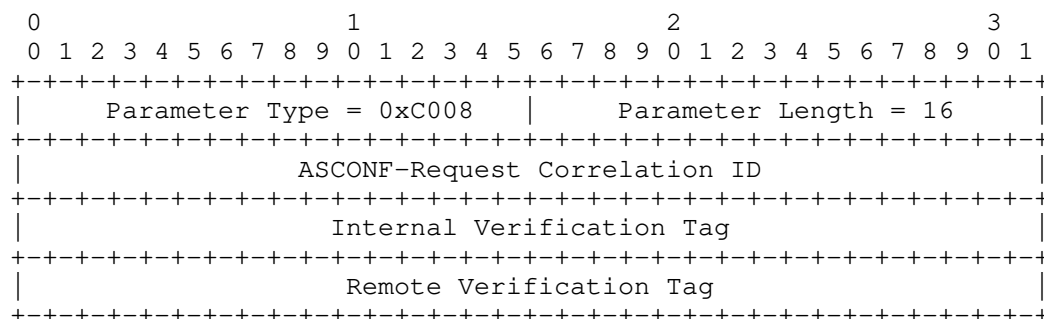
This field holds the length in bytes of the parameter. The value MUST be 4.

[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

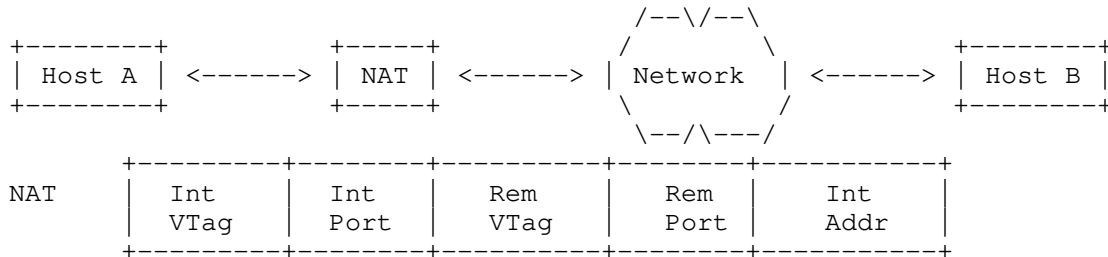
Please note that this section is informational only.

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:

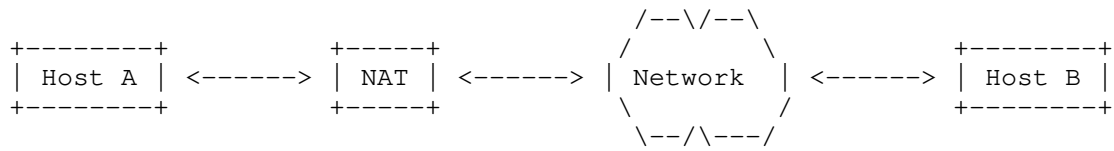
	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
NAT					
	1234	1	0	2	10.0.0.1

```

                                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.



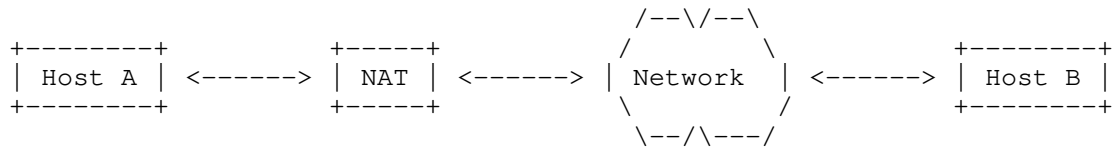
INIT ACK[Initiate-Tag = 5678]  
 192.0.2.1:1 <----- 203.0.113.1:2  
 Int-VTag = 1234

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

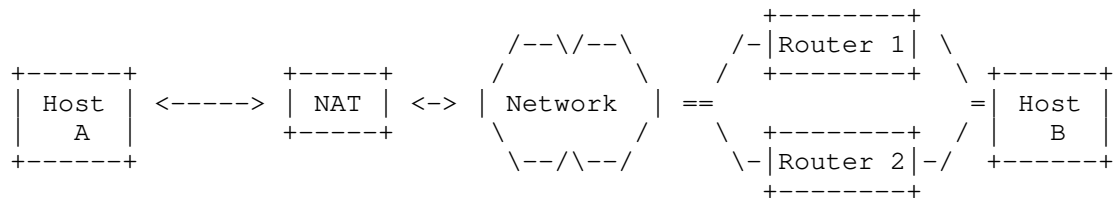
COOKIE ACK  
 192.0.2.1:1 <----- 203.0.113.1:2  
 Int-VTag = 1234

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
  
```

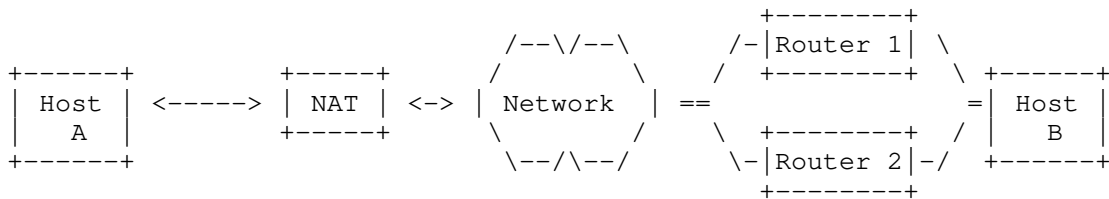
NAT function creates entry:

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.



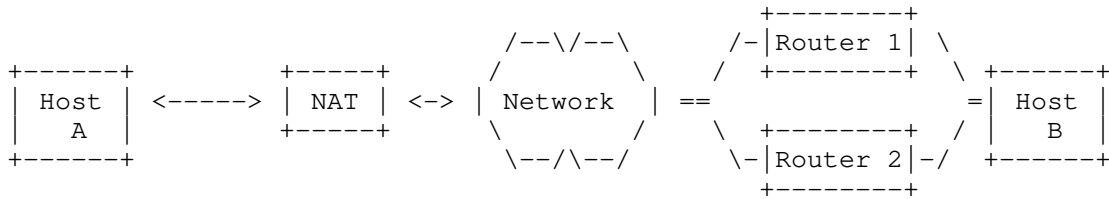
```
INIT ACK[Initiate-tag = 5678, IP-Addr = 203.0.113.129]
192.0.2.1:1 <----- 203.0.113.1:2
                        Int-VTag = 1234
```

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

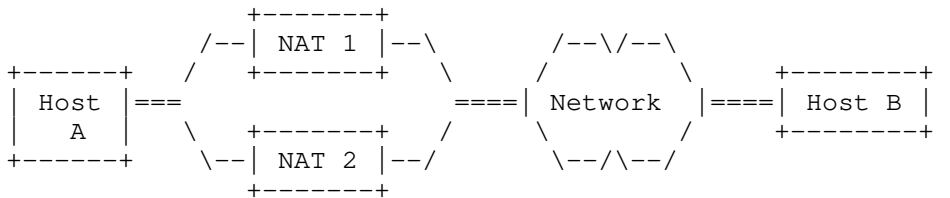
COOKIE ECHO  
192.0.2.1:1 -----> 203.0.113.1:2  
Rem-VTag = 5678

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:

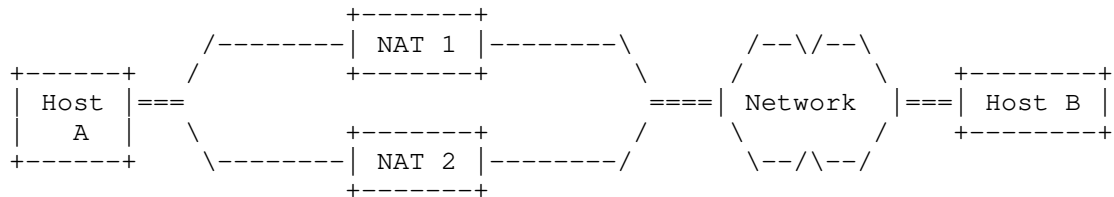
NAT 1	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	0	2	10.0.0.1

```

                                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:

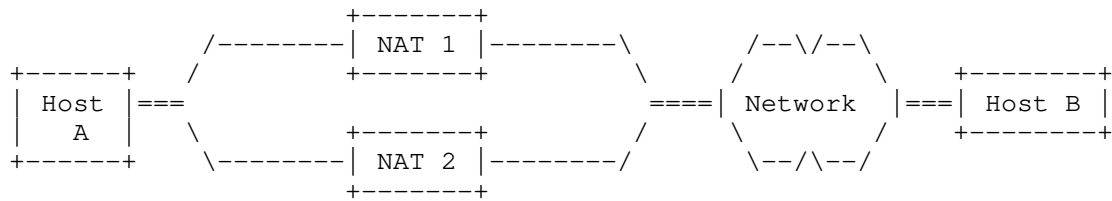
NAT 1	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr
	1234	1	5678	2	10.0.0.1

```

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.



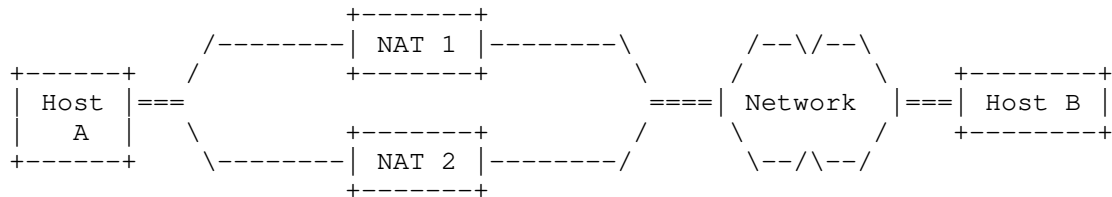
COOKIE ECHO  
 10.0.0.1:1 -----> 203.0.113.1:2  
 Rem-VTag = 5678

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 VTag	Int Port	Rem VTag	Rem Port	Int 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

```

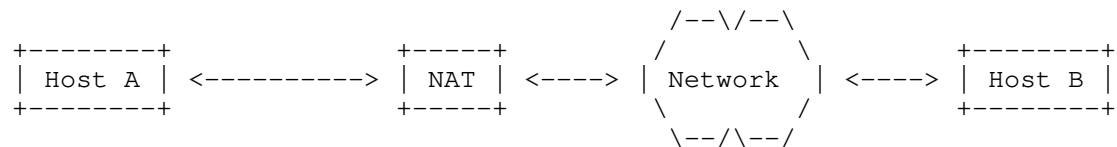
```

                        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.



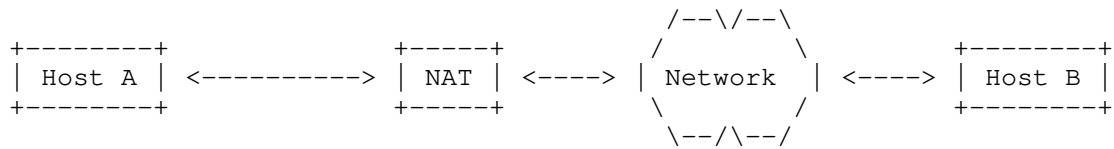
NAT	Int VTag	Int Port	Rem VTag	Rem Port	Int Addr

```

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

```

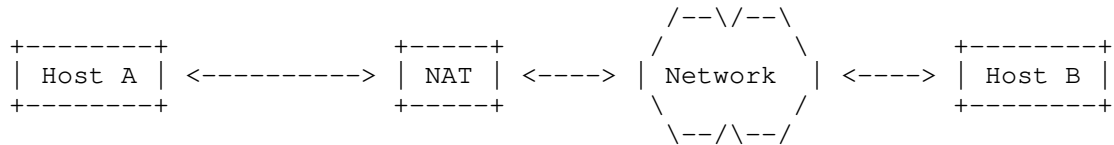
The NAT function cannot find an entry in the NAT binding table for the association. It sends a packet containing an ERROR chunk with the M bit set and the cause "NAT state missing".



```

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.



```

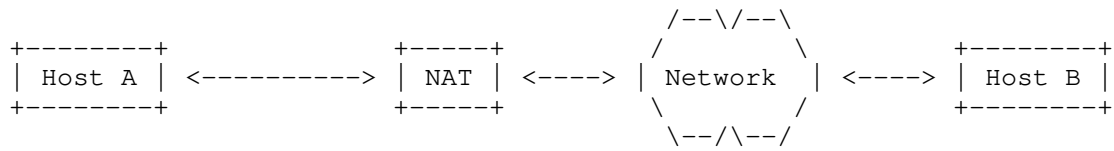
ASCONF [ADD-IP, DELETE-IP, Int-VTag=1234, Rem-VTag = 5678]
10.0.0.1:1 -----> 203.0.113.129:2
      Rem-VTag = 5678
  
```

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

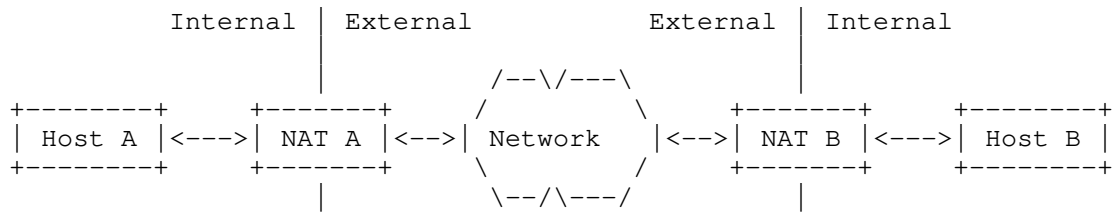
DATA  
 10.0.0.1:1 -----> 203.0.113.1:2  
 Rem-VTag = 5678

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

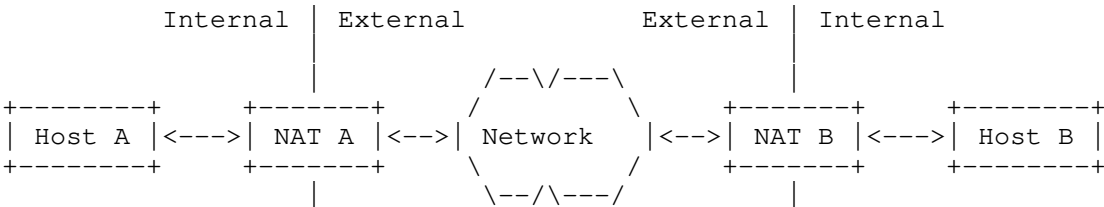
```

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.

