Abstract

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

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

1.1 Background

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

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

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

1.2 Terminology and Abbreviations

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [RFC2119].
The initial discussions and presentation of [draft-rtp-ecn] produced a consensus that the specification of signaling was to be done within the AVTCore working group, and any subsequent discussion on end-to-end reactions to the signaling would be accomplished in the Transport Services (TSV) working group. This draft satisfies the latter effort.

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

3. Congestion Control Algorithms

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

- Smoother rate variation: (than for bulk data) to accommodate the underlying media flow’s characteristics.
- Low latency: Maintaining latencies sufficient to be usable, where 150ms is understood to be a good target [ITU.G114.2003].
- Burst handling: Ability to handle bursts due to the nature of the media and codec (e.g. I-frames etc)

3.1 TCP Friendly Rate Control (TFRC)

TFRC has a smoother response to congestion than TCP-like approaches, thus making it more suitable for real-time interactive multimedia applications. It has been cited in a number of other documents within the IETF for use with UDP and media flows [rfc3714, bcp145] and is seeing full and partial deployment in related solutions such as Empathy/Farsight, and GoogleTalk [goog1].
However it should be noted that TFRC is only recommended for real-time media use with ECN response. TFRC is not recommended for non-ECN paths due to its loss based operation which leads to full queues with maximised latencies. It is assumed that ECN markings will usually occur with lower queue occupancy and thus lower latency. However it is understood that ECN marks may not provide for sufficiently low latencies in some situations so other congestion control solutions would be preferable.

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

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

3.2 Related Work

3.2.1 3GPP

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

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

Though it is unclear exactly what the 3GPP community consider as too restrictive and whether some alteration of the response may be suitable. It should be noted that the 3GPP document only referred to an older version of TFRC defined in [RFC3448]. Given that the current version of TFRC [RFC5348] has made significant changes to the idle and data-limited responses it is unclear whether their assessment is relevant to current TFRC implementations.
Furthermore the specification [TR26.114] only outlines a rudimentary approach to congestion control, providing an example of a 60% back-off reaction to loss within an RTCP reporting period. The proposed signalling employs Temporary Maximum Media Stream Bit Rate Request (TMMBR) [RFC5104] and Codec Mode Request (CMR) [RFC4867] for video and audio respectively, which would only provide for very basic rate control if used as specified. We note that [TR26.114] specifies terminal behavior, while [TS36.300] specifies base station behaviour, though neither specify any standardised congestion control approach.

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

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

3.2.2 RTCweb

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

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

3.3 ECN response

As mentioned above and in accordance to [rfc3168], the actual response to the reception of an ECN-CE marked packet MUST normally be the same as that of a lost packet. However there are a number of contexts where one
may also be interested in more varied approaches. We expand on this in Section 5 below.

4. Application Layer Congestion Response

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

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

5. Other Reactions

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

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

5.1 Signaling

5.1.1 RSVP

The resource Reservation Protocol (RSVP) can be used to signal a desired set of path characteristics (e.g., bandwidth, delay) in response to CE feedback [rfc2205]. Its operation is based on the use of PATH messages sent downstream hop-by-hop from the source to a destination that specify requested forwarding characteristics. In return, the destination sends a hop-by-hop RESV message upstream towards the source confirming the resources that have been reserved for that flow.
[rfc3181] defines a priority policy element that specifies both an allocation and defending priority. This dual specification supports the use of preemption of existing reservations. [draft-priority-rsvp] is a work-in-progress that defines a new policy element that only conveys priority during reservation establishment. This latter effort also presents several reservation models, including one that describes engineered resources set aside for priority users.

5.1.1.1 Issues

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

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

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

5.1.2 Differentiated Services

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

As in the case of RSVP, applications could rely on the reception of CE feedback to initiate a subsequent setting of diff-serv code points to provide additional protection or explicit association of forwarding characteristics of a given flow of packets. In addition, the setting of
diff-serv code points would be done on an as-needed basis in reaction to CE feedback. Recommendations concerning specific diff-serv values are outside the scope of this document.

5.1.2.1 Issues

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

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

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

5.2 Fault Tolerance

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

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

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

Codec Swapping: This approach involves changing codecs to either reduce load or achieve an improvement in compensating for lost packets. Depending on the codec, the reduction of load may be a simple step
function, or it may involve a gradual and variable reduction in load based on the rate of congestion feedback received by the source.

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

5.2.1 Issues

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

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

5.3 Alternative Reaction for Emergency Communications

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

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

Simulations of the use of ECN exemption with TFRC and have found that it has limited effect on the normal flows with low numbers of exempt flows. A half-dumbbell network was used with a RED router queue configured using the
settings recommended by Sally Floyd. The candidate flows are 1Mbit/s each with a backhaul 100Mbit/s link. In the standard case where 1% of flows would be exempt the remaining flows achieve 99.99% of the bandwidth that they would achieve without the presence of the exempt flows. This is what would be expected from the simple calculation of the allocation, given that the exempt flows achieve their full rate (1Mbit/s); With 100 normal plus 1 exempt flow, assuming that the except flow uses 1Mbit/s, the remaining capacity is 99Mbit/s which is divided between the 100 normal flows. Whilst when 101 normal flows are run over the 100Mbit/s link they would have to share it evenly, so it works out thus: \((99/100)/(100/101)\)\*100=99.99%. In the case of 5% exempt flows then the proportion is very slightly lower at \((95/100)/(100/105)\)\*100=99.75%. Both these calculations are borne out in the simulation runs.

The level of exemption employed can be altered in a number of ways. Two simple approaches would be to either set a threshold number of ECN marked packets that could be considered as a loss, and another approach would be to set a percentage threshold of ECN marked packet that would be considered as a loss.

It should be noted that in the simulations the end-to-end delay of the packets within the flows was monitored and the relative delay of the exempt flows apparently rises somewhat when exemption is enacted. However what is actually occurring is that the ‘normal’ flows are reducing their throughput and are thus reducing their latency somewhat. There is normally some limited latency when using loss-based techniques such as TFRC because it fills the queues to ascertain the link capacity and maintains that level of delay throughout a session. However the level of latency is clearly limited by the queue sizes in the network and on media specific links these queue sizes are typically quite small, so the resulting latency is limited.

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

5.3.1 Issues

To Be Done

6. IANA Considerations

This document requires no actions from IANA.

7. Security Considerations

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

TBD

9. References

9.1 Normative


[rfc3583] Chaskar, H., "Requirements of a Quality of Service (QoS) Solution for Mobile IP", RFC 3583, September 2003


9.2 Informative


[tr26.114] "IMS; Multimedia telephony; Media Handling and Interaction", 3GPP, version 10, April 2011


Author's Addresses

Piers O’Hanlon
TCP and SCTP RTO Restart
draft-hurtig-tcpm-rtorestart-01

Abstract

This document describes a modified algorithm for managing the TCP and
SCTP retransmission timers, that provides faster loss recovery when a
connection's amount of outstanding data is small. The modification
allows the transport to restart its retransmission timer more
aggressively in situations where fast retransmit can not be used.
This enables faster loss detection, and recovery, for connections
that are short-lived or application-limited.