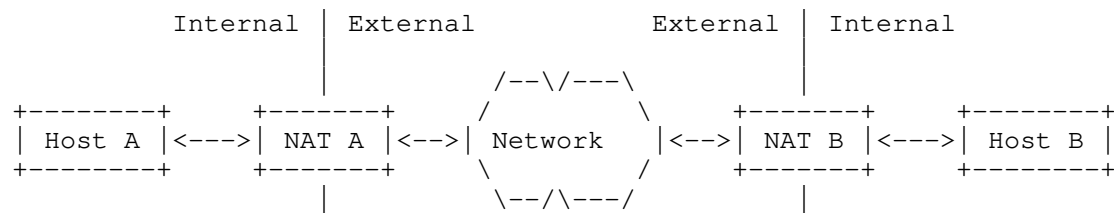


INIT[Initiate-Tag = 5678]  
192.0.2.1:1 <-- 10.1.0.1:2  
Rem-VTag = 0

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:

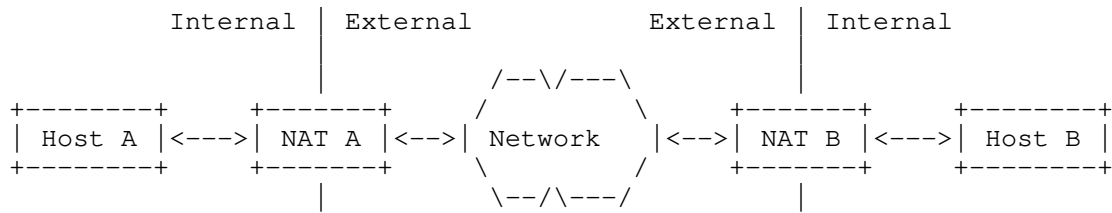


VTag != Int-VTag, but Rem-VTag == 0, find entry.

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
  
```

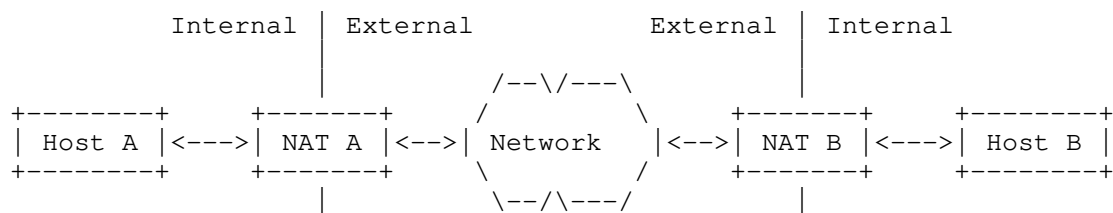
NAT function B updates entry:

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



Network WG  
Internet-Draft  
Expires: January 4, 2015  
Intended Status: Standards Track  
Updates: RFC 2872 (if accepted)

James Polk  
Subha Dhesikan  
Cisco Systems  
July 4, 2014

Resource Reservation Protocol (RSVP) Application-ID  
Profiles for Voice and Video Streams  
draft-ietf-tsvwg-rsvp-app-id-vv-profiles-02

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 July 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 . . . . .	2
2.	RSVP 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	Application Profiles . . . . .	7
5.1.1	Broadcast Profiles IANA Registry . . . . .	8
5.1.2	Realtime-Interactive Profiles IANA Registry . . . . .	8
5.1.3	Multimedia-Conferencing Profiles IANA Registry . . . . .	9
5.1.4	Multimedia-Streaming Profiles IANA Registry . . . . .	10
5.1.5	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

## 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 is currently being defined so as to establish the type of traffic contained in 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, that RFC does not allow the flexibility of having different domains choosing the DSCP value for the traffic classes 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.

## 1.1 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 [RFC 2119].

## 2. RSVP 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 uses the existing P-type "AUTH\_APP" for application traffic class.

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.

This document creates 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

```
AUTH_APP, POLICY_LOCATOR, ASCII_DN,  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=broadcast.video.surveillance, VER="
```

[Editor's Note: "rfcXXXX" will be replaced with the RFC number assigned to the [ID-TRAF-CLASS] reference. This 'note' should be removed during the RFC-Editor review process.]

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.

This document creates Realtime Interactive profiles for

- Realtime-Interactive Gaming
- Realtime-Interactive Remote-Desktop
- Realtime-Interactive Virtualized-Desktop

Here is the profile for identifying Realtime-Interactive Gaming

```
AUTH_APP, POLICY_LOCATOR, ASCII_DN,  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.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.

This document creates Multimedia-Conferencing profiles for

- Multimedia-Conferencing presentation-data
- Multimedia-Conferencing presentation-video
- Multimedia-Conferencing presentation-audio
- Multimedia-Conferencing application-sharing
- Multimedia-Conferencing whiteboarding

Here is the profile for identifying Multimedia-Conferencing Application-sharing

```
AUTH_APP, POLICY_LOCATOR, ASCII_DN,  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.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...

This document creates Multimedia Streaming profiles for

- Multimedia-Streaming multiplex
- Multimedia-Streaming webcast

Here is the profile for identifying Multimedia Streaming webcast

```
AUTH_APP, POLICY_LOCATOR, ASCII_DN,  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.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.

This document creates 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:

```
AUTH_APP, POLICY_LOCATOR, ASCII_DN,  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.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 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]
```

```
[Editor's Note: "rfcXXXX" will be replaced with the RFC number  
assigned to the [ID-TRAF-CLASS] reference. This  
'note' should be removed during the RFC-Editor  
review process.]
```

##### 5.1.1 Broadcast Profiles IANA Registry

###### Broadcast Audio IPTV Profile

```
P-type = AUTH_APP  
A-type = POLICY_LOCATOR  
Sub-type = ASCII_DN  
Conformant policy locator =  
    "GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
    APP=broadcast.audio.iptv, VER="
```

Reference: [this document]

## Broadcast Video IPTV Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=broadcast.video.iptv, VER="

Reference: [this document]

## Broadcast Audio Live-events Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=broadcast.audio.live-events, VER="

Reference: [this document]

## Broadcast Video Live-events Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=broadcast.video.live-events, VER="

Reference: [this document]

## Broadcast Audio-Surveillance Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=broadcast.audio.surveillance, VER="

Reference: [this document]

## Broadcast Video-Surveillance Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=broadcast.video.surveillance, VER="

Reference: [this document]

## 5.1.2 Realtime-Interactive Profiles IANA Registry

## Realtime-Interactive Gaming Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP= realtime-interactive.gaming, VER="

Reference: [this document]

#### Real-time Interactive Remote-Desktop Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=realtime-interactive.remote-desktop, VER="

Reference: [this document]

#### Real-time Interactive Virtualized-Desktop Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=realtime-interactive.  
remote-desktop.virtual, VER="

Reference: [this document]

#### Real-time Interactive Telemetry Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=realtime-interactive.telemetry, VER="

Reference: [this document]

### 5.1.3 Multimedia-Conferencing Profiles IANA Registry

#### Multimedia-Conferencing Presentation-Data Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP= multimedia-conferencing.presentation-data,  
VER="

Reference: [this document]

#### Multimedia-Conferencing Presentation-Video Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,



APP= multimedia-conferencing.presentation-video,  
VER="

Reference: [this document]

#### Multimedia-Conferencing Presentation-Audio Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP= multimedia-conferencing.presentation-audio,  
VER="

Reference: [this document]

#### Multimedia-Conferencing Application-Sharing Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP= multimedia-conferencing.application-sharing,  
VER="

Reference: [this document]

#### Multimedia-Conferencing Whiteboarding Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP= multimedia-conferencing.whiteboarding, VER="

Reference: [this document]

### 5.1.4 Multimedia-Streaming Profiles IANA Registry

#### Multimedia-Streaming Multiplex Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=multimedia-streaming.multiplex, VER="

Reference: [this document]

#### Multimedia-Streaming Webcast Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=multimedia-streaming.webcast, VER="

Reference: [this document]

#### 5.1.5 Conversational Profiles IANA Registry

##### Conversational Audio Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=conversational.audio, VER="

Reference: [this document]

##### Conversational Audio Admitted Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=conversational.audio.aq:admitted, VER="

Reference: [this document]

##### Conversational Video Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=conversational.video, VER="

Reference: [this document]

##### Conversational Video Admitted Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=conversational.video.aq:admitted, VER="

Reference: [this document]

##### Conversational Audio Avconf Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=conversational.audio.avconf, VER="

Reference: [this document]

##### Conversational Audio Avconf Admitted Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
    APP=conversational.audio.avconf.aq:admitted,  
    VER="

Reference: [this document]

Conversational Video Avconf Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
    APP=conversational.video.avconf, VER="

Reference: [this document]

Conversational Video Avconf Admitted Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
    APP=conversational.video.avconf.aq:admitted,  
    VER="

Reference: [this document]

Conversational Audio Immersive Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
    APP=conversational.audio.immersive, VER="

Reference: [this document]

Conversational Audio Immersive Admitted Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
    APP=conversational.audio.immersive.aq:admitted,  
    VER="

Reference: [this document]

Conversational Video Immersive Profile

P-type = AUTH\_APP  
A-type = POLICY\_LOCATOR  
Sub-type = ASCII\_DN  
Conformant policy locator =  
    "GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,

APP=conversational.video.immersive, VER="

Reference: [this document]

Conversational Video Immersive Admitted Profile

P-type = AUTH\_APP

A-type = POLICY\_LOCATOR

Sub-type = ASCII\_DN

Conformant policy locator =

"GUID=http://www.rfc-editor.org/rfc/rfcXXXX.txt,  
APP=conversational.video.immersive.aq:admitted,  
VER="

Reference: [this document]

## 6. Acknowledgments

To Francois Le Faucheur, Paul Jones, Ken Carlberg, Georgios Karagiannis and Glen Lavers for their helpful comments, document reviews and encouragement.

## 7. References

### 7.1. 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, Feb 2013

## 7.2. Informative References

[RFC4594] J. Babiarez, 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

[Editor's Note: this appendix should be removed in the RFC-Editor's process.]

### A.1 - Changes from WG version -00 to WG version -01

The following changes were made in this version:

- corrected nits
- globally replaced GUID link from the MMUSIC Trafficclass ID to the future RFC of that document.
- added profiles for presentation-video and presentation-audio

### A.2 - Changes from Individual -04 to WG version -00

The following changes were made in this version:

- changed P-Type from APP\_TC back to AUTH\_APP, which is already defined.
- fixed nits and inconsistencies

#### A.3 - 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.4 - 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-03.txt
- changed 'desktop' adjective to 'avconf' to keep in alignment with [ID-TRAF-CLASS]
- Have a complete IANA Registry proposal for each application-ID discussed in this draft.
- General text clean-up of the draft.

Internet Engineering Task Force  
Internet-Draft  
Intended status: Experimental  
Expires: August 14, 2014

Georgios Karagiannis  
University of Twente  
Anurag Bhargava  
Cisco Systems, Inc.  
February 14, 2014

Generic Aggregation of Resource ReSerVation Protocol (RSVP)  
for IPv4 And IPv6 Reservations over PCN domains  
draft-ietf-tsvwg-rsvp-pcn-08

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 August 14, 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.

## 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 . . . . .	4
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
2.13. Message Integrity and Node Authentication . . . . .	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 . . . . .	23
7. Acknowledgments . . . . .	24
8. Normative References . . . . .	24
9. Informative References . . . . .	25
10. Appendix A: Example Signaling Flow . . . . .	25
11. Authors' Address . . . . .	28

## 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.

### 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 Diffserv region or RSVP aware. 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 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):

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.

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 This is an RSVP reservation such that:

- (i) corresponding RSVP Path messages are initiated upstream of the Aggregator and terminated downstream of the Deaggregator, and
- (ii) corresponding RSVP Resv messages are initiated downstream of the Deaggregator and terminated upstream of the Aggregator, and
- (iii) this RSVP reservation is aggregated over an Ingress Egress Aggregate (IEA) between the Aggregator and Deaggregator.

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.

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 for the PCN-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

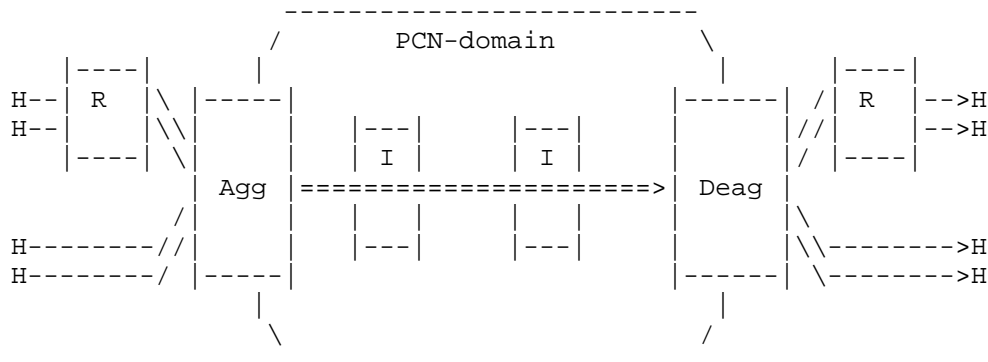
This document is organized as follows. Section 2 gives an overview of RSVP extensions and operations. The elements of the used procedures are specified in Section 3. Section 4 describes the protocol elements. The security considerations are given in section 5 and the IANA considerations are provided 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]. Furthermore, it is considered that by configuration the PCN-interior-nodes do not intercept (nor process) RSVP messages associated with generic aggregated reservation [RFC4860], or with end to end RSVP reservations [RFC2205]. 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].

## 2.13. Message Integrity and Node Authentication

To comply with this specification, for message integrity and node authentication, the same methods MUST be used as the ones described in Section 4 of [RFC4860] and [RFC5559].

## 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 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 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 = (to be replaced by IANA) (PCN)  
 C-Type = RSVP-AGGREGATE-IPv4-PCN-request (to be replaced by IANA)

```

+-----+-----+-----+-----+
|      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 = (to be replaced by IANA) (PCN)  
 C-Type = RSVP-AGGREGATE-IPv6-PCN-request (to be replaced by IANA)

```

+-----+-----+-----+-----+
|      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 = (to be replaced by IANA) (PCN)  
 C-Type = RSVP-AGGREGATE-IPv4-PCN-response (To be replaced by IANA)

```

+-----+-----+-----+-----+
| 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 = (to be replaced by IANA) (PCN)  
 C-Type = RSVP-AGGREGATE-IPv6-PCN-response (to be replaced by IANA)

```

+-----+-----+-----+-----+
|                                     |
+                                     +
| 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 same security considerations specified in [RFC2205], [RFC4230], [RFC4860], [RFC5559] and [RFC6411].

## 6. IANA Considerations

This document makes the following requests to the IANA. IANA needs to modify the RSVP parameters registry, 'Class Names, Class Numbers, and Class Types' subregistry, and add a new Class Number as well as assign 4 new C-Types under this new Class Number, as described below, see Section 4.1:

Class Number	Class Name	Reference
-----	-----	-----
(defined by IANA)	PCN	this document

### Class Types or C-Types:

(defined by IANA)	RSVP-AGGREGATE-IPv4-PCN-request	this document
(defined by IANA)	RSVP-AGGREGATE-IPv6-PCN-request	this document
(defined by IANA)	RSVP-AGGREGATE-IPv4-PCN-response	this document
(defined by IANA)	RSVP-AGGREGATE-IPv6-PCN-response	this document

## 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 and Lixia Zhang for the provided comments. In particular, we would like to thank Francois Le Faucheur for contributing in addition to comments also 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. Babiarez, "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.

[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.

[RFC4230] H. Tschafenig, 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.

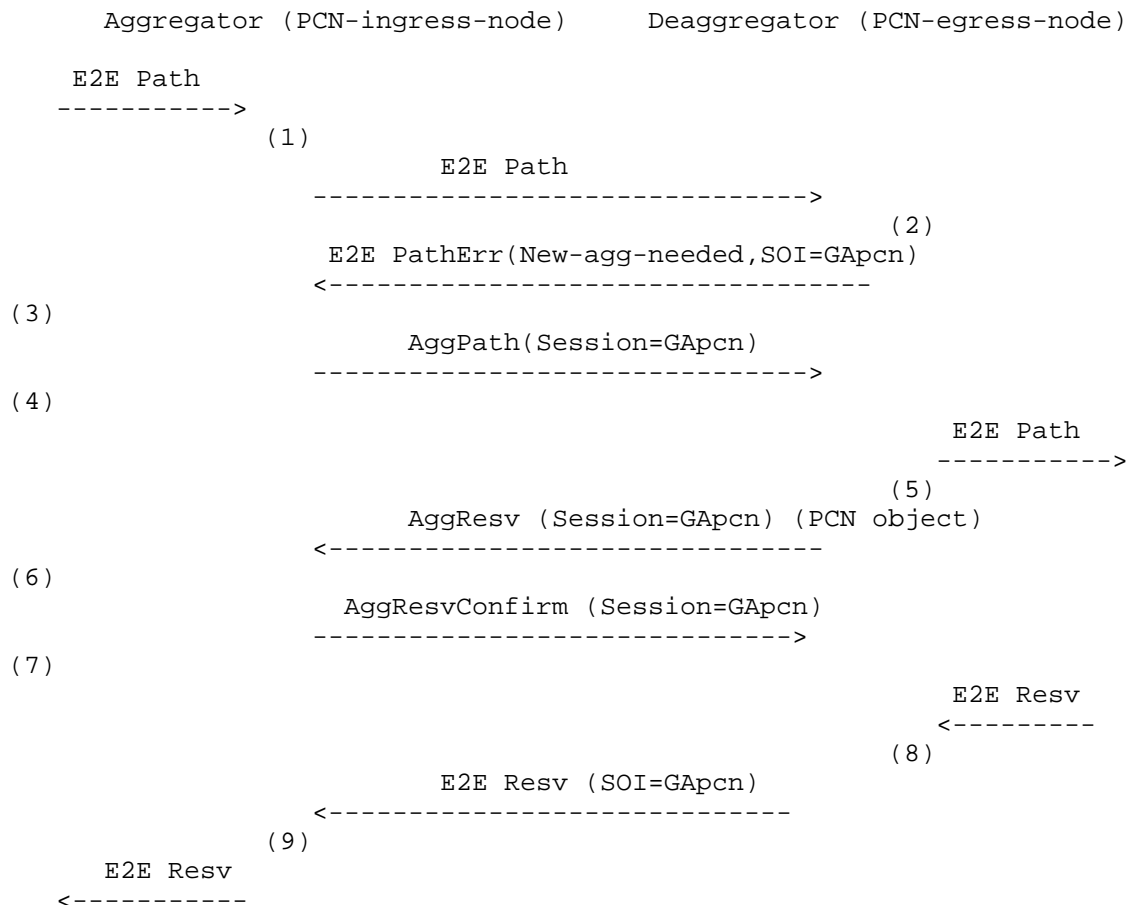
[SIG-NESTED] Baker, F. and P. Bose, "QoS Signaling in a Nested Virtual Private Network", Work in Progress, July 2007.

[RFC2753] Yavatkar, R., D. Pendarakis and R. Guerin, "A Framework for Policy-based Admission Control", January 2000.

## 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 (GApCn).
- (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 (GApCn) 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 (GApCn). 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 GApn (and hence onto the PCN PHB-ID).

- (9) The Aggregator records the mapping of the E2E Resv onto GApn (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  
University of Twente  
P.O. Box 217  
7500 AE Enschede,  
The Netherlands  
EMail: g.karagiannis@utwente.nl

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 11, 2014

M. Tuexen  
Muenster Univ. of Appl. Sciences  
R. Stewart  
Adara Networks  
R. Jesup  
WorldGate Communications  
S. Loreto  
Ericsson  
February 7, 2014

DTLS Encapsulation of SCTP Packets  
draft-ietf-tsvwg-sctp-dtls-encaps-03.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 document specifies how SCTP can be used on top of the Datagram Transport Layer Security (DTLS) protocol.

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 11, 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. Encapsulation and Decapsulation Procedure . . . . .	3
4. DTLS Considerations . . . . .	3
5. SCTP Considerations . . . . .	3
6. IANA Considerations . . . . .	5
7. Security Considerations . . . . .	5
8. Acknowledgments . . . . .	5
9. References . . . . .	5
Authors' Addresses . . . . .	6

## 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 document specifies how SCTP is used on top of the Datagram Transport Layer Security (DTLS) protocol defined in [RFC6347]. This encapsulation is used for example within 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.ietf-rtcweb-data-channel].

### 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.

The DTLS MUST support sending messages larger than the current path MTU. This might result in sending IP level fragmented messages.

## 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.

#### 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

If the dynamic address reconfiguration extension defined in [RFC5061] is used, only wildcard addresses MUST be used in ASCONF chunks.

#### 5.4. SCTP Authentication Extension

The SCTP authentication extension defined in [RFC4895] can be used with DTLS encapsulation, but does not provide any additional benefit.

#### 5.5. Partial Reliability Extension

Partial reliability as defined in [RFC3758] can be used in combination with DTLS encapsulation. It is also possible to use additional PR-SCTP policies.

#### 5.6. Stream Reset Extension

The SCTP stream reset extension defined in [RFC6525] can be used with DTLS encapsulation. It is used to reset streams and add streams during the lifetime of the SCTP association.

### 5.7. Interleaving of Large User Messages

SCTP as defined in [RFC4960] does not support the interleaving of large user messages that need to be fragmented and reassembled by the SCTP layer. The protocol extension defined in [I-D.ietf-tsvwg-sctp-ndata] overcomes this limitation and can be used with DTLS encapsulation.

### 6. IANA Considerations

This document requires no actions from IANA.

### 7. Security Considerations

Security considerations for DTLS are specified in [RFC6347] and for SCTP in [RFC4960], [RFC3758], and [RFC6525]. The combination of SCTP and DTLS introduces no new security considerations.

### 8. Acknowledgments

The authors wish to thank Gorrry Fairhurst for his 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.
- [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.
- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, 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.

## 9.2. Informative References

- [RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, May 2004.
- [RFC4895] Tuexen, M., Stewart, R., Lei, P., and E. Rescorla, "Authenticated Chunks for the Stream Control Transmission Protocol (SCTP)", RFC 4895, August 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.
- [RFC6525] Stewart, R., Tuexen, M., and P. Lei, "Stream Control Transmission Protocol (SCTP) Stream Reconfiguration", RFC 6525, February 2012.
- [I-D.ietf-rtcweb-overview] Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-08 (work in progress), September 2013.
- [I-D.ietf-rtcweb-data-channel] Jesup, R., Loreto, S., and M. Tuexen, "RTCWeb Data Channels", draft-ietf-rtcweb-data-channel-06 (work in progress), October 2013.
- [I-D.ietf-tsvwg-sctp-ndata] Stewart, R., Tuexen, M., Loreto, S., and R. Seggelmann, "A New Data Chunk for Stream Control Transmission Protocol", draft-ietf-tsvwg-sctp-ndata-00 (work in progress), February 2014.

## Authors' Addresses

Michael Tuexen  
Muenster University of Applied Sciences  
Stegerwaldstrasse 39  
48565 Steinfurt  
DE

Email: tuexen@fh-muenster.de



Randall R. Stewart  
Adara Networks  
Chapin, SC 29036  
US

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

Randell Jesup  
WorldGate Communications  
3800 Horizon Blvd, Suite #103  
Trevose, PA 19053-4947  
US

Phone: +1-215-354-5166  
Email: [randell\\_ietf@jesup.org](mailto:randell_ietf@jesup.org)

Salvatore Loreto  
Ericsson  
Hirsalantie 11  
Jorvas 02420  
FI

Email: [Salvatore.Loreto@ericsson.com](mailto:Salvatore.Loreto@ericsson.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: acar@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 Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: March 5, 2018

R. Stewart  
Netflix, Inc.  
M. Tuexen  
Muenster Univ. of Appl. Sciences  
S. Loreto  
Ericsson  
R. Seggelmann  
Metafinanz Informationssysteme GmbH  
September 1, 2017

Stream Schedulers and User Message Interleaving for the Stream Control  
Transmission Protocol  
draft-ietf-tsvwg-sctp-ndata-13.txt

Abstract

The Stream Control Transmission Protocol (SCTP) is a message oriented transport protocol supporting arbitrarily large user messages. This document adds a new chunk to SCTP for carrying payload data. This allows a sender to interleave different user messages that would otherwise result in head of line blocking at the sender. The interleaving of user messages is required for WebRTC Datachannels.

Whenever an SCTP sender is allowed to send user data, it may choose from multiple outgoing SCTP streams. Multiple ways for performing this selection, called stream schedulers, are defined in this document. A stream scheduler can choose to either implement, or not implement, user message interleaving.

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 March 5, 2018.



## Copyright Notice

Copyright (c) 2017 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. Overview . . . . .	3
1.2. Conventions . . . . .	5
2. User Message Interleaving . . . . .	5
2.1. The I-DATA Chunk Supporting User Message Interleaving . .	6
2.2. Procedures . . . . .	8
2.2.1. Negotiation . . . . .	9
2.2.2. Sender Side Considerations . . . . .	9
2.2.3. Receiver Side Considerations . . . . .	10
2.3. Interaction with other SCTP Extensions . . . . .	10
2.3.1. SCTP Partial Reliability Extension . . . . .	10
2.3.2. SCTP Stream Reconfiguration Extension . . . . .	12
3. Stream Schedulers . . . . .	12
3.1. First Come First Served Scheduler (SCTP_SS_FCFS) . . . .	13
3.2. Round Robin Scheduler (SCTP_SS_RR) . . . . .	13
3.3. Round Robin Scheduler per Packet (SCTP_SS_RR_PKT) . . . .	13
3.4. Priority Based Scheduler (SCTP_SS_PRIO) . . . . .	13
3.5. Fair Capacity Scheduler (SCTP_SS_FC) . . . . .	14
3.6. Weighted Fair Queueing Scheduler (SCTP_SS_WFQ) . . . . .	14
4. Socket API Considerations . . . . .	14
4.1. Exposure of the Stream Sequence Number (SSN) . . . . .	14
4.2. SCTP_ASSOC_CHANGE Notification . . . . .	15
4.3. Socket Options . . . . .	15
4.3.1. Enable or Disable the Support of User Message Interleaving (SCTP_INTERLEAVING_SUPPORTED) . . . . .	15
4.3.2. Get or Set the Stream Scheduler (SCTP_STREAM_SCHEDULER) . . . . .	16
4.3.3. Get or Set the Stream Scheduler Parameter (SCTP_STREAM_SCHEDULER_VALUE) . . . . .	17
4.4. Explicit EOR Marking . . . . .	18
5. IANA Considerations . . . . .	18

5.1. I-DATA Chunk . . . . .	18
5.2. I-FORWARD-TSN Chunk . . . . .	19
6. Security Considerations . . . . .	19
7. Acknowledgments . . . . .	20
8. References . . . . .	20
8.1. Normative References . . . . .	20
8.2. Informative References . . . . .	21
Authors' Addresses . . . . .	21

## 1. Introduction

### 1.1. Overview

When SCTP [RFC4960] was initially designed it was mainly envisioned for the transport of small signaling messages. Late in the design stage it was decided to add support for fragmentation and reassembly of larger messages with the thought that someday Session Initiation Protocol (SIP) [RFC3261] style signaling messages may also need to use SCTP and a single Maximum Transmission Unit (MTU) sized message would be too small. Unfortunately this design decision, though valid at the time, did not account for other applications that might send large messages over SCTP. The sending of such large messages over SCTP as specified in [RFC4960] can result in a form of sender side head of line blocking (e.g., when the transmission of a message is blocked from transmission because the sender has started the transmission of another, possibly large, message). This head of line blocking is caused by the use of the Transmission Sequence Number (TSN) for three different purposes:

1. As an identifier for DATA chunks to provide a reliable transfer.
2. As an identifier for the sequence of fragments to allow reassembly.
3. As a sequence number allowing up to  $2^{16} - 1$  Stream Sequence Numbers (SSNs) outstanding.

The protocol requires all fragments of a user message to have consecutive TSNs. This document allows an SCTP sender to interleave different user messages.

This document also defines several stream schedulers for general SCTP associations allowing different relative stream treatments. The stream schedulers may behave differently depending on whether user message interleaving has been negotiated for the association or not.

Figure 1 illustrates the behaviour of a round robin stream scheduler using DATA chunks when three streams with the Stream Identifiers

(SIDs) 0, 1, and 2 are used. Each queue for SID 0 and SID 2 contains a single user message requiring three chunks, the queue for SID 1 contains three user messages each requiring a single chunk. It is shown how these user messages are encapsulated in chunk using TSN 0 to TSN 8. Please note that the use of such a scheduler implies late TSN assignment but it can be used with an [RFC4960] compliant implementation that does not support user message interleaving. Late TSN assignment means that the sender generates chunks from user messages and assigns the TSN as late as possible in the process of sending the user messages.