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 1, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
1. Introduction

TCP uses two mechanisms to detect segment loss. First, if a segment is not acknowledged within a certain amount of time, a retransmission timeout (RTO) occurs, and the segment is retransmitted [RFC6298]. While the RTO is based on measured round-trip times (RTTs) between the sender and receiver, it also has a conservative lower bound of 1 second to ensure that delayed segments are not mistaken as lost. Second, when a sender receives duplicate acknowledgments, the fast retransmit algorithm infers segment loss and triggers a retransmission. Duplicate acknowledgments are generated by a receiver when out-of-order segments arrive. As both segment loss and segment reordering cause out-of-order arrival, fast retransmit waits for three duplicate acknowledgments before considering the segment as lost. In some situations, however, the number of outstanding segments is not enough to trigger three duplicate acknowledgments, and the sender must rely on lengthy RTOs for loss recovery.

The amount of outstanding segments can be small for several reasons:

1. The connection is limited by the congestion control when the path has a low total capacity (bandwidth-delay product) or the connection’s share of the capacity is small. It is also limited by the congestion control in the first RTTs of a connection when the available capacity is probed using slow-start.

2. The connection is limited by the receiver’s available buffer space.

3. The connection is limited by the application when the total amount of data is small (e.g. web traffic) or if the available capacity of the path is not fully utilized (e.g. interactive applications).

The first two situations can occur for any flow, as external factors at the network and/or host level cause them. However, the third situation only affects flows that are short or have a low transmission rate. Typical examples of applications that produce short flows are web servers. [RJ10] shows that 70% of all web objects, found at the top 500 sites, are too small for fast retransmit to work. [BPS98] shows that about 56% of all objects...
retransmissions sent by a busy web server are sent after RTO expiry. While the experiments were not conducted using SACK [RFC2018], only 4% of the RTO-based retransmissions could have been avoided. Applications have a low transmission rate when data is sent in response to actions, or as a reaction to real life events. Typical examples of such applications are stock trading systems, remote computer operations and online games. What is special about this class of applications is that they are time-dependant, and extra latency can reduce the application service level [P09]. Although such applications may represent a small amount of data sent on the network, a considerable number of flows have such properties and the importance of low latency is high.

To enable timely loss recovery in the above situations a number of proposals have been made. The limited transmit mechanism [RFC3042] allows a TCP sender to transmit a previously unsent segment for each of the first two duplicate acknowledgments. By transmitting new segments, the sender attempts to generate additional duplicate acknowledgments to enable fast retransmit. However, the limited transmit algorithm does not help if no previously unset data is ready for transmission or if the receiver is out of buffer space. [RFC5827] specifies an early retransmit algorithm to enable fast loss recovery in such situations. By dynamically lowering the amount of duplicate acknowledgments needed for fast retransmit (dupthresh), based on the number of outstanding segments, a smaller number of duplicate acknowledgments are needed to trigger a retransmission. In some situations, however, the algorithm is of no use or might not work properly. First, if a single segment is outstanding, and lost, it is impossible to use early retransmit. Second, if the network path reorders segments, the algorithm might cause more unnecessary retransmissions than fast retransmit.

The RTO restart approach outlined in this document makes the RTO slightly more aggressive when the number of outstanding segments is small, in an attempt to enable faster loss recovery for all segments while being robust to reordering.

While this document focuses on TCP, the described changes are also valid for the Stream Control Transmission Protocol (SCTP) [RFC4960] which has similar loss recovery and congestion control algorithms.

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 RFC 2119 [RFC2119].
2. RTO Restart Overview

The RTO management algorithm described in [RFC6298] recommends that the retransmission timer is restarted when an acknowledgment (ACK) that acknowledges new data is received and there is still outstanding data. The restart is conducted to guarantee that unacknowledged segments will be retransmitted after approximately RTO seconds. However, by restarting the timer on each incoming acknowledgment, retransmissions are not typically triggered RTO seconds after their previous transmission but rather RTO seconds after the last ACK arrived. The duration of this extra delay depends on several factors but is in most cases approximately one RTT. Hence, in most situations the time needed to trigger an RTO is equal to "RTO + RTT". The restart approach is illustrated in Figure 1 where a TCP sender transmits three segments to a receiver. The arrival of the first and second segment triggers a delayed ACK [RFC1122], which restarts the RTO timer at the sender. The RTO restart is performed approximately one RTT after the transmission of the third segment. Thus, if the third segment is lost, as indicated in Figure 1, the effective loss detection time is "RTO + RTT" seconds. In some situations, the effective loss detection time becomes even longer. Consider a scenario where only two segments are outstanding. If the second segment is lost, the time to expire the delayed ACK timer will also be included in the effective loss detection time.

![Figure 1: RTO restart example](image)

During normal TCP bulk transfer the current RTO restart approach is not a problem. Actually, as long as enough segments arrive at a receiver to enable fast retransmit, RTO-based loss recovery should be avoided. RTOs should only be used as a last resort, as they drastically lower the congestion window compared to fast retransmit. There are only a few situations where timeouts are appropriate, or the only choice. For example, if the network is severely congested and no segments arrive, RTO-based recovery should be used. In this
situations, the time to recover from the loss(es) will not be the performance bottleneck. Furthermore, for connections that do not utilize enough capacity to enable fast retransmit, RTO is the only choice. The time needed for loss detection in such scenarios can become a serious performance bottleneck.

3. RTO Restart Algorithm

To enable faster loss recovery for connections that are unable to use fast retransmit, an alternative RTO restart can be used. By resetting the timer to "RTO - T_earliest", where T_earliest is the time elapsed since the earliest outstanding segment was transmitted, retransmissions will always occur after exactly RTO seconds. This approach makes the RTO more aggressive than the standardized approach in [RFC6298] but still conforms to the requirement in [RFC6298] that segments must not be retransmitted earlier than RTO seconds after their original transmission. Furthermore, the possible negative impacts of a more aggressive RTO are less severe when the amount of outstanding data is small. For example, if a spurious RTO is performed the resulting congestion window will not be smaller than if fast retransmit was used, as the window is already small. For larger congestion windows the RTT sometimes increases within a congestion window, due to the increased queueing demand of the flow. Making the timer more aggressive could trigger spurious retransmissions in this situation. However, as the restart is only conducted for small congestion windows, it is less likely that such RTT fluctuations would occur.