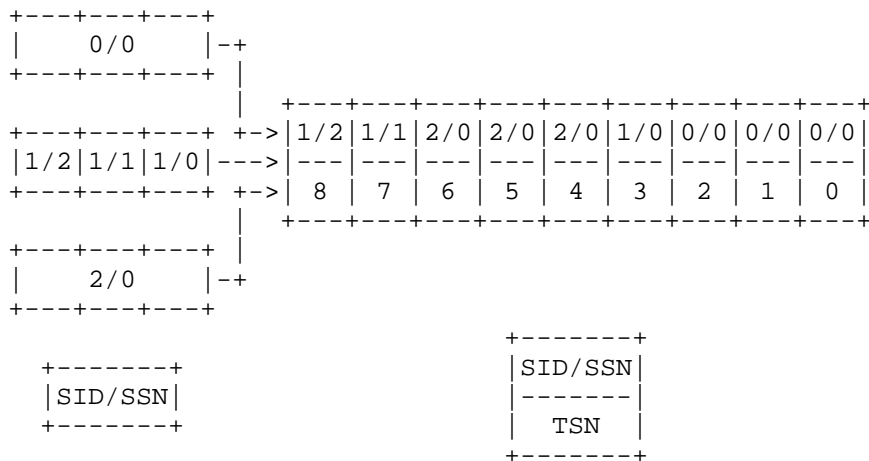


Figure 1: Round Robin Scheduler without User Message Interleaving

This document describes a new chunk carrying payload data called I-DATA. This chunk incorporates the properties of the current SCTP DATA chunk, all the flags and fields except the Stream Sequence Number (SSN), but also adds two new fields in its chunk header, the Fragment Sequence Number (FSN) and the Message Identifier (MID). The FSN is only used for reassembling all fragments having the same MID and ordering property. The TSN is only used for the reliable transfer in combination with Selective Acknowledgment (SACK) chunks.

In addition, the MID is also used for ensuring ordered delivery instead of using the stream sequence number (The I-DATA chunk omits a SSN.).

Figure 2 illustrates the behaviour of an interleaving round robin stream scheduler using I-DATA chunks.

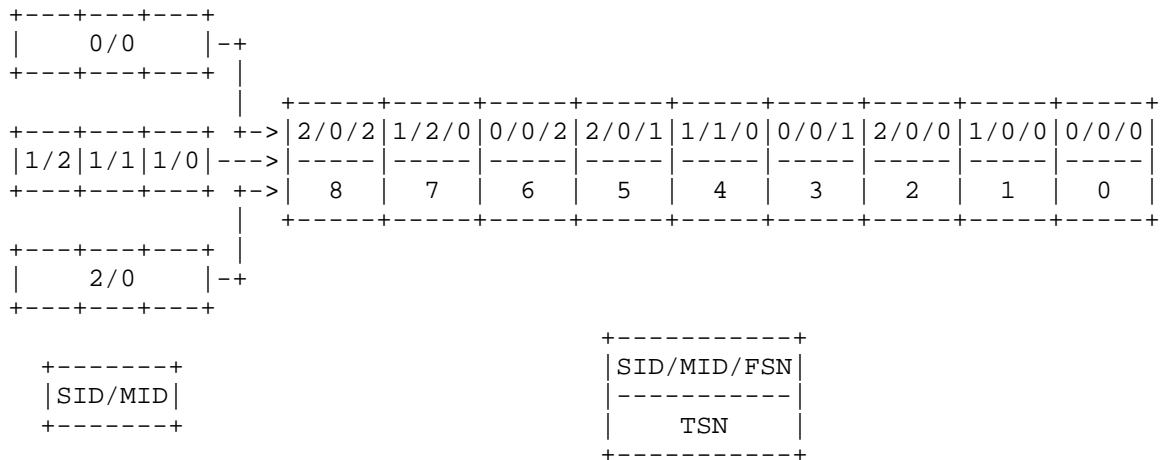


Figure 2: Round Robin Scheduler with User Message Interleaving

The support of the I-DATA chunk is negotiated during the association setup using the Supported Extensions Parameter as defined in [RFC5061]. If I-DATA support has been negotiated for an association, I-DATA chunks are used for all user-messages. DATA chunks are not permitted when I-DATA support has been negotiated. It should be noted that an SCTP implementation supporting I-DATA chunks needs to allow the coexistence of associations using DATA chunks and associations using I-DATA chunks.

In Section 2 this document specifies the user message interleaving by defining the I-DATA chunk, the procedures to use it and its interactions with other SCTP extensions. Multiple stream schedulers are defined in Section 3 followed in Section 4 by describing an extension to the socket API for using what is specified in this document.

## 1.2. 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].

## 2. User Message Interleaving

The protocol mechanisms described in this document allow the interleaving of user messages sent on different streams. They do not support the interleaving of multiple messages (ordered or unordered) sent on the same stream.

The interleaving of user messages is required for WebRTC Datachannels as specified in [I-D.ietf-rtcweb-data-channel].

An SCTP implementation supporting user message interleaving is REQUIRED to support the coexistence of associations using DATA chunks and associations using I-DATA chunks. If an SCTP implementation supports user message interleaving and the Partial Reliability extension described in [RFC3758] or the Stream Reconfiguration Extension described in [RFC6525], it is REQUIRED to implement the corresponding changes specified in Section 2.3.

## 2.1. The I-DATA Chunk Supporting User Message Interleaving

The following Figure 3 shows the new I-DATA chunk allowing user message interleaving.

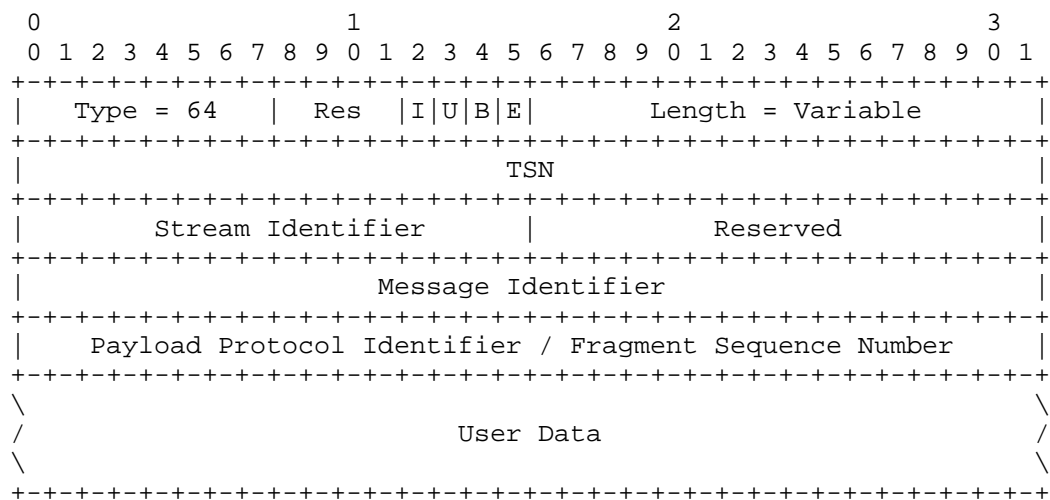


Figure 3: I-DATA chunk format

The only differences between the I-DATA chunk in Figure 3 and the DATA chunk defined in [RFC4960] and [RFC7053] are the addition of the new Message Identifier (MID) and the new Fragment Sequence Number (FSN) and the removal of the Stream Sequence Number (SSN). The Payload Protocol Identifier (PPID) already defined for DATA chunks in [RFC4960] and the new FSN are stored at the same location of the packet using the B bit to determine which value is stored at the location. The length of the I-DATA chunk header is 20 bytes, which is 4 bytes more than the length of the DATA chunk header defined in [RFC4960] and [RFC7053].

The old fields are:

Res: 4 bits

These bits are reserved. They MUST be set to 0 by the sender and MUST be ignored by the receiver.

I bit: 1 bit

The (I)mmmediate Bit, if set, indicates that the receiver SHOULD NOT delay the sending of the corresponding SACK chunk. Same as the I bit for DATA chunks as specified in [RFC7053].

U bit: 1 bit

The (U)nordered bit, if set, indicates the user message is unordered. Same as the U bit for DATA chunks as specified in [RFC4960].

B bit: 1 bit

The (B)eginning fragment bit, if set, indicates the first fragment of a user message. Same as the B bit for DATA chunks as specified in [RFC4960].

E bit: 1 bit

The (E)nding fragment bit, if set, indicates the last fragment of a user message. Same as the E bit for DATA chunks as specified in [RFC4960].

Length: 16 bits (unsigned integer)

This field indicates the length of the DATA chunk in bytes from the beginning of the type field to the end of the User Data field excluding any padding. Similar to the Length for DATA chunks as specified in [RFC4960].

TSN: 32 bits (unsigned integer)

This value represents the TSN for this I-DATA chunk. Same as the TSN for DATA chunks as specified in [RFC4960].

Stream Identifier: 16 bits (unsigned integer)

Identifies the stream to which the user data belongs. Same as the Stream Identifier for DATA chunks as specified in [RFC4960].

The new fields are:

Reserved: 16 bits (unsigned integer)

This field is reserved. It MUST be set to 0 by the sender and MUST be ignored by the receiver.

Message Identifier (MID): 32 bits (unsigned integer)

The MID is the same for all fragments of a user message, it is used to determine which fragments (enumerated by the FSN) belong to the same user message. For ordered user messages, the MID is

also used by the SCTP receiver to deliver the user messages in the correct order to the upper layer (similar to the SSN of the DATA chunk defined in [RFC4960]). The sender uses for each outgoing stream two counters, one for ordered messages, one for unordered messages. All of these counters are independent and initially 0. They are incremented by 1 for each user message. Please note that the serial number arithmetic defined in [RFC1982] using `SERIAL_BITS = 32` applies. Therefore, the sender MUST NOT have more than  $2^{31} - 1$  ordered messages for each outgoing stream in flight and MUST NOT have more than  $2^{31} - 1$  unordered messages for each outgoing stream in flight. A message is considered in flight, if at least one of its I-DATA chunks is not acknowledged in a non-renegable way (i.e. not acknowledged by the cumulative TSN Ack). Please note that the MID is in "network byte order", a.k.a. Big Endian.

Payload Protocol Identifier (PPID) / Fragment Sequence Number (FSN):  
32 bits (unsigned integer)

If the B bit is set, this field contains the PPID of the user message. Note that in this case, this field is not touched by an SCTP implementation; therefore, its byte order is not necessarily in network byte order. The upper layer is responsible for any byte order conversions to this field, similar to the PPID of DATA chunks. In this case the FSN is implicitly considered to be 0. If the B bit is not set, this field contains the FSN. The FSN is used to enumerate all fragments of a single user message, starting from 0 and incremented by 1. The last fragment of a message MUST have the E bit set. Note that the FSN MAY wrap completely multiple times allowing arbitrarily large user messages. For the FSN the serial number arithmetic defined in [RFC1982] applies with `SERIAL_BITS = 32`. Therefore, a sender MUST NOT have more than  $2^{31} - 1$  fragments of a single user message in flight. A fragment is considered in flight, if it is not acknowledged in a non-renegable way. Please note that the FSN is in "network byte order", a.k.a. Big Endian.

## 2.2. Procedures

This subsection describes how the support of the I-DATA chunk is negotiated and how the I-DATA chunk is used by the sender and receiver.

The handling of the I bit for the I-DATA chunk corresponds to the handling of the I bit for the DATA chunk described in [RFC7053].

### 2.2.1. Negotiation

An SCTP end point indicates user message interleaving support by listing the I-DATA Chunk within the Supported Extensions Parameter as defined in [RFC5061]. User message interleaving has been negotiated for an association if both end points have indicated I-DATA support.

If user message interleaving support has been negotiated for an association, I-DATA chunks MUST be used for all user messages and DATA-chunks MUST NOT be used. If user message interleaving support has not been negotiated for an association, DATA chunks MUST be used for all user messages and I-DATA chunks MUST NOT be used.

An end point implementing the socket API specified in [RFC6458] MUST NOT indicate user message interleaving support unless the user has requested its use (e.g. via the socket API, see Section 4.3). This constraint is made since the usage of this chunk requires that the application is capable of handling interleaved messages upon reception within an association. This is not the default choice within the socket API (see the `SCTP_FRAGMENT_INTERLEAVE` socket option in Section 8.1.20 of [RFC6458]) thus the user MUST indicate to the SCTP implementation its support for receiving completely interleaved messages.

Note that stacks that do not implement [RFC6458] may use other methods to indicate interleaved message support and thus indicate the support of user message interleaving. The crucial point is that the SCTP stack MUST know that the application can handle interleaved messages before indicating the I-DATA support.

### 2.2.2. Sender Side Considerations

The sender side usage of the I-DATA chunk is quite simple. Instead of using the TSN for fragmentation purposes, the sender uses the new FSN field to indicate which fragment number is being sent. The first fragment MUST have the B bit set. The last fragment MUST have the E bit set. All other fragments MUST NOT have the B bit or E bit set. All other properties of the existing SCTP DATA chunk also apply to the I-DATA chunk, i.e. congestion control as well as receiver window conditions MUST be observed as defined in [RFC4960].

Note that the usage of this chunk implies the late assignment of the actual TSN to any chunk being sent. Each I-DATA chunk uses a single TSN. This way messages from other streams may be interleaved with the fragmented message. Please note that this is the only form of interleaving support. For example, it is not possible to interleave multiple ordered or unordered user messages from the same stream.



The sender MUST NOT process (move user data into I-DATA chunks and assign a TSN to it) more than one user message in any given stream at any time. At any time, a sender MAY process multiple user messages, each of them on different streams.

The sender MUST assign TSNs to I-DATA chunks in a way that the receiver can make progress. One way to achieve this is to assign a higher TSN to the later fragments of a user message and send out the I-DATA chunks such that the TSNs are in sequence.

### 2.2.3. Receiver Side Considerations

Upon reception of an SCTP packet containing an I-DATA chunk whose user message needs to be reassembled, the receiver MUST first use the SID to identify the stream, consider the U bit to determine if it is part of an ordered or unordered message, find the user message identified by the MID and finally use the FSN for reassembly of the message and not the TSN. The receiver MUST NOT make any assumption about the TSN assignments of the sender. Note that a non-fragmented message is indicated by the fact that both the E and B bits are set. A message (either ordered or unordered) may be identified as being fragmented whose E and B bits are not both set.

If I-DATA support has been negotiated for an association, the reception of a DATA chunk is a violation of the above rules and therefore the receiver of the DATA chunk MUST abort the association by sending an ABORT chunk. The ABORT chunk MAY include the 'Protocol Violation' error cause. The same applies if I-DATA support has not been negotiated for an association and an I-DATA chunk is received.

## 2.3. Interaction with other SCTP Extensions

The usage of the I-DATA chunk might interfere with other SCTP extensions. Future SCTP extensions MUST describe if and how they interfere with the usage of I-DATA chunks. For the SCTP extensions already defined when this document was published, the details are given in the following subsections.

### 2.3.1. SCTP Partial Reliability Extension

When the SCTP extension defined in [RFC3758] is used in combination with the user message interleaving extension, the new I-FORWARD-TSN chunk MUST be used instead of the FORWARD-TSN chunk. The difference between the FORWARD-TSN and the I-FORWARD-TSN chunk is that the 16-bit Stream Sequence Number (SSN) has been replaced by the 32-bit Message Identifier (MID) and the largest skipped MID can also be provided for unordered messages. Therefore, the principle applied to

ordered message when using FORWARD-TSN chunks is applied to ordered and unordered messages when using I-FORWARD-TSN chunks.

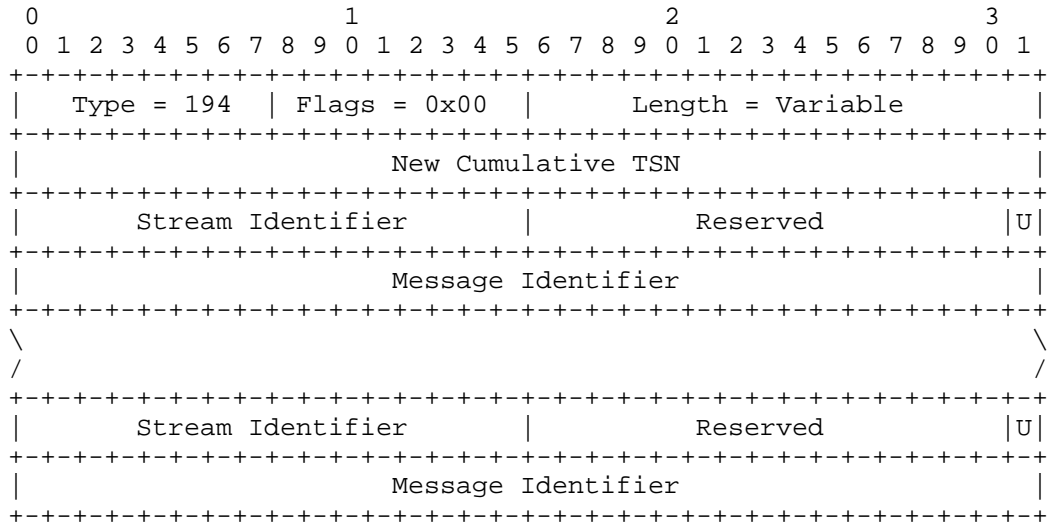


Figure 4: I-FORWARD-TSN chunk format

The old fields are:

Flags: 8-bits (unsigned integer)

These bits are reserved. They MUST be set to 0 by the sender and MUST be ignored by the receiver. Same as the Flags for FORWARD TSN chunks as specified in [RFC3758].

Length: 16-bits (unsigned integer)

This field holds the length of the chunk. Similar to the Length for FORWARD TSN chunks as specified in [RFC3758].

New Cumulative TSN: 32-bits (unsigned integer)

This indicates the new cumulative TSN to the data receiver. Same as the New Cumulative TSN for FORWARD TSN chunks as specified in [RFC3758].

The new fields are:

Stream Identifier (SID): 16-bits (unsigned integer)

This field holds the stream number this entry refers to.

Reserved: 15 bits

This field is reserved. It MUST be set to 0 by the sender and MUST be ignored by the receiver.

U bit: 1 bit

The U bit specifies if the Message Identifier of this entry refers to unordered messages (U bit is set) or ordered messages (U bit is not set).

Message Identifier (MID): 32 bits (unsigned integer)

This field holds the largest Message Identifier for ordered or unordered messages indicated by the U bit that was skipped for the stream specified by the Stream Identifier. For ordered messages this is similar to the FORWARD-TSN chunk, just replacing the 16-bit SSN by the 32-bit MID.

Support for the I-FORWARD-TSN chunk is negotiated during the SCTP association setup via the Supported Extensions Parameter as defined in [RFC5061]. Only if both end points indicated their support of user message interleaving and the I-FORWARD-TSN chunk, the partial reliability extension is negotiated and can be used in combination with user message interleaving.

The FORWARD-TSN chunk MUST be used in combination with the DATA chunk and MUST NOT be used in combination with the I-DATA chunk. The I-FORWARD-TSN chunk MUST be used in combination with the I-DATA chunk and MUST NOT be used in combination with the DATA chunk.

If I-FORWARD-TSN support has been negotiated for an association, the reception of a FORWARD-TSN chunk is a violation of the above rules and therefore the receiver of the FORWARD-TSN chunk MUST abort the association by sending an ABORT chunk. The ABORT chunk MAY include the 'Protocol Violation' error cause. The same applies if I-FORWARD-TSN support has not been negotiated for an association and a FORWARD-TSN chunk is received.

### 2.3.2. SCTP Stream Reconfiguration Extension

When an association resets the SSN using the SCTP extension defined in [RFC6525], the two counters (one for the ordered messages, one for the unordered messages) used for the MIDs MUST be reset to 0.

Since most schedulers, especially all schedulers supporting user message interleaving, require late TSN assignment, it should be noted that the implementation of [RFC6525] needs to handle this.

## 3. Stream Schedulers

This section defines several stream schedulers. The stream schedulers may behave differently depending on whether user message interleaving has been negotiated for the association or not. An implementation MAY implement any subset of them. If the

implementation is used for WebRTC Datachannels as specified in [I-D.ietf-rtcweb-data-channel] it MUST implement the Weighted Fair Queueing Scheduler defined in Section 3.6.

The selection of the stream scheduler is done at the sender side. There is no mechanism provided for signalling the stream scheduler being used to the receiver side or even let the receiver side influence the selection of the stream scheduler used at the sender side.

### 3.1. First Come First Served Scheduler (SCTP\_SS\_FCFS)

The simple first-come, first-served scheduler of user messages is used. It just passes through the messages in the order in which they have been delivered by the application. No modification of the order is done at all. The usage of user message interleaving does not affect the sending of the chunks, except that I-DATA chunks are used instead of DATA chunks.

### 3.2. Round Robin Scheduler (SCTP\_SS\_RR)

When not using user message interleaving, this scheduler provides a fair scheduling based on the number of user messages by cycling around non-empty stream queues. When using user message interleaving, this scheduler provides a fair scheduling based on the number of I-DATA chunks by cycling around non-empty stream queues.

### 3.3. Round Robin Scheduler per Packet (SCTP\_SS\_RR\_PKT)

This is a round-robin scheduler, which only switches streams when starting to fill a new packet. It bundles only DATA or I-DATA chunks referring to the same stream in a packet. This scheduler minimizes head-of-line blocking when a packet is lost because only a single stream is affected.

### 3.4. Priority Based Scheduler (SCTP\_SS\_PRIO)

Scheduling of user messages with strict priorities is used. The priority is configurable per outgoing SCTP stream. Streams having a higher priority will be scheduled first and when multiple streams have the same priority, the scheduling between them is implementation dependent. When using user message interleaving, the sending of large lower priority user messages will not delay the sending of higher priority user messages.

### 3.5. Fair Capacity Scheduler (SCTP\_SS\_FC)

A fair capacity distribution between the streams is used. This scheduler considers the lengths of the messages of each stream and schedules them in a specific way to maintain an equal capacity for all streams. The details are implementation dependent. Using user message interleaving allows for a better realization of the fair capacity usage.

### 3.6. Weighted Fair Queueing Scheduler (SCTP\_SS\_WFQ)

A weighted fair queueing scheduler between the streams is used. The weight is configurable per outgoing SCTP stream. This scheduler considers the lengths of the messages of each stream and schedules them in a specific way to use the capacity according to the given weights. If the weight of stream S1 is n times the weight of stream S2, the scheduler should assign to stream S1 n times the capacity it assigns to stream S2. The details are implementation dependent. Using user message interleaving allows for a better realization of the capacity usage according to the given weights.

This scheduler in combination with user message interleaving is used for WebRTC Datachannels as specified in [I-D.ietf-rtcweb-data-channel].

## 4. Socket API Considerations

This section describes how the socket API defined in [RFC6458] is extended to allow applications to use the extension described in this document.

Please note that this section is informational only.

### 4.1. Exposure of the Stream Sequence Number (SSN)

The socket API defined in [RFC6458] defines several structures in which the SSN of a received user message is exposed to the application. The list of these structures includes:

```
struct sctp_sndrcvinfo
    Specified in Section 5.3.2 SCTP Header Information Structure
    (SCTP_SNDRCV) of [RFC6458] and marked as deprecated.

struct sctp_extrcvinfo
    Specified in Section 5.3.3 Extended SCTP Header Information
    Structure (SCTP_EXTRCV) of [RFC6458] and marked as deprecated.

struct sctp_rcvinfo
```

Specified in Section 5.3.5 SCTP Receive Information Structure (SCTP\_RCVINFO) of [RFC6458].

If user message interleaving is used, the lower order 16 bits of the MID are used as the SSN when filling out these structures.

#### 4.2. SCTP\_ASSOC\_CHANGE Notification

When an SCTP\_ASSOC\_CHANGE notification (specified in Section 6.1.1 of [RFC6458]) is delivered indicating a sac\_state of SCTP\_COMM\_UP or SCTP\_RESTART for an SCTP association where both peers support the I-DATA chunk, SCTP\_ASSOC\_SUPPORTS\_INTERLEAVING should be listed in the sac\_info field.

#### 4.3. Socket Options

option name	data type	get	set
SCTP_INTERLEAVING_SUPPORTED	struct sctp_assoc_value	X	X
SCTP_STREAM_SCHEDULER	struct sctp_assoc_value	X	X
SCTP_STREAM_SCHEDULER_VALUE	struct sctp_stream_value	X	X

##### 4.3.1. Enable or Disable the Support of User Message Interleaving (SCTP\_INTERLEAVING\_SUPPORTED)

This socket option allows the enabling or disabling of the negotiation of user message interleaving support for future associations. For existing associations it allows to query whether user message interleaving support was negotiated or not on a particular association.

This socket option uses IPPROTO\_SCTP as its level and SCTP\_INTERLEAVING\_SUPPORTED as its name. It can be used with getsockopt() and setsockopt(). The socket option value uses the following structure defined in [RFC6458]:

```
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, this parameter indicates upon which association the user is performing an action. The special

sctp\_assoc\_t Sctp\_FUTURE\_ASSOC can also be used, it is an error to use Sctp\_{CURRENT|ALL}\_ASSOC in assoc\_id.

assoc\_value: A non-zero value encodes the enabling of user message interleaving whereas a value of 0 encodes the disabling of user message interleaving.

sctp\_opt\_info() needs to be extended to support Sctp\_INTERLEAVING\_SUPPORTED.

An application using user message interleaving should also set the fragment interleave level to 2 by using the Sctp\_FRAGMENT\_INTERLEAVE socket option specified in Section 8.1.20 of [RFC6458]. This allows the interleaving of user messages from different streams. Please note that it does not allow the interleaving of user messages (ordered or unordered) on the same stream. Failure to set this option can possibly lead to application deadlock. Some implementations might therefore put some restrictions on setting combinations of these values. Setting the interleaving level to at least 2 before enabling the negotiation of user message interleaving should work on all platforms. Since the default fragment interleave level is not 2, user message interleaving is disabled per default.

#### 4.3.2. Get or Set the Stream Scheduler (Sctp\_STREAM\_SCHEDULER)

A stream scheduler can be selected with the Sctp\_STREAM\_SCHEDULER option for setsockopt(). The struct sctp\_assoc\_value is used to specify the association for which the scheduler should be changed and the value of the desired algorithm.

The definition of struct sctp\_assoc\_value is the same as in [RFC6458]:

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

assoc\_id: Holds the identifier for the association of which the scheduler should be changed. The special Sctp\_{FUTURE|CURRENT|ALL}\_ASSOC can also be used. This parameter is ignored for one-to-one style sockets.

assoc\_value: This specifies which scheduler is used. The following constants can be used:

Sctp\_SS\_DEFAULT: The default scheduler used by the Sctp implementation. Typical values are Sctp\_SS\_FCFS or Sctp\_SS\_RR.

SCTP\_SS\_FCFS: Use the scheduler specified in Section 3.1.

SCTP\_SS\_RR: Use the scheduler specified in Section 3.2.

SCTP\_SS\_RR\_PKT: Use the scheduler specified in Section 3.3.

SCTP\_SS\_PRIO: Use the scheduler specified in Section 3.4. The priority can be assigned with the `sctp_stream_value` struct. The higher the assigned value, the lower the priority, that is the default value 0 is the highest priority and therefore the default scheduling will be used if no priorities have been assigned.

SCTP\_SS\_FB: Use the scheduler specified in Section 3.5.

SCTP\_SS\_WFQ: Use the scheduler specified in Section 3.6. The weight can be assigned with the `sctp_stream_value` struct.

`sctp_opt_info()` needs to be extended to support `SCTP_STREAM_SCHEDULER`.

#### 4.3.3. Get or Set the Stream Scheduler Parameter (`SCTP_STREAM_SCHEDULER_VALUE`)

Some schedulers require additional information to be set for individual streams as shown in the following table:

name	per stream info
SCTP_SS_DEFAULT	n/a
SCTP_SS_FCFS	no
SCTP_SS_RR	no
SCTP_SS_RR_PKT	no
SCTP_SS_PRIO	yes
SCTP_SS_FB	no
SCTP_SS_WFQ	yes

This is achieved with the `SCTP_STREAM_SCHEDULER_VALUE` option and the corresponding struct `sctp_stream_value`. The definition of struct `sctp_stream_value` is as follows:

```
struct sctp_stream_value {
    sctp_assoc_t assoc_id;
    uint16_t stream_id;
    uint16_t stream_value;
};
```



assoc\_id: Holds the identifier for the association of which the scheduler should be changed. The special SCTP\_{FUTURE|CURRENT|ALL}\_ASSOC can also be used. This parameter is ignored for one-to-one style sockets.

stream\_id: Holds the stream id of the stream for which additional information has to be provided.

stream\_value: The meaning of this field depends on the scheduler specified. It is ignored when the scheduler does not need additional information.

sctp\_opt\_info() needs to be extended to support SCTP\_STREAM\_SCHEDULER\_VALUE.

#### 4.4. Explicit EOR Marking

Using explicit End of Record (EOR) marking for an SCTP association supporting user message interleaving allows the user to interleave the sending of user messages on different streams.

#### 5. 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 types and the chunk flags are tentative and to be confirmed by IANA.

]

This document (RFCXXXX) is the reference for all registrations described in this section.

Two new chunk types have to be assigned by IANA.

##### 5.1. I-DATA Chunk

IANA should assign the chunk type for this chunk from the pool of chunks with the upper two bits set to '01'. This requires an additional line in the "Chunk Types" registry for SCTP:

ID Value	Chunk Type	Reference
64	Payload Data supporting Interleaving (I-DATA)	[RFCXXXX]

The registration table as defined in [RFC6096] for the chunk flags of this chunk type is initially given by the following table:

Chunk Flag Value	Chunk Flag Name	Reference
0x01	E bit	[RFCXXXX]
0x02	B bit	[RFCXXXX]
0x04	U bit	[RFCXXXX]
0x08	I bit	[RFCXXXX]
0x10	Unassigned	
0x20	Unassigned	
0x40	Unassigned	
0x80	Unassigned	

## 5.2. I-FORWARD-TSN Chunk

IANA should assign the chunk type for this chunk from the pool of chunks with the upper two bits set to '11'. This requires an additional line in the "Chunk Types" registry for SCTP:

ID Value	Chunk Type	Reference
194	I-FORWARD-TSN	[RFCXXXX]

The registration table as defined in [RFC6096] for the chunk flags of this chunk type is initially empty.

## 6. Security Considerations

This document does not add any additional security considerations in addition to the ones given in [RFC4960] and [RFC6458].

It should be noted that the application has to consent that it is willing to do the more complex reassembly support required for user message interleaving. When doing so, an application has to provide a reassembly buffer for each incoming stream. It has to protect itself against these buffers taking too many resources. If user message

interleaving is not used, only a single reassembly buffer needs to be provided for each association. But the application has to protect itself for excessive resource usages there too.

## 7. Acknowledgments

The authors wish to thank Benoit Claise, Julian Cordes, Spencer Dawkins, Gorry Fairhurst, Lennart Grahl, Christer Holmberg, Mirja Kuehlewind, Marcelo Ricardo Leitner, Karen E. Egede Nielsen, Maksim Proshin, Eric Rescorla, Irene Ruengeler, Felix Weinrank, Michael Welzl, Magnus Westerlund, and Lixia Zhang for their invaluable comments.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 644334 (NEAT). The views expressed are solely those of the author(s).

## 8. References

### 8.1. Normative References

- [RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982, DOI 10.17487/RFC1982, August 1996, <<https://www.rfc-editor.org/info/rfc1982>>.
- [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>>.
- [RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, DOI 10.17487/RFC3758, May 2004, <<https://www.rfc-editor.org/info/rfc3758>>.
- [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>>.

- [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>>.
- [RFC6525]    Stewart, R., Tuexen, M., and P. Lei, "Stream Control Transmission Protocol (SCTP) Stream Reconfiguration", RFC 6525, DOI 10.17487/RFC6525, February 2012, <<https://www.rfc-editor.org/info/rfc6525>>.
- [RFC7053]    Tuexen, M., Ruengeler, I., and R. Stewart, "SACK-IMMEDIATELY Extension for the Stream Control Transmission Protocol", RFC 7053, DOI 10.17487/RFC7053, November 2013, <<https://www.rfc-editor.org/info/rfc7053>>.

## 8.2. Informative References

- [I-D.ietf-rtcweb-data-channel]    Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channels", draft-ietf-rtcweb-data-channel-13 (work in progress), January 2015.
- [RFC3261]    Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<https://www.rfc-editor.org/info/rfc3261>>.
- [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>>.

## Authors' Addresses

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

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

Michael Tuexen  
Muenster University of Applied Sciences  
Stegerwaldstrasse 39  
48565 Steinfurt  
Germany

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

Salvatore Loreto  
Ericsson  
Torshamnsgatan 21  
164 80 Stockholm  
Sweden

Email: [Salvatore.Loreto@ericsson.com](mailto:Salvatore.Loreto@ericsson.com)

Robin Seggelmann  
Metafinanz Informationssysteme GmbH  
Leopoldstrasse 146  
80804 Muenchen  
Germany

Email: [rfc@robin-seggelmann.com](mailto:rfc@robin-seggelmann.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 11, 2015

M. Tuexen  
Muenster Univ. of Appl. Sciences  
R. Seggelmann  
T-Systems International GmbH  
R. Stewart  
Netflix, Inc.  
S. Loreto  
Ericsson  
February 7, 2015

Additional Policies for the Partial Reliability Extension of the Stream  
Control Transmission Protocol  
draft-ietf-tsvwg-sctp-prpolicies-07.txt

## Abstract

This document defines two additional policies for the Partial Reliability Extension of the Stream Control Transmission Protocol (PR-SCTP) allowing to limit the number of retransmissions or to prioritize user messages for more efficient send buffer usage.

## 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 11, 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.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions . . . . .	3
3. Additional PR-SCTP Policies . . . . .	3
3.1. Limited Retransmissions Policy . . . . .	3
3.2. Priority Policy . . . . .	3
4. Socket API Considerations . . . . .	4
4.1. Data Types . . . . .	4
4.2. Support for Added PR-SCTP Policies . . . . .	4
4.3. Socket Option for Getting the Stream Specific PR-SCTP Status (SCTP_PR_STREAM_STATUS) . . . . .	5
4.4. Socket Option for Getting the Association Specific PR- SCTP Status (SCTP_PR_ASSOC_STATUS) . . . . .	6
4.5. Socket Option for Getting and Setting the PR-SCTP Support (SCTP_PR_SUPPORTED) . . . . .	7
5. IANA Considerations . . . . .	8
6. Security Considerations . . . . .	8
7. Acknowledgments . . . . .	8
8. References . . . . .	8
8.1. Normative References . . . . .	8
8.2. Informative References . . . . .	9
Authors' Addresses . . . . .	9

## 1. Introduction

The SCTP Partial Reliability Extension (PR-SCTP) defined in [RFC3758] provides a generic method for senders to abandon user messages. The decision to abandon a user message is sender side only and the exact condition is called a PR-SCTP policy ([RFC3758] refers to them as 'PR-SCTP Services'). [RFC3758] also defines one particular PR-SCTP policy, called Timed Reliability. This allows the sender to specify a timeout for a user message after which the SCTP stack abandons the user message.

This document specifies the following two additional PR-SCTP policies:

Limited Retransmission Policy: Allows to limit the number of retransmissions.

Priority Policy: Allows to discard lower priority messages if space for higher priority messages is needed in the send buffer.

## 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. Additional PR-SCTP Policies

This section defines two new PR-SCTP policies, one in each subsection.

Please note that it is REQUIRED to implement [RFC3758], if you want to implement these additional policies. However, these additional policies are OPTIONAL when implementing [RFC3758].

### 3.1. Limited Retransmissions Policy

Using the Limited Retransmission Policy allows the sender of a user message to specify an upper limit for the number of retransmissions for each DATA chunk of the given user messages. The sender MUST abandon a user message if the number of retransmissions of any of the DATA chunks of the user message would exceed the provided limit. The sender MUST perform all other actions required for processing the retransmission event, such as adapting the congestion window and the retransmission timeout. Please note that the number of retransmissions includes both fast and timer-based retransmissions.

The sender MAY limit the number of retransmissions to 0. This will result in abandoning the message when it would get retransmitted for the first time. The use of this setting provides a service similar to UDP, which also does not perform any retransmissions.

Please note that using this policy does not affect the handling of the thresholds 'Association.Max.Retrans' and 'Path.Max.Retrans' as specified in Section 8 of [RFC4960].

The WebRTC protocol stack (see [I-D.ietf-rtcweb-data-channel]), is an example of where the Limited Retransmissions Policy is used.

### 3.2. Priority Policy

Using the Priority Policy allows the sender of a user message to specify a priority. When storing a user message in the send buffer while there is not enough available space, the SCTP stack at the sender side MAY abandon other user message(s) of the same SCTP



association (with the same or a different stream) with a priority lower than the provided one. User messages sent reliable are considered having a priority higher than all messages sent with the Priority Policy. The algorithm for selecting the message(s) being abandoned is implementation specific.

After lower priority messages have been abandoned high priority messages can be transferred without the send call blocking (if used in blocking mode) or the send call failing (if used in non-blocking mode).

The IPFIX protocol stack (see [RFC7011]) is an example of where the Priority Policy can be used. Template records would be sent with full reliability, while billing, security-related, and other monitoring flow records would be sent using the Priority Policy with varying priority. The priority of security related flow-records would be chosen higher than the the priority of monitoring flow records.

#### 4. Socket API Considerations

This section describes how the socket API defined in [RFC6458] is extended to support the newly defined PR-SCTP policies, to provide some statistical information and to control the negotiation of the PR-SCTP extension during the SCTP association setup.

Please note that this section is informational only.

##### 4.1. Data Types

This section uses data types from [IEEE.1003-1G.1997]: `uintN_t` means an unsigned integer of exactly N bits (e.g. `uint16_t`). This is the same as in [RFC6458].

##### 4.2. Support for Added PR-SCTP Policies

As defined in [RFC6458], the PR-SCTP policy is specified and configured by using the following `sctp_prinfo` structure:

```
struct sctp_prinfo {
    uint16_t pr_policy;
    uint32_t pr_value;
};
```

When the Limited Retransmission Policy described in Section 3.1 is used, `pr_policy` has the value `SCTP_PR_SCTP_RTX` and the number of retransmissions is given in `pr_value`.

When using the Priority Policy described in Section 3.2, `pr_policy` has the value `SCTP_PR_SCTP_PRIO`. The priority is given in `pr_value`. The value of zero is the highest priority and larger numbers in `pr_value` denote lower priorities.

The following table summarizes the possible parameter settings defined in [RFC6458] and this document:

<code>pr_policy</code>	<code>pr_value</code>	Specification
<code>SCTP_PR_SCTP_NONE</code>	Ignored	[RFC6458]
<code>SCTP_PR_SCTP_TTL</code>	Lifetime in ms	[RFC6458]
<code>SCTP_PR_SCTP_RTX</code>	Number of retransmissions	Section 3.1
<code>SCTP_PR_SCTP_PRIO</code>	Priority	Section 3.2

#### 4.3. Socket Option for Getting the Stream Specific PR-SCTP Status (`SCTP_PR_STREAM_STATUS`)

This socket option uses `IPPROTO_SCTP` as its level and `SCTP_PR_STREAM_STATUS` as its name. It can only be used with `getsockopt()`, but not with `setsockopt()`. The socket option value uses the following structure:

```
struct sctp_prstatus {
    sctp_assoc_t sprstat_assoc_id;
    uint16_t sprstat_sid;
    uint16_t sprstat_policy;
    uint64_t sprstat_abandoned_unsent;
    uint64_t sprstat_abandoned_sent;
};
```

`sprstat_assoc_id`: This parameter is ignored for one-to-one style sockets. For one-to-many style sockets this parameter indicates for which association the user wants the information. It is an error to use `SCTP_{CURRENT|ALL|FUTURE}_ASSOC` in `sprstat_assoc_id`.

`sprstat_sid`: This parameter indicates for which outgoing SCTP stream the user wants the information.

`sprstat_policy`: This parameter indicates for which PR-SCTP policy the user wants the information. It is an error to use `SCTP_PR_SCTP_NONE` in `sprstat_policy`. If `SCTP_PR_SCTP_ALL` is used, the counters provided are aggregated over all supported policies.

`sprstat_abandoned_unsent`: The number of user messages which have been abandoned using the policy specified in `sprstat_policy` on the

stream specified in `sprstat_sid` for the association specified by `sprstat_assoc_id`, before any part of the user message could be sent.

`sprstat_abandoned_sent`: The number of user messages which have been abandoned using the policy specified in `sprstat_policy` on the stream specified in `sprstat_sid` for the association specified by `sprstat_assoc_id`, after a part of the user message has been sent.

There are separate counters for unsent and sent user messages because the `SCTP_SEND_FAILED_EVENT` supports a similar differentiation. Please note that an abandoned large user message requiring an SCTP level fragmentation is reported in the `sprstat_abandoned_sent` counter as soon as at least one fragment of it has been sent. Therefore each abandoned user message is either counted in `sprstat_abandoned_unsent` or `sprstat_abandoned_sent`.

If more detailed information about abandoned user messages is required, the subscription to the `SCTP_SEND_FAILED_EVENT` is recommended. Please note that some implementations might choose not to support this option, since it increases the resources needed for an outgoing SCTP stream. For the same reasons, some implementations might only support using `SCTP_PR_SCTP_ALL` in `sprstat_policy`.

`sctp_opt_info()` needs to be extended to support `SCTP_PR_STREAM_STATUS`.

#### 4.4. Socket Option for Getting the Association Specific PR-SCTP Status (`SCTP_PR_ASSOC_STATUS`)

This socket option uses `IPPROTO_SCTP` as its level and `SCTP_PR_ASSOC_STATUS` as its name. It can only be used with `getsockopt()`, but not with `setsockopt()`. The socket option value uses the same structure as described in Section 4.3:

```
struct sctp_prstatus {
    sctp_assoc_t sprstat_assoc_id;
    uint16_t sprstat_sid;
    uint16_t sprstat_policy;
    uint64_t sprstat_abandoned_unsent;
    uint64_t sprstat_abandoned_sent;
};
```

`sprstat_assoc_id`: This parameter is ignored for one-to-one style sockets. For one-to-many style sockets this parameter indicates for which association the user wants the information. It is an error to use `SCTP_{CURRENT|ALL|FUTURE}_ASSOC` in `sprstat_assoc_id`.

`sprstat_sid`: This parameter is ignored.

`sprstat_policy`: This parameter indicates for which PR-SCTP policy the user wants the information. It is an error to use `SCTP_PR_SCTP_NONE` in `sprstat_policy`. If `SCTP_PR_SCTP_ALL` is used, the counters provided are aggregated over all supported policies.

`sprstat_abandoned_unsent`: The number of user messages which have been abandoned using the policy specified in `sprstat_policy` for the association specified by `sprstat_assoc_id`, before any part of the user message could be sent.

`sprstat_abandoned_sent`: The number of user messages which have been abandoned using the policy specified in `sprstat_policy` for the association specified by `sprstat_assoc_id`, after a part of the user message has been sent.

There are separate counters for unsent and sent user messages because the `SCTP_SEND_FAILED_EVENT` supports a similar differentiation. Please note that an abandoned large user message requiring an SCTP level fragmentation is reported in the `sprstat_abandoned_sent` counter as soon as at least one fragment of it has been sent. Therefore each abandoned user message is either counted in `sprstat_abandoned_unsent` or `sprstat_abandoned_sent`.

If more detailed information about abandoned user messages is required, the usage of the option described in Section 4.3 or the subscription to the `SCTP_SEND_FAILED_EVENT` is recommended.

`sctp_opt_info()` needs to be extended to support `SCTP_PR_ASSOC_STATUS`.

#### 4.5. Socket Option for Getting and Setting the PR-SCTP Support (`SCTP_PR_SUPPORTED`)

This socket option allows the enabling or disabling of the negotiation of PR-SCTP support for future associations. For existing associations it allows to query whether PR-SCTP support was negotiated or not on a particular association.

Whether PR-SCTP is enabled or not per default is implementation specific.

This socket option uses `IPPROTO_SCTP` as its level and `SCTP_PR_SUPPORTED` as its name. It can be used with `getsockopt()` and `setsockopt()`. The socket option value uses the following structure defined in [RFC6458]:

```
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, this parameter indicates upon which association the user is performing an action. The special `sctp_assoc_t Sctp_FUTURE_ASSOC` can also be used, it is an error to use `Sctp_{CURRENT|ALL}_ASSOC` in `assoc_id`.

assoc\_value: A non-zero value encodes the enabling of PR-SCTP whereas a value of 0 encodes the disabling of PR-SCTP.