This document specifies the following update of step 5.3 in Section 5 of [RFC6298] (and a similar update in Section 6.3.2 of [RFC4960] for SCTP):

When an ACK is received that acknowledges new data:

1. Set T_earliest = 0.

2. If the following two conditions hold:
   
   (a) The number of outstanding segments is less than four.

   (b) There is no unsent data ready for transmission or the receiver’s advertised window does not permit transmission.

   set T_earliest to the time elapsed since the earliest outstanding segment was sent.
(3) Restart the retransmission timer so that it will expire after "RTO - T_earliest" seconds (for the current value of RTO).

The update requires TCP implementations to track the time elapsed since the transmission of the earliest outstanding segment (T_earliest). In practice, this requires the implementation to store the transmission time of each unacknowledged segment. In modern implementations like Linux, this is already done to easily conduct RTT measurements.

There are several proposals that make use of a different dupthresh than three. TCP-NCR [RFC4653] sets the dupthresh to three or more, to better disambiguate reordered and lost segments. In addition, the earlier mentioned early retransmit mechanism dynamically lowers the dupthresh when the amount of outstanding data is small, to enable faster loss recovery. The reasons why the RTO restart procedure described in this document does not take dynamic dupthresh considerations into account are twofold. First, if a larger dupthresh is used, the RTO restart approach could be used when the congestion window, and the amount of outstanding data, is larger. However, in such situations the actual amount of outstanding data can significantly impact the RTT of the connection, making it potentially dangerous to be more aggressive. Second, if a smaller dupthresh is used, the amount of outstanding data needed for a restart is smaller. However, as the congestion window is already small, it does not matter if a retransmission is due to a fast retransmit or an RTO. The resulting congestion window will still be very small, and the only difference is how quickly TCP infers segment loss.

4. Discussion

5. Related Work

The currently standardized algorithm has been shown to add at least one RTT to the loss recovery process in TCP [LS00] and SCTP [HB08][PBP09]. Applications that have strict timing requirements (e.g. telephony signaling and gaming) rather than throughput requirements may want to use a lower RTOmin than the standard of 1 second [RFC4166]. For such applications the modified restart approach could be important as the RTT and also the delayed ACK timer of receivers will be large components of the effective loss recovery time. In [HB08] where an RTOmin of 100 ms was used, it was shown that the total transfer time of a lost segment (including the original transmission time and the loss recovery time) could be reduced with up to 35% using the suggested approach.
6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This document discusses a change in how to set the retransmission timer’s value when restarted. This change does not raise any new security issues with TCP or SCTP.

8. References

8.1. Normative References


8.2. Informative References


Authors’ Addresses

Per Hurtig
Karlstad University
Universitetsgatan 2
Karlstad, 651 88
Sweden

Phone: +46 54 700 23 35
Email: per.hurtig@kau.se
Datagram Congestion Control Protocol (DCCP) Encapsulation for NAT Traversal (DCCP-UDP)
draft-ietf-dccp-udpencap-11

Abstract

This document specifies an alternative encapsulation of the Datagram Congestion Control Protocol (DCCP), referred to as DCCP-UDP. This encapsulation allows DCCP to be carried through the current generation of Network Address Translation (NAT) middleboxes without modification of those middleboxes. This document also updates the SDP information for DCCP defined in RFC 5762.

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 27, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction ........................................................................ 3
2. Terminology ........................................................................ 4
3. DCCP-UDP ....................................................................... 4
   3.1. The UDP Header .......................................................... 5
   3.2. The DCCP Generic Header .............................................. 5
   3.3. DCCP-UDP Checksum Procedures .................................. 6
       3.3.1. Partial Checksums and the Minimum Checksum Coverage Feature ........................................ 7
   3.4. Network Layer Options ................................................ 8
   3.5. Explicit Congestion Notification .................................... 8
   3.6. ICMP handling for messages relating to DCCP-UDP ........ 8
   3.7. Path Maximum Transmission Unit Discovery .................. 9
   3.8. Usage of the UDP port by DCCP-UDP ............................ 9
   3.9. Service Codes and the DCCP Port Registry ..................... 11
4. DCCP-UDP and Higher-Layer Protocols ................................. 11
   5.1. Protocol Identification ................................................ 12
   5.2. Signalling Encapsulated DCCP Ports .............................. 13
   5.3. Connection Management .............................................. 14
   5.4. Negotiating the DCCP-UDP encapsulation versus native DCCP ...................................................... 14
   5.5. Example of SDP use .................................................... 15
6. Security Considerations ..................................................... 16
7. IANA Considerations ........................................................ 16
   7.1. UDP Port Allocation .................................................... 17
   7.2. DCCP Reset ............................................................... 17
   7.3. SDP Attribute Allocation .............................................. 17
8. Acknowledgments ............................................................. 18
9. References ......................................................................... 18
   9.1. Normative References ................................................ 18
   9.2. Informative References ............................................... 18
Authors’ Addresses .......................................................... 20
1. Introduction

The Datagram Congestion Control Protocol (DCCP) [RFC4340] is a transport-layer protocol that provides upper layers with the ability to use non-reliable congestion-controlled flows. The current specification for DCCP specifies a direct native encapsulation in IPv4 or IPv6 packets.

DCCP support has been specified for devices that use Network Address Translation (NAT) or Network Address and Port Translation (NAPT) [RFC5597]. However, there is a significant installed base of NAT/NAPT devices that do not support RFC 5597. It is therefore useful to have an encapsulation for DCCP that is compatible with this installed base of NAT/NAPT devices that support [RFC4787], but do not support RFC 5597. This document specifies that encapsulation, which is referred to as DCCP-UDP. For convenience, the standard encapsulation for DCCP [RFC4340] (including [RFC5596] as required) is referred to as DCCP-STD.

The encapsulation described in this document may also be used as a transition mechanism to enable support for DCCP in devices that support UDP, but do not yet natively support DCCP. This also allows the DCCP transport to be implemented within an application using DCCP-UDP.

The document also updates the SDP specification for DCCP to convey the encapsulation type. In this respect only, it updates the method in [RFC5762].

The DCCP-UDP encapsulation specified in this document supports all of the features contained in DCCP-STD, but with limited functionality for partial checksums.

Network optimisations for DCCP-STP and UDP may need to be updated to allow these optimisations to take advantage of DCCP-UDP. Encapsulation with an additional UDP protocol header can complicate or prevent inspection of DCCP header fields by equipment along the network path in the case where multiple DCCP connections share the same UDP 4-tuple. For example, routers that wish to identify DCCP ports to perform Equal-Cost Multi-Path routing, ECMP, network devices that wish to inspect DCCP ports to inform algorithms for sharing the network load across multiple links; firewalls that wish to inspect DCCP ports and service codes to inform algorithms that implement access rules; media gateways that inspect SDP information to derive characteristics of the transport and session, etc.
2. Terminology

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

3. DCCP-UDP

The basic approach is to insert a UDP [RFC0768] header between the IP header and the DCCP packet. Note that this is not a tunneling approach. The IP addresses of the communicating end systems are carried in the IP header. The method does not embed additional IP addresses.