`sctp_opt_info()` needs to be extended to support `Sctp_PR_SUPPORTED`.

## 5. IANA Considerations

This document requires no actions from IANA.

## 6. Security Considerations

This document does not add any additional security considerations in addition to the ones given in [RFC4960], [RFC3758], and [RFC6458]. As indicated in the Security Section of [RFC3758], transport layer security in the form of TLS over SCTP (see [RFC3436]) can't be used for PR-SCTP. However, DTLS over SCTP (see [RFC6083]) could be used instead. If DTLS over SCTP as specified in [RFC6083] is used, the security considerations of [RFC6083] do apply. It should also be noted that using PR-SCTP for an SCTP association doesn't allow that association to behave more aggressively than an SCTP association not using PR-SCTP.

## 7. Acknowledgments

The authors wish to thank Benoit Claise, Spencer Dawkins, Stephen Farrell, Gorrry Fairhurst, Barry Leiba, Karen Egede Nielsen, Ka-Cheong Poon, Dan Romascanu, Irene Ruengeler, Jamal Hadi Salim, Joseph Salowey, Brian Trammell, and Vlad Yasevich for their invaluable comments.

## 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.

- [RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, May 2004.
- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

## 8.2. Informative References

- [RFC3436] Jungmaier, A., Rescorla, E., and M. Tuexen, "Transport Layer Security over Stream Control Transmission Protocol", RFC 3436, December 2002.
- [RFC6083] Tuexen, M., Seggelmann, R., and E. Rescorla, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)", RFC 6083, January 2011.
- [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.
- [RFC7011] Claise, B., Trammell, B., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013.
- [I-D.ietf-rtcweb-data-channel]  
Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channels", draft-ietf-rtcweb-data-channel-13 (work in progress), January 2015.
- [IEEE.1003-1G.1997]  
Institute of Electrical and Electronics Engineers,  
"Protocol Independent Interfaces", IEEE Standard 1003.1G,  
March 1997.

## Authors' Addresses

Michael Tuexen  
Muenster University of Applied Sciences  
Stegerwaldstrasse 39  
48565 Steinfurt  
DE

Email: tuexen@fh-muenster.de

Robin Seggelmann  
T-Systems International GmbH  
Fasanenweg 5  
70771 Leinfelden-Echterdingen  
DE

Email: [rfc@robin-seggelmann.com](mailto:rfc@robin-seggelmann.com)

Randall R. Stewart  
Netflix, Inc.  
Chapin, SC 29036  
US

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

Salvatore Loreto  
Ericsson  
Hirsalantie 11  
Jorvas 02420  
FI

Email: [Salvatore.Loreto@ericsson.com](mailto:Salvatore.Loreto@ericsson.com)

IETF  
Internet-Draft  
Intended status: Best Current Practice  
Expires: December 14, 2013

G. Shepherd, Ed.  
Cisco Systems  
June 12, 2013

Multicast UDP Usage Guidelines for Application Designers  
draft-shepherd-multicast-udp-guidelines-01

Abstract

The multi-recipient nature of Multicast prevents the use of any point-to-point connection-oriented transport, therefore restricts all Multicast data to be sent over the User Datagram Protocol (UDP). UDP provides a minimal message-passing transport that has no inherent congestion control mechanisms. Because congestion control is critical to the stable operation of the Internet, applications and upper-layer protocols that choose to use Multicast UDP as an Internet service must employ mechanisms to prevent congestion collapse and to establish some degree of fairness with concurrent traffic. This document provides guidelines on the use of UDP for the designers of multicast applications and higher-level protocols.

Status of This Memo

This Internet-Draft is submitted to IETF 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 December 14, 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.

## Table of Contents

1. Introduction . . . . .	2
1.1. Requirements Language . . . . .	2
2. Multicast UDP Usage Guidelines . . . . .	3
2.1. Congestion Control Guidelines . . . . .	3
2.1.1. Bulk Transfer Applications . . . . .	3
2.1.2. Low Data-Volume Applications . . . . .	4
2.1.3. UDP Tunnels . . . . .	4
2.1.4. Message Size Guidelines . . . . .	4
3. Acknowledgements . . . . .	5
4. IANA Considerations . . . . .	5
5. Security Considerations . . . . .	6
6. References . . . . .	6
6.1. Normative References . . . . .	6
6.2. Informative References . . . . .	6
Appendix A. Additional Stuff . . . . .	8
Author's Address . . . . .	8

## 1. Introduction

The User Datagram Protocol (UDP) [RFC0768] provides a minimal, unreliable, best-effort, message-passing transport to applications and upper-layer protocols (both simply called "applications" in the remainder of this document). [RFC5405] is scoped to provide guidelines for unicast applications only, but all of the general requirements, references, and use cases apply to multicast [RFC1112][RFC4607] UDP application designers as well. This document chooses to only make recommendations in requirements, use cases, and references where they differ from [RFC5405] or are unique for applications sending multicast UDP data (simply called "multicast" in the remainder of this document).

### 1.1. 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 [RFC2119]

## 2. Multicast UDP Usage Guidelines

### 2.1. Congestion Control Guidelines

[RFC2309] discusses the dangers of congestion-unresponsive flows and states that "all UDP-based streaming applications should incorporate effective congestion avoidance mechanisms". Many large-scale multicast deployments are within a single administrative domain, and are provisioned over a bandwidth-reserved path or paths where congestion control is less relevant. But there are a growing number of deployment cases where multicast is transiting multiple domains, is tunneled across the unicast Internet, or transits the Internet through a unicast overlay network. This document is only concerned with the latter case of multicast data transiting the larger Internet, either as native IP multicast or encapsulated in a unicast tunnel and does not apply to administratively scoped deployments.

When the multicast traffic exits the administrative domain of a single network or the bi-laterally agreed path between networks, or is tunneled across the unicast Internet either to another multicast network or to an end device, the application SHOULD provide a TCP-compatible aggregate flow over the end-to-end path to each leaf.

There are currently two models of multicast delivery: the Any-Source Multicast (ASM) model as defined in [RFC1112] and the Source-Specific Multicast (SSM) model as defined in [RFC4607]. ASM group members will receive all data sent to the group by any source, while SSM constrains the distribution tree to only one single source. Many congestion-controlled transport protocols are often not applicable to multicast distribution services, or simply won't scale well to very large multicast trees since they require bi-directional communication and adapt the data-rate to accommodate the network conditions to a single receiver. Multicast distribution trees can often fan out to massive numbers of receivers limiting the scalability of an in-band return channel to control the data-rate, and the one-to-many nature of multicast distribution trees prevent adapting the data-rate to individual receiver requirements. For this reason, TCP-compatible aggregate flow for Internet multicast data, either native or tunneled, is the responsibility of the application.

#### 2.1.1. Bulk Transfer Applications

Applications that perform bulk transmission of data over a multicast distribution tree, i.e., applications that exchange more than a small number of UDP datagrams per maximum receiver RTT, SHOULD implement Asynchronous Layered Coding (ALC) [RFC5775], TCP-Friendly Multicast Congestion Control (TFMCC) [RFC4654], Wave and Equation Based Rate Control (WEBRC) [RFC3738], NACK-Oriented Reliable Multicast (NORM)

transport protocol [RFC5740], File Delivery over Unidirectional Transport (FLUTE) [RFC6726], Real Time Protocol/Control Protocol (RTP/RTCP), [RFC3550] or another congestion control scheme following the guidelines of [RFC2887] and utilizing the framework of [RFC3048].

Bulk transfer applications that choose not to implement [RFC4654], [RFC5775], [RFC3738], [RFC5740], [RFC6726], or [RFC3550] SHOULD implement a congestion control scheme that results in bandwidth use that competes fairly with TCP within an order of magnitude. Section 2 of [RFC3551] suggests that applications SHOULD monitor the packet loss rate to ensure that it is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path under the same network conditions would achieve an average throughput, measured on a reasonable timescale, that is not less than that of the UDP flow. The comparison to TCP cannot be specified exactly, but is intended as an "order-of-magnitude" comparison in timescale and throughput.

Finally, some bulk transfer applications may choose not to implement any congestion control mechanism and instead rely on transmitting across reserved path capacity. This might be an acceptable choice for a subset of restricted networking environments, but is by no means a safe practice for operation in the Internet. When the multicast traffic of such applications leaks out on unprovisioned Internet paths, it can significantly degrade the performance of other traffic sharing the path and even result in congestion collapse. Applications that support an uncontrolled or unadaptive transmission behavior SHOULD NOT do so by default and SHOULD instead require users to explicitly enable this mode of operation.

#### 2.1.2. Low Data-Volume Applications

All of the recommendations in section 3.1.2 of [RFC5405] are applicable to multicast as well.

#### 2.1.3. UDP Tunnels

All of the recommendations in section 3.1.3 of [RFC5405] are applicable to multicast carried inside of unicast UDP tunnels. There are, however deployment cases and solutions where the outer header of a UDP tunnel contains a multicast destination address, such as [RFC6513], but these are primarily deployed in bandwidth reserved environments within a single administrative domain, or between two domains where a bi-laterally agreed upon path and bandwidth is in place and so congestion control is not an issue.

#### 2.1.4. Message Size Guidelines

IP fragmentation lowers the efficiency and reliability of Internet communication. The loss of a single fragment results in the loss of an entire fragmented packet, because even if all other fragments are received correctly, the original packet cannot be reassembled and delivered. This fundamental issue with fragmentation exists for both IPv4 and IPv6, unicast and multicast packets. In addition, some network address translators (NATs) and firewalls drop IP fragments. The network address translation performed by a NAT only operates on complete IP packets, and some firewall policies also require inspection of complete IP packets. Even with these being the case, some NATs and firewalls simply do not implement the necessary reassembly functionality, and instead choose to drop all fragments. Finally, [RFC4963] documents other issues specific to IPv4 fragmentation.

Due to these issues, a multicast application SHOULD NOT send UDP datagrams that result in IP packets that exceed the effective MTU as described in section 3 of [RFC6807]. Consequently, an application SHOULD either use the effective MTU information provided by the Population Count Extensions to Protocol Independent Multicast [RFC6807] or implement path MTU discovery itself [RFC1191][RFC1981][RFC4821] to determine whether the path to each destination will support its desired message size without fragmentation.

If the multicast application is incapable of, or choose not to implement a worst-cast path MTU solution, the application SHOULD assume the maximum MTU of any link will be affected by multiple levels of encapsulation and SHOULD NOT send any packet larger than 1280 bytes.

### 3. Acknowledgements

This template was derived from an initial version written by Pekka Savola and contributed by him to the xml2rfc project.

This document is part of a plan to make xml2rfc indispensable [DOMINATION].

### 4. IANA Considerations

This memo includes no request to IANA.

All drafts are required to have an IANA considerations section (see the update of RFC 2434 [I-D.narten-iana-considerations-rfc2434bis] for a guide). If the draft does not require IANA to do anything, the section contains an explicit statement that this is the case (as above). If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.

## 5. Security Considerations

All drafts are required to have a security considerations section. See RFC 3552 [RFC3552] for a guide.

## 6. References

### 6.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [min\_ref] authSurName, authInitials., "Minimal Reference", 2006.

### 6.2. Informative References

- [DOMINATION] Mad Dominators, Inc., "Ultimate Plan for Taking Over the World", 1984, <<http://www.example.com/dominator.html>>.
- [I-D.narten-iana-considerations-rfc2434bis] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", draft-narten-iana-considerations-rfc2434bis-09 (work in progress), March 2008.
- [RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.
- [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.
- [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.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC2887] Handley, M., Floyd, S., Whetten, B., Kermode, R., Vicisano, L., and M. Luby, "The Reliable Multicast Design Space for Bulk Data Transfer", RFC 2887, August 2000.
- [RFC3048] Whetten, B., Vicisano, L., Kermode, R., Handley, M., Floyd, S., and M. Luby, "Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer", RFC 3048, January 2001.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.
- [RFC3738] Luby, M. and V. Goyal, "Wave and Equation Based Rate Control (WEBRC) Building Block", RFC 3738, April 2004.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.
- [RFC4654] Widmer, J. and M. Handley, "TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification", RFC 4654, August 2006.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.
- [RFC4963] Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly Errors at High Data Rates", RFC 4963, July 2007.
- [RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.

- [RFC5740] Adamson, B., Bormann, C., Handley, M., and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol", RFC 5740, November 2009.
- [RFC5775] Luby, M., Watson, M., and L. Vicisano, "Asynchronous Layered Coding (ALC) Protocol Instantiation", RFC 5775, April 2010.
- [RFC6513] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.
- [RFC6726] Paila, T., Walsh, R., Luby, M., Roca, V., and R. Lehtonen, "FLUTE - File Delivery over Unidirectional Transport", RFC 6726, November 2012.
- [RFC6807] Farinacci, D., Shepherd, G., Venaas, S., and Y. Cai, "Population Count Extensions to Protocol Independent Multicast (PIM)", RFC 6807, December 2012.

#### Appendix A. Additional Stuff

This becomes an Appendix.

#### Author's Address

Greg Shepherd (editor)  
Cisco Systems  
Tasman Drive  
San Jose  
USA

Email: gjshep@gmail.com

Internet Engineering Task Force  
INTERNET-DRAFT  
Intended Status: Standards Track  
Expires: January 2, 2016

X. Wei  
L.Zhu  
Huawei Technologies  
L.Deng  
China Mobile  
B.Briscoe  
July 1, 2015

Tunnel Congestion Feedback  
draft-wei-tsvwg-tunnel-congestion-feedback-04

## Abstract

This document describes a mechanism to calculate congestion of a tunnel segment based on RFC 6040 recommendations, and a feedback protocol by which to send the measured congestion of the tunnel from egress to ingress. A basic model for measuring tunnel congestion and feedback is described, and a protocol for carrying the feedback data is outlined.

## Status of this Memo

This Internet-Draft is submitted to IETF 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/lid-abstracts.html>

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

## Copyright and License 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.

## Table of Contents

1. Introduction . . . . .	3
2. Conventions . . . . .	3
3. Congestion Information Feedback Models . . . . .	4
3.1 Direct Model . . . . .	4
3.2 Centralized Model . . . . .	4
4. Congestion Level Measurement . . . . .	5
5. Congestion Information Delivery . . . . .	7
5.1 IPFIX Extentions . . . . .	7
5.1.1 ce-cePacketTotalCount . . . . .	7
5.1.2 ect-nectPacketTotalCount . . . . .	8
5.1.3 ce-nectPacketTotalCount . . . . .	8
5.1.4 ce-ectPacketTotalCount . . . . .	8
5.1.5 ect-ectPacketTotalCount . . . . .	9
6. Congestion Management . . . . .	9
7. Security . . . . .	9
8. IANA Considerations . . . . .	10
9. References . . . . .	10
9.1 Normative References . . . . .	10
9.2 Informative References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

In IP network, persistent congestion (or named congestion collapse) would cause transport throughput to drop down, lead to waste of network resource, so appropriate congestion control mechanisms are critical to make sure the network not fall into persistent congestion state. Currently, transport protocols such as TCP, SCTP, DCCP, has their built-in congestion control mechanism, and even for certain single transport protocol like TCP there could be a couple of different congestion control mechanism to choose. All these congestion control mechanisms are implemented on host side, and there are reasons that only host side congestion control is not sufficient for the whole network to keep away from persistent congestion, e.g., (1) some protocol's congestion control scheme might has internal design flaws; (2) improper software implementation of protocol; (3) some transport protocols even don't provide congestion control at all.

In order to have a better control on network congestion status, it's necessary for the network side to do certain kind of traffic control. For example, ConEx [ConEx] provides a method for network operator to learn about traffic's congestion contribution information, and then congestion management action could be taken based on this information.

Tunnels are widely deployed in various networks including public Internet, datacenter network, and enterprise network etc, a tunnel consists of an ingress, an egress and a set of interior routers. For the tunnel scenario, a tunnel-based mechanism which is different from ConEx is introduced for network traffic control to keep network away from persistent congestion; in this case, tunnel ingress will implement congestion management function to control the traffic entering the tunnel.

In order to do congestion management at ingress, the ingress must first get the inner tunnel congestion level information. But the ingress cannot use the locally visible traffic rates, because it would require additional knowledge of downstream capacity and topology, as well as cross traffic that does not pass through this ingress.

This document provide a mechanism of feeding back inner tunnel congestion level to ingress, using this mechanism the egress could feed the tunnel congestion level information it collects back to ingress, after receiving the information ingress could do congestion management according to network management policy.

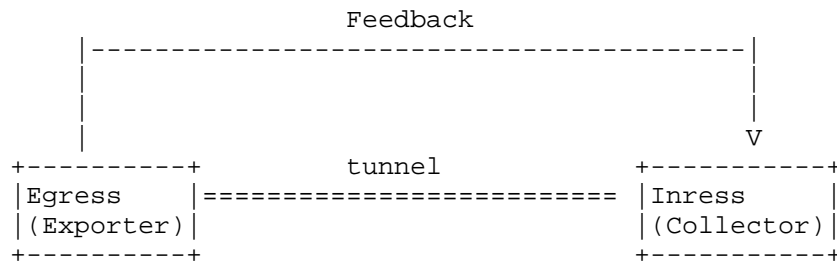
## 2. Conventions

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 RFC 2119 [RFC2119]

### 3. Congestion Information Feedback Models

According to specific network deployment, there are two kinds of feedback model: direct model and centralized model.

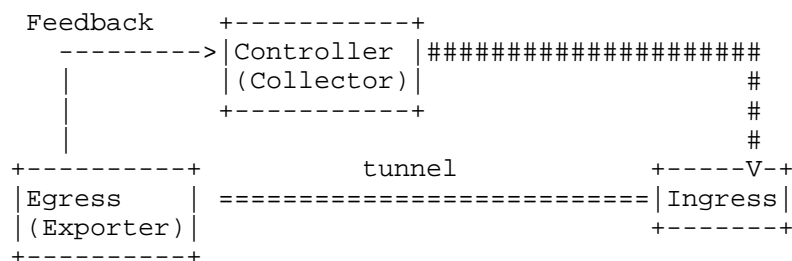
### 3.1 Direct Model



(a) Direct Feedback Model.

Direct model means egress feeds information directly to ingress. In this model, egress collects network congestion level information and feedback the information to ingress for congestion management. The ingress here will act as both decision point that decides how to do congestion management and action point that implements congestion management decision.

### 3.2 Centralized Model



(b) Centralized Feedback Model

There are scenarios that ingress only takes the role of action point, and it implements traffic control decision from another entity, named "controller" here.

In this model, after egress collects network congestion level information, it feeds back the information to controller instead of ingress, and then the controller makes congestion management decision and sends the decision to ingress.

#### 4. Congestion Level Measurement

This section describes how to measure congestion level in tunnel.

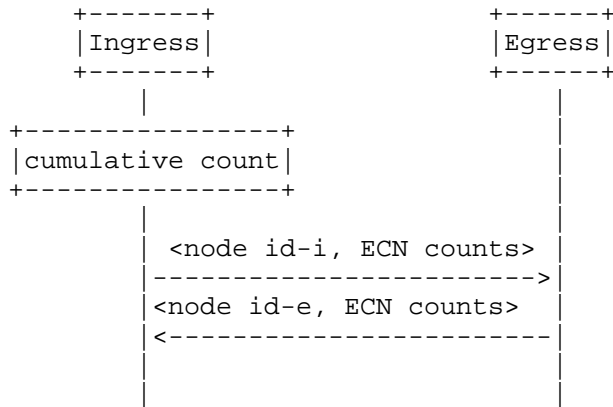
There may be different approaches of packet loss detection for different tunneling protocol scenarios, for instance, if there is a sequence field in tunneling protocol header, it will be easy for egress to detect packet loss through the gaps in sequence number space; another approach is to compare the number of packets entering ingress and the number of packets arriving at egress over the same span of packets. This document will focus on the latter one which is a more general approach.

If the routers support ECN, after router's queue length is over a predefined threshold, the routers will mark ECN packets as CE packets or drop not-ECN packets with the probability proportional to queue length, if the queue overflows all packets will be dropped; if the routers don't support ECN, after router's queue length is over a predefined threshold, the routers will drop both ECN packets and not-ECN packets with the probability proportional to queue length. It's assumed all routers in the tunnel support ECN.

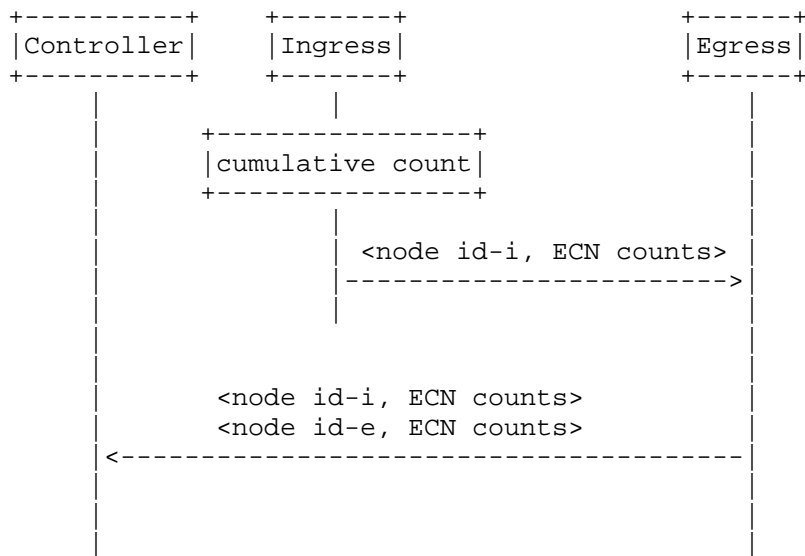
Faked ECT is used at ingress to defer packet loss to egress. The basic idea of faked ECT is that, when encapsulating packets, ingress first marks tunnel outer header according to RFC6040, and then remarks outer header of Not-ECT packet as ECT, there will be three kinds of combination of outer header ECN field and inner header ECN field: CE|CE, ECT|N-ECT, ECT|ECT (in the form of outer ECN| inner ECN).

In case all interior routers support ECN, the network congestion level could be indicated through the ratio of CE-marked packet and the ratio of packet drop, the relationship between these two kinds of indicator is complementary. If the congestion level in tunnel is not high enough, the packets would be marked as CE instead of being dropped, and then it is easy to calculate congestion level according to the ratio of CE-marked packets; if the congestion level is so high that ECT packet will be dropped, then the packet loss ratio could be calculated by comparing total packets entering ingress and total packets arriving at egress over the same span of packets, if packet loss is detected, it could be assumed that severe congestion has occurred in the tunnel, because loss is only ever a sign of serious congestion, so it doesn't need to measure loss ratio accurately.

The basic procedure of congestion level measurement is as follows:



(a) Direct model feedback procedure



(b) Centralized model feedback procedure

Ingress encapsulates packets and marks outer header according to faked ECT as described above. Ingress cumulatively counts packets for three types of ECN combination (CE|CE, ECT|N-ECT, ECT|ECT) and then the ingress regularly sends cumulative packet counts message of each type of ECN combination to the egress. When each message arrives, the

egress cumulatively counts packets coming from the ingress and adds its own packet counts of each type of ECN combination (CE|CE, ECT|N-ECT, CE|N-ECT, CE|ECT, ECT|ECT) to the message and either returns the whole message to the ingress, or to a central controller.

The counting of packets could be at the granularity of the all traffic from the ingress to the egress to learn about the overall congestion status of the path between the ingress and the egress; or at the granularity of individual customer's traffic or a specific set of flows to learn about their congestion contribution.

## 5. Congestion Information Delivery

As described above, the tunnel ingress needs to convey message of cumulative packet counts of each type of ECN combination to tunnel egress, and the tunnel egress also needs to feed the message of cumulative packet counts of each type of ECN combination to the ingress or central collector. This section describes how the messages could be conveyed.

The message could be along the same path with network data traffic, referred as in band signal; or go through a different path with network data traffic, referred as out of band signal. Because out of band scheme needs additional separate path which might limit its actual deployment, so the in band scheme will be discussed here.

Because the message is transmitted in band, so the message packet might get lost in case of network congestion. To cope with the situation that message packet gets lost, the packet counts values are sent as cumulative counters, so if a message is lost the next message will recover the missing information.

IPFIX [RFC7011] is selected as a choice of candidate protocol. IPFIX is preferred to use SCTP as transport, and because SCTP allows partially reliable delivery [RFC3758], which makes sure the feedback message will not be blocked to be sent in case of SCTP packets lost due to network congestion.

When sending message from ingress to egress, the ingress acts as IPFIX exporter and egress acts as IPFIX collector; when sending message from egress to ingress or controller, the egress acts as IPFIX exporter and ingress or controller acts as IPFIX collector.

### 5.1 IPFIX Extensions

#### 5.1.1 ce-cePacketTotalCount

Description: The total number of incoming packets with CE|CE ECN

marking combination for this Flow at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

ElementId: TBD1

Statuses: current

Units: packets

#### 5.1.2 ect-nectPacketTotalCount

Description: The total number of incoming packets with ECT|N-ECT ECN marking combination for this Flow at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

ElementId: TBD2

Statuses: current

Units: packets

#### 5.1.3 ce-nectPacketTotalCount

Description: The total number of incoming packets with CE|N-ECT ECN marking combination for this Flow at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

ElementId: TBD3

Statuses: current

Units: packets

#### 5.1.4 ce-ectPacketTotalCount

Description: The total number of incoming packets with CE|ECT ECN

marking combination for this Flow at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

ElementId: TBD4

Statuses: current

Units: packets

#### 5.1.5 ect-ectPacketTotalCount

Description: The total number of incoming packets with ECT|ECT ECN marking combination for this Flow at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

ElementId: TBD5

Statuses: current

Units: packets

### 6. Congestion Management

After tunnel ingress (or controller) receives congestion level information, then congestion management actions could be taken based on the information, e.g. if the congestion level is higher than a predefined threshold, then action could be taken to reduce the congestion level.

Congestion management action must be delayed by more than a worst-case global RTT, otherwise tunnel traffic management will not give normal e2e congestion control enough time to do its job, and the system could go unstable. The detailed description of congestion management is out of scope of this document, as examples, congestion management such as circuit breaker [CB] and congestion policing [CP] could be applied.

### 7. Security

This document describes the tunnel congestion calculation and



feedback. For feeding back congestion, security mechanisms of IPFIX are expected to be sufficient. No additional security concerns are expected.

## 8. IANA Considerations

This document defines a set of new IPFIX Information Elements (IE). New registry for these IE identifiers is needed.

TBD1~TBD5.

## 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001, <<http://www.rfc-editor.org/info/rfc3168>>.
- [RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, May 2004, <<http://www.rfc-editor.org/info/rfc3758>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013, <<http://www.rfc-editor.org/info/rfc7011>>.

### 9.2 Informative References

- [CONEX] Matt Mathis, Bob Briscoe. "Congestion Exposure (ConEx) Concepts, Abstract Mechanism and Requirements", draft-ietf-conex-abstract-mech-13, October 24, 2014
- [CB] G. Fairhurst. "Network Transport Circuit Breakers", draft-ietf-tsvwg-circuit-breaker-01, April 02, 2015
- [CP] Bob Briscoe, Murari Sridharan. "Network Performance Isolation in Data Centres using Congestion Policing", draft-briscoe-

conex-data-centre-02, February 14, 2014

Authors' Addresses

Xinpeng Wei  
Beiqing Rd. Z-park No.156, Haidian District,  
Beijing, 100095, P. R. China  
E-mail: weixinpeng@huawei.com

Zhu Lei  
Beiqing Rd. Z-park No.156, Haidian District,  
Beijing, 100095, P. R. China  
E-mail: lei.zhu@huawei.com

Lingli Deng  
Beijing, 100095, P. R. China  
E-mail: denglingli@gmail.com

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