The method is designed to support use when these addresses are modified by a device that implements NAT/NAPT. A NAT translates the IP addresses, which impacts the transport-layer checksum. A NAPT device may also translate the port values (usually the source port). In both cases, the outer transport header that includes these values would need to be updated by the NAT/NAPT.

A device offering or using DCCP services via DCCP-UDP encapsulation listens on a UDP port (default port, XXX IANA PORT XXX), or may bind to a specified port utilising out-of-band signalling, such as the Session Description Protocol (SDP). The DCCP-UDP server accepts incoming packets over the UDP transport and passes the received packets to the DCCP protocol module, after removing the UDP encapsulation.

A DCCP implementation endpoint may simultaneously provide services over any or all combinations of DCCP-STD and/or DCCP-UDP encapsulations with IPv4 and/or IPv6.

The basic format of a DCCP-UDP packet is:

```
+-----------------------------------+  Variable length
|     IP Header (IPv4 or IPv6)     |
|---------------------------------+  8 bytes
|       UDP Header                |
|---------------------------------+  12 or 16 bytes
|       DCCP Generic Header       |
|---------------------------------+  Variable length (could be 0)
| Additional (type-specific) Fields|
|---------------------------------+  Variable length (could be 0)
|       DCCP Options              |
|---------------------------------+  Variable length (could be 0)
|     Application Data Area       |
```
Section 3.8 describes usage of UDP ports. This includes implementation of a DCCP-UDP encapsulation service as a daemon that listens on a well-known port, allowing multiplexing of different DCCP applications over the same port.

3.1. The UDP Header

The format of the UDP header is specified in [RFC0768]:

```
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |          Source Port          |           Dest Port           |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |             Length            |           Checksum            |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

For DCCP-UDP, the fields are interpreted as follows:

Source and Dest(ination) Ports: 16 bits each

These fields identify the UDP ports on which the source and destination (respectively) of the packet are listening for incoming DCCP-UDP packets. The UDP port values do not identify the DCCP source and destination ports.

Length: 16 bits

This field is the length of the UDP datagram, including the UDP header and the payload (for DCCP-UDP, the payload is a DCCP-UDP datagram).

Checksum: 16 bits

This field is the Internet checksum of a network-layer pseudoheader and Length bytes of the UDP packet [RFC0768]. The UDP checksum MUST NOT be zero for a UDP packet that carries DCCP-UDP.

3.2. The DCCP Generic Header

The DCCP Generic Header [RFC4340] takes two forms, one with long sequence numbers (48 bits) and the other with short sequence numbers (24 bits).
3.3. DCCP-UDP Checksum Procedures

DCCP-UDP employs a checksum at the UDP level and eliminates the use of the DCCP checksum. This approach was chosen to enable use of current NAT/NATP traversal methods developed for UDP. Such methods will generally be unaware whether DCCP is being encapsulated and hence do not update the inner checksum in the DCCP header. Standard DCCP requires protection of the DCCP header fields, this justifies any processing overhead incurred from calculating the UDP checksum.

In addition, UDP NAT traversal does not support partial checksums. Although this is still permitted end-to-end in the encapsulated DCCP
datagram, links along the path will treat these as UDP packets and can not enable special partial checksum processing.

DCCP-UDP does not update or modify the operation of UDP. The UDP transport protocol is used in the following way:

For DCCP-UDP, the function of the DCCP Checksum field is performed by the UDP checksum field. On transmit, the DCCP Checksum field SHOULD be set to zero. On receive, the DCCP Checksum field MUST be ignored.

The UDP checksum MUST NOT be zero for a UDP packet that is sent using DCCP-UDP. If the received UDP Checksum field is zero, the packet MUST be dropped [RFC5405].

If the UDP Length field is less than 20 (the UDP Header length and minimum DCCP-UDP header length), the packet MUST be dropped [RFC5405].

If the UDP Checksum field, computed using standard UDP methods, is invalid, the packet MUST be dropped [RFC5405].

If the UDP Length field in a received packet is less than the length of the UDP header plus the entire DCCP-UDP header (including the generic header and type-specific fields and options, if present), or the UDP Length field is greater than the length of the packet from the beginning of the UDP header to the end of the packet, the packet MUST be dropped.

3.3.1. Partial Checksums and the Minimum Checksum Coverage Feature

This document describes an encapsulation for DCCP that uses the UDP transport. It requires the UDP checksum to be enabled. This checksum provides coverage of the entire encapsulated DCCP datagram.

DCCP-UDP supports the syntax of partial checksums. It also supports negotiation of the Minimum Checksum Coverage feature and settings of the CsCov field. However, the UDP checksum field in DCCP-UDP always covers the entire DCCP datagram and the DCCP checksum is ignored on receipt. An application that enables the partial checksums feature in the DCCP Module will therefore experience a service that is functionally identical to using full DCCP checksum coverage. This is also the service that the application would have received if it had used a network path that did not provide optimised processing for DCCP partial checksums.
3.4. Network Layer Options

A DCCP-UDP implementation MAY transfer network-layer options intended for DCCP to the network-layer header of the encapsulating UDP packet.

A DCCP-UDP endpoint that receives IP-options for the encapsulating UDP packet MAY forward these to the DCCP protocol module. If the endpoint forwards a specific network layer option to the DCCP module, it MUST also forward all subsequent packets with this option. Consistent forwarding is essential for correct operation of many end-to-end options.

3.5. Explicit Congestion Notification

A DCCP-UDP endpoint SHOULD follow the procedures of DCCP-STD section 12 by setting the ECN fields in the IP Headers of outgoing packets and examining the values received in the ECN fields of incoming IP packets, relaying any packet markings to the DCCP module.

Implementations that do not support ECN MUST follow the procedures in DCCP-STD section 12.1 with regard to implementations that are not ECN capable.

3.6. ICMP handling for messages relating to DCCP-UDP

To allow ICMP messages to be demultiplexed by the receiving endpoint, part of the original packet that resulted in the message is included in the payload of the ICMP error message. The receiving endpoint can therefore use this information to associate the ICMP error with the transport protocol instance that resulted in the ICMP message. When DCCP-UDP is used, the error message and the payload of the ICMP error message relate to the UDP transport.

DCCP-UDP endpoints SHOULD forward ICMP messages relating to a UDP packet that carries a DCCP-UDP to the DCCP module. This may imply translation of the payload of the ICMP message into a form that is recognised by the DCCP stack. [RFC5927] describes precautions that are desirable before TCP acts on the receipt of an ICMP message. Similar precautions are desirable prior to forwarding by DCCP-UDP to the DCCP module.

The minimal length ICMP error message generated in response to processing a UDP Datagram only identifies the Source UDP Port and Destination UDP Port. This ICMP message does not carry sufficient information to discover the encapsulated DCCP Port values. A DCCP-UDP endpoint that supports multiple DCCP connections over the same pair of UDP ports (see section Section 3.8) may not therefore be able to associate an ICMP message with a unique DCCP-UDP connection.
3.7. Path Maximum Transmission Unit Discovery

DCCP-UDP implementations MUST follow DCCP-STD [RFC4340], section 14 with regard to determining the maximum packet size and the use of Path Maximum Transmission Unit Discovery (PMTUD). This requires the processing of ICMP Destination Unreachable messages with a Code that indicates that an unfragmentable packet was too large to be forwarded (a "Datagram Too Big" message), as defined in RFC 4340.

An effect of encapsulation is to incur additional datagram overhead. This will reduce the Maximum Packet Size (MPS) at the DCCP level.

3.8. Usage of the UDP port by DCCP-UDP

A DCCP-UDP server (that is, an initially passive endpoint that wishes to receive DCCP-Request packets [RFC4340] over DCCP-UDP) listens for connections on one or more UDP ports. UDP port number XXX IANA PORT XXX has been reserved as the default listening UDP port for a DCCP-UDP server. Some NAT/NAPT topologies may require using a non-default listening port.

The purpose of this IANA-assigned port is for the operating system or a framework to receive and process DCCP-UDP datagrams for delivery to the DCCP module (e.g. to support a system-wide DCCP-UDP daemon serving multiple DCCP applications or a DCCP-UDP server placed behind a firewall).

An application-specific implementation SHOULD use an ephemeral port and advertise this port using outside means, e.g. SDP. This method of implementation SHOULD NOT use the IANA-assigned port to listen for incoming DCCP-UDP packets.

A DCCP-UDP client provides UDP source and destination ports as well as DCCP source and destination ports at connection initiation time. A client SHOULD ensure that each DCCP connection maps to a single DCCP-UDP connection by setting the UDP source port. Choosing a distinct source UDP port for each distinct DCCP connection ensures that UDP-based flow identifiers differ whenever DCCP-based flow identifiers differ. Specifically, two connections with different <source IP address, source DCCP port, destination IP address, destination DCCP port> DCCP 4-tuples will have different <source IP address, source UDP port, destination IP address, destination UDP port> UDP 4-tuples.

A DCCP-UDP server SHOULD accept datagrams from any UDP source port. There is a risk that the same DCCP source port number could be used by two endpoints each behind a NAPT. A DCCP-UDP server MUST therefore demultiplex a DCCP-UDP flow using both the UDP source and
destination port numbers and the encapsulated DCCP ports. This ensures that an active DCCP connection is uniquely identified by the 6-tuple <source IP address, source UDP port, source DCCP port, destination IP address, destination UDP port, destination DCCP port>. (The active state of a DCCP connection is defined in Section 3.8: A DCCP connection becomes active following transmission of a DCCP-Request, and become inactive after sending a DCCP-Close.)

This demultiplexing at a DCCP-UDP endpoint occurs in two stages:

1) In the first stage, DCCP-UDP packets are demultiplexed using the UDP 4-tuple: <source IP address, source UDP port, destination IP address, destination UDP port>.

2) In the second stage, a receiving endpoint MUST ensure that two independent DCCP connections that were multiplexed to the same UDP 4-tuple are not associated with the same connection in the DCCP module. The endpoint therefore needs to keep state for the set of active DCCP-UDP endpoints using each combination of a UDP 4-tuple: <source IP address, source UDP port, destination IP address, destination UDP port>. Two DCCP endpoint methods are specified. A DCCP-UDP implementation MUST implement exactly one of these:

- The DCCP server may accept only one active 6-tuple at any one time for a given UDP 4-tuple. In this method, DCCP-UDP packets that do not match an active 6-tuple MUST NOT be passed to the DCCP module and the DCCP Server SHOULD send a DCCP-Reset with with Reset Code XXX IANA Port Reuse XXX, "Encapsulated Port Reuse". An endpoint that receives a DCCP-Reset with this reset code will clear its connection state, but MAY immediately try again using a different 4-tuple. This provides protection should the same UDP 4-tuple be re-used by multiple DCCP connections, ensuring that only one DCCP connection is established at one time.

- The DCCP server may support multiple DCCP connections over the same UDP 4-tuple. In this method, the endpoint MUST then associate each 6-tuple with a single DCCP connection. If an endpoint is unable to demultiplex the 6-tuple (e.g. due to internal resource limits), it MUST discard DCCP-UDP packets that do not match an active 6-tuple instead of forwarding them to the DCCP module. The DCCP endpoint MAY send a DCCP-Reset with Reset Code XXX IANA Port Reuse XXX, "Encapsulated Port Reuse", indicating the connection has been closed, but may be retried using a different UDP 4-tuple.
3.9. Service Codes and the DCCP Port Registry

This section clarifies the usage of DCCP Service Codes and the registration of server ports by DCCP-UDP. The section is not intended to update the procedures for allocating Service Codes or server ports.

There is one Service Code registry and one DCCP port registration that apply to all combinations of encapsulation and IP version. A DCCP Service Code specifies an application using DCCP regardless of the combination of DCCP encapsulation and IP version. An application may choose not to support some combinations of encapsulation and IP version, but its Service Code will remain registered for those combinations and the Service Code must not be used by other applications. An application should not register different Service Codes for different combinations of encapsulation and IP version. [RFC5595] provides additional information about DCCP Service Codes.

Similarly, a DCCP port registration is applicable to all combinations of encapsulation and IP version. Again, an application may choose not to support some combinations of encapsulation and IP version on its registered DCCP port, although the port will remain registered for those combinations. Applications should not register different DCCP ports just for the purpose of using different combinations of encapsulation.

4. DCCP-UDP and Higher-Layer Protocols

The encapsulation of a higher-layer protocol within DCCP MUST be the same for both DCCP-STD and DCCP-UDP. Encapsulation of Datagram Transport Layer Security (DTLS) over DCCP is defined in [RFC5238] and RTP over DCCP is defined in [RFC5762]. This document therefore does not update these encapsulations when using DCCP-UDP.

5. Signaling the Use of DCCP-UDP

Applications often signal transport connection parameters through outside means, such as SDP. Applications that define such methods for DCCP MUST define how the DCCP encapsulation is chosen, and MUST allow either encapsulation to be signaled. Where DCCP-STD and DCCP-UDP are both supported, DCCP-STD SHOULD be preferred.

The Session Description Protocol (SDP) [RFC4566] and the offer/answer model [RFC3264] can be used to negotiate DCCP sessions, and [RFC5762] defines SDP extensions for signalling the use of an RTP session running over DCCP connections. However, since [RFC5762] predates
this document, it does not define a mechanism for signalling that the
DCCP-UDP encapsulation is to be used. This section updates [RFC5762]
to describe how SDP can be used to signal RTP sessions running over
the DCCP-UDP encapsulation.

The new SDP support specified in this section is expected to be
useful when the offering party is on the public Internet, or in the
same private addressing realm as the answering party. In this case,
the DCCP-UDP server has a public address. The client may either have
a public address or be behind a NAT/NAPT. This scenario has the
potential to be an important use-case. Some other NAT/NAPT
topologies may result in the advertised port being unreachable via
the NAT/NAPT.

5.1. Protocol Identification

SDP uses a media ("m=") line to convey details of the media format
and transport protocol used. The ABNF syntax [RFC5124] of a media
line for DCCP is as follows (from [RFC4566]):

media-field = %x6d "=" media SP port ["/"] integer SP proto
1*(SP fmt) CRLF

The proto field denotes the transport protocol used for the media,
while the port indicates the transport port to which the media is
sent, following [RFC5762]. This document defines the following five
values of the proto field to indicate media transported using DCCP-UDP encapsulation:

UDP/DCCP
UDP/DCCP/RTP/AVP
UDP/DCCP/RTP/SAVP
UDP/DCCP/RTP/AVPF
UDP/DCCP/RTP/SAVPF

The "UDP/DCCP" protocol identifier is similar to the "DCCP" protocol
identifier defined in [RFC5762] and denotes the DCCP transport
protocol encapsulated in UDP, but not its upper-layer protocol.

The "UDP/DCCP/RTP/AVP" protocol identifier refers to RTP using the
RTP Profile for Audio and Video Conferences with Minimal Control
[RFC3551] running over the DCCP-UDP encapsulation.

The "UDP/DCCP/RTP/AVPF" protocol identifier refers to RTP using the Extended RTP Profile for RTCP-based Feedback [RFC4585] running over the DCCP-UDP encapsulation.

The "UDP/DCCP/RTP/SAVPF" protocol identifier refers to RTP using the Extended Secure RTP Profile for RTCP-based Feedback [RFC5124] running over the DCCP-UDP encapsulation.

The fmt value in the "m=" line is used as described in [RFC5762].

The port number specified in the "m=" line indicates the UDP port that is used for the DCCP-UDP encapsulation service. The DCCP port number MUST be sent using an associated "a=dccp-port:" attribute, as described in Section 5.2.

The use of ports with DCCP-UDP encapsulation is described further in Section 3.8.

5.2. Signalling Encapsulated DCCP Ports

When using DCCP-UDP, the UDP port used for the encapsulation is signalled using the SDP "m=" line. The DCCP ports MUST NOT be included in the "m=" line, but are instead signalled using a new SDP attribute ("dccp-port") defined according to the following ABNF:

```
dccp-port-attr = %x61 "=dccp-port:" dccp-port

dccp-port = 1*DIGIT
```

where DIGIT is as defined in [RFC5234]. This is a media level attribute, that is not subject to the charset attribute. The "a=dccp-port:" attribute MUST be included when the protocol identifiers described in Section 5.1 are used.

The use of ports with DCCP-UDP encapsulation is described further in Section 3.8.

- If the "a=rtcp:" attribute [RFC3605] is used, then the signalled port is the DCCP port used for RTCP.
- If the "a=rtcp-mux" attribute [RFC5761] is negotiated, then RTP and RTCP are multiplexed onto a single DCCP port, otherwise separate DCCP ports are used for RTP and RTCP [RFC5762].
In each case, only a single UDP port is used for the DCCP-UDP encapsulation.

- If the "a=rtcp-mux" attribute is not present, then the second of the two demultiplexing methods described in Section 3.8 MUST be implemented, otherwise the second DCCP connection for the RTCP flow will be rejected. For this reason, using "a=rtcp-mux" is RECOMMENDED when using RTP over DCCP-UDP.

5.3. Connection Management

The "a=setup:" attribute is used in a manner compatible with [RFC5762] Section 5.3 to indicate which of the DCCP-UDP endpoints should initiate the DCCP-UDP connection establishment.

5.4. Negotiating the DCCP-UDP encapsulation versus native DCCP

An endpoint that supports both native DCCP and the DCCP-UDP encapsulation may wish to signal support for both options in an SDP offer, allowing the answering party the option of using native DCCP where possible, while falling back to the DCCP-UDP encapsulation otherwise.

An approach to doing this might be to include candidates for the DCCP-UDP encapsulation and native DCCP into an Interactive Connectivity Establishment (ICE) [RFC5245] exchange. Since DCCP is connection-oriented, these candidates would need to be encoded into ICE in a manner analogous to TCP candidates defined in [RFC6544]. Both active and passive candidates could be supported for native DCCP and DCCP-UDP encapsulation, as may DCCP simultaneous open [RFC5596]. In choosing local preference values, it may make sense to to prefer DCCP-UDP over native DCCP in cases where low connection setup time is important, and to prioritise native DCCP in cases where low overhead is preferred (on the assumption that DCCP-UDP is more likely to work through legacy NAT, but has higher overhead). The details of this encoding into ICE are left for future study.

While ICE is appropriate for selecting basic use of DCCP-UDP versus DCCP-STD, it may not be appropriate for negotiating different RTP profiles with each transport encapsulation. The SDP Capability Negotiation framework [RFC5939] may be more suitable. Section 3.7 of RFC 5939 specifies how to provide attributes and transport protocols as capabilities and negotiate them using the framework. The details of the use of SDP Capability Negotiation with DCCP are left for future study.
5.5. Example of SDP use

The example below shows an SDP offer, where an application signals support for DCCP-UDP:

```
v=0
o=alice 1129377363 1 IN IP4 192.0.2.47
s=-
c=IN IP4 192.0.2.47
t=0 0
m=video 50234 UDP/DCCP/RTP/AVP 99
a=rtpmap:99 h261/90000
a=dccp-service-code:SC=x52545056
a=dccp-port:5004
a=rtcp:5005
a=setup:passive
a=connection:new
```

The answering party at 192.0.2.128 receives this offer and responds with the following answer:

```
v=0
o=bob 1129377364 1 IN IP4 192.0.2.128
s=-
c=IN IP4 192.0.2.128
t=0 0
m=video 40123 UDP/DCCP/RTP/AVP 99
a=rtpmap:99 h261/90000
a=dccp-service-code:SC:RTPV
a=dccp-port:9
a=setup:active
a=connection:new
```

Note that the "m=" line in the answer includes the UDP port number of the encapsulation service. The DCCP service code is set to "RTPV", signalled using the "a=dccp-service-code" attribute [RFC5762]. The "a=dccp-port:" attribute in the answer is set to 9 (the discard port) in the usual manner for an active connection-oriented endpoint.

The answering party will then attempt to establish a DCCP-UDP connection to the offering party. The connection request will use an ephemeral DCCP source port and DCCP destination port 5004. The UDP packet encapsulating that request will have UDP source port 40123 and UDP destination port 50234.
6. Security Considerations

DCCP-UDP provides all of the security risk-mitigation measures present in DCCP-STD, and also all of the security risks. It does not maintain additional state at the encapsulation layer.

The tunnel encapsulation recommends processing of ICMP messages received for packets sent using DCCP-UDP and translation to allow use by TCP. [RFC5927] describes precautions that are desirable before TCP acts on receipt of ICMP messages. Similar precautions are desirable for endpoints processing ICMP for DCCP-UDP. The purpose of DCCP-UDP is to allow DCCP to pass through NAT/NAPT devices, and therefore it exposes DCCP to the risks associated with passing through NAT devices. It does not create any new risks with regard to NAT/NAPT devices.

DCCP-UDP may also allow DCCP applications to pass through existing firewall devices using rules for UDP, if the administrators of the devices so choose. A simple use may either allow all DCCP applications or allow none.

A firewall that interprets this specification could inspect the encapsulated DCCP header to filter based on the inner DCCP header information. Full control of DCCP connections by applications will require enhancements to firewalls, as discussed in [RFC4340] and related RFCs (e.g. [RFC5595]).

Datagram Transport Layer Security (DTLS) TLS provides mechanisms that can be used to provide security protection for the encapsulated DCCP packets. DTLS may be used in two ways:

- Individual DCCP connections may be protected in the same way that DTLS is used with native DCCP [RFC5595]. This does not encrypt the UDP transport header added by DCCP-UDP.

- This specification also permits the use of DTLS with the UDP transport that encapsulates DCCP packets. When DTLS is used at the encapsulation layer this protects the DCCP headers. This prevents the headers from being inspected or updated by network middleboxes (such as firewalls and NAT). It also eliminates the need for a separate DTLS handshake for each DCCP connection.

7. IANA Considerations

This document requests IANA to make the allocations described in the following sections.
7.1. UDP Port Allocation

IANA is requested to allocate a UDP port for the DCCP-UDP service. This port is allocated for use by a transport service, rather than an application. In this case, the name of the transport should explicitly appear in the registry. Use of this port is defined in section Section 3.8

XXX Note: IANA is requested to replace all occurrences of "XXX IANA PORT XXX" by the allocated port value prior to publication. XXX

7.2. DCCP Reset

IANA is requested to assign a new DCCP Reset Code in the DCCP Reset Codes Registry, with the short description "Encapsulated Port Reuse". This code applies to all DCCP congestion control IDs and should be allocated a value less than 120 decimal. Use of this reset code is defined in section Section 3.8. Section 5.6 of RFC4340 defines three "Data" bytes that are carried by a DCCP Reset. For this Reset Code these are defined as below:

- Data byte 1: The DCCP Packet Type of the DCCP datagram that resulted in the error message.

- Data byte 2 & 3: The encapsulated Source UDP Port from the DCCP-UDP datagram that triggered the ICMP message, in network order.

XXX Note: IANA is requested to replace all occurrences of "XXX IANA Port Reuse XXX" by the allocated DCCP reset code value prior to publication. XXX

7.3. SDP Attribute Allocation

IANA is requested to allocate the following new SDP attribute ("att-field"):

- Contact name: DCCP Working Group
- Attribute name: dccp-port
- Long-form attribute name in English: Encapsulated DCCP Port
- Type of attribute: Media level
- Subject to charset attribute? No
- Purpose of the attribute: See this document, section Section 5.1
8. Acknowledgments

This document was produced by the DCCP WG. The following contributed during the working group last call:

Andrew Lentvorski, Lloyd Wood, Pasi Sarolahti, Gerrit Renker, Eddie Kohler, and Dan Wing.

9. References

9.1. Normative References


9.2. Informative References


Authors’ Addresses

Tom Phelan  
Sonus Networks  
7 Technology Dr.  
Westford, MA  01886  
US  
Phone: +1 978 614 8456  
Email: tphelan@sonusnet.com

Godred Fairhurst  
University of Aberdeen  
School of Engineering  
Fraser Noble Building  
Aberdeen, Scotland  AB24 3UE  
UK  
Email: gorry@erg.abdn.ac.uk  
URI:   http://www.erg.abdn.ac.uk

Colin Perkins  
University of Glasgow  
School of Computing Science  
Glasgow, Scotland  G12 8QQ  
UK  
Email: csp@csperskins.org  
URI:   http://csperkins.org/
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.
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).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Formula</th>
<th>Small packets</th>
<th>Large packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet size</td>
<td>s/8</td>
<td>60B</td>
<td>1,500B</td>
</tr>
<tr>
<td>Packet size</td>
<td>s</td>
<td>480b</td>
<td>12,000b</td>
</tr>
<tr>
<td>Bit-rate</td>
<td>x</td>
<td>48Mbps</td>
<td>48Mbps</td>
</tr>
<tr>
<td>Packet-rate</td>
<td>u = x/s</td>
<td>100kpps</td>
<td>4kpps</td>
</tr>
</tbody>
</table>

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:

- updating host-based congestion control protocols to take account of packet size
- making transports less sensitive to losing control packets like SYN 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
Internet-Draft   Byte and Packet Congestion Notification   November 2013

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
Internet-Draft   Byte and Packet Congestion Notification   November 2013

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:

- 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.
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:

- 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.

- 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:

- whether to measure queue length in bytes or packets (Section 4.1)
- 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:

- 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.

- 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/marking 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.

4.1.2. 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/marking 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 Cnodder 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 Cnodder’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

<table>
<thead>
<tr>
<th>transport cc</th>
<th>RED_1 (packet mode drop)</th>
<th>RED_4 (linear byte mode drop)</th>
<th>RED_5 (square byte mode drop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP or TFRC</td>
<td>s/sqrt(p)</td>
<td>sqrt(s/p)</td>
<td>1/sqrt(p)</td>
</tr>
<tr>
<td>TFRC-SP</td>
<td>1/sqrt(p)</td>
<td>1/sqrt(sp)</td>
<td>1/(s.sqrt(p))</td>
</tr>
</tbody>
</table>

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 Cnodder 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:

- How to handle any legacy of AQM with byte-mode drop already deployed;

- 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:
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.

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.

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 SYNs, 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


11.2. Informative References


[CCvarPktSize] Widmer, J., Boutremans, C., and J-Y. Le


[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.


RFC 2309, April 1998.


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.

<table>
<thead>
<tr>
<th>Response</th>
<th>No. of vendors</th>
<th>%age of vendors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not implemented</td>
<td>14</td>
<td>17%</td>
</tr>
<tr>
<td>Not implemented (probably)</td>
<td>2</td>
<td>2%</td>
</tr>
<tr>
<td>Implemented</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>No response</td>
<td>68</td>
<td>81%</td>
</tr>
<tr>
<td>Total companies/orgs surveyed</td>
<td>84</td>
<td>100%</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Network</th>
<th>Transport</th>
<th>a) Independent of packet size of congestion notifications</th>
<th>b) Dependent on packet size of congestion notifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Predominantly bit-congestible network</td>
<td>Scenario a1)</td>
<td>Scenario b1)</td>
<td></td>
</tr>
<tr>
<td>2) Mix of bit-congestible and pkt-congestible network</td>
<td>Scenario a2)</td>
<td>Scenario b2)</td>
<td></td>
</tr>
</tbody>
</table>

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 (96Mbps/26 = 4Mbps 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".
### Table 5: Absolute Loss Rates and Loss Ratios for Flows of Small and Large Packets and Both Combined

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Formula</th>
<th>Flow 1</th>
<th>Flow 2</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet size</td>
<td>s/8</td>
<td>60B</td>
<td>1,500B</td>
<td>(Mix)</td>
</tr>
<tr>
<td>Packet size</td>
<td>s</td>
<td>480b</td>
<td>12,000b</td>
<td>(Mix)</td>
</tr>
<tr>
<td>Pkt loss probability</td>
<td>p</td>
<td>0.1%</td>
<td>0.1%</td>
<td>0.1%</td>
</tr>
</tbody>
</table>

#### EQUAL BIT-RATE CASE

| Bit-rate                  | x         | 48Mbps | 48Mbps | 96Mbps   |
| Absolute pkt-loss-rate    | p*u       | 100pps | 4kpps  | 104kpps  |
| 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   |
| Absolute pkt-loss-rate    | p*u       | 8pps   | 8pps   | 15kpps   |
| Absolute bit-loss-rate    | p*u*s     | 4kpps  | 92kpps | 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%     |

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*480b and 8*12,000b 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 \( p(u_1s_1 + u_2s_2) \).

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 Jaeggl]:
  + 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.


* 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 SYNvs and pure ACKs more than equalising the bit rates of TCPvs 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/~jmannер/
Abstract

This document defines extensions to Integrated Services (IntServ) allowing multiple traffic specifications and multiple flow specifications to be conveyed in the same Resource Reservation Protocol (RSVPv1) reservation message exchange. This ability helps optimize an agreeable bandwidth through a network between endpoints in a single round trip.

Status of this Memo

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

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

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

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

Copyright Notice

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

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

1. Introduction ........................................... 3
2. Overview of the Proposal for including multiple TSPECs and FLOWSPECs .......... 6
3. MULTI_TSPEC and MULTI_FLOWSPEC Solution ............... 8
   3.1 New MULTI_TSPEC and MULTI-RSPEC Parameters .......... 9
   3.2 Multiple TSPEC in a PATH Message ...................... 9
   3.3 Multiple FLOWSPEC for Controlled Load Service .......... 12
   3.4 Multiple FLOWSPEC for Guaranteed Service .............. 14
4. Rules of Usage ........................................... 17
   4.1 Backward Compatibility ............................... 17
   4.2 Applies to Only a Single Session ...................... 17
   4.3 No Special Error Handling for PATH Message .......... 17
   4.4 Preference Order to be Maintained .................... 18
   4.5 Bandwidth Reduction in Downstream Routers ............ 18
   4.6 Merging Rules ....................................... 19
   4.7 Applicability to Multicast ............................ 19
   4.8 MULTI_TSPEC Specific Error .......................... 20
   4.9 Other Considerations ................................ 20
   4.10 Known Open Issues .................................. 21
5. Security considerations .................................. 21
6. IANA considerations ..................................... 22
7. Acknowledgments ......................................... 22
8. References ............................................... 22
   8.1 Normative References .................................. 23
   8.2. Informative References ............................... 23
   Authors’ Addresses ...................................... 23
   Appendix A. Alternatives for Sending Multiple TSPECs..... 23

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

This document defines how Integrated Services (IntServ) [RFC2210] includes multiple traffic specifications and multiple flow specifications in the same Resource Reservation Protocol (RSVPv1) [RFC2205] message. This ability helps optimize an agreeable bandwidth through a network between endpoints in a single round trip.

There is a separation of function between RSVP and IntServ, in which RSVP does not define the internal objects to establish controlled load or guarantee services. These are generally left to be opaque in RSVP. At the same time, IntServ does not require that RSVP be the only reservation protocol for transporting both the controlled load or guaranteed service objects - but RSVP does often carry the objects anyway. This makes the two independent - yet related in usage, but are also frequently talked about as if they are one and the same. They are not.

The ‘traffic specification’ contains the traffic characteristics of a sender’s data flow and is a required object in a PATH message. The TSPEC object is defined in RFC 2210 to convey the traffic specification from the sender and is opaque to RSVP. The ADSPEC object - for ‘advertising specification’ - is used to gather information along the downstream data path to aid the receiver in the computation of QoS properties of this data path. The ADSPEC is also opaque to RSVP and is defined in RFC 2210. Both of these IntServ objects are part of the Sender Descriptor [RFC2205].

Once the Sender Descriptor is received at its destination node, after having traveled through the network of routers, the SENDER_TSPEC information is matched with the information gathered in the ADSPEC, if present, about the data path. Together, these two objects help the receiver build its flow specification (encoded in the FLOWSPEC object) for the RESV message. The RESV message establishes the reservation through the network of routers on the data path established by the PATH message. If the ADSPEC is not present in the Sender_Descriptor, it cannot aid the receiver in building the flow specification.

The SENDER_TSPEC is not changed in transit between endpoints (i.e., there are no bandwidth request adjustments along the way). However, the ADSPEC is changed, based on the conditions experienced through the network (i.e., bandwidth availability within each router) as the RSVP message travels hop-by-hop.

Today, real-time applications have evolved such that they are able to dynamically adapt to available bandwidth, not only by dropping and adding layers, but also by reducing frame rates and resolution. It is therefore limiting to have a single bandwidth request in Integrated Services, and by extension, RSVP.
With only one traffic specification in a PATH message and only one flow specification in a RESV message (with some styles of reservations a RESV message may actually contain multiple flow specifications, but then there is only one per sender), applications will either have to give up altogether on session establishment in case of failure of the reservation establishment for the highest "bandwidth or will have to resort to multiple successive RSVP signaling attempts in a trial-and-error manner until they finally establish the reservation a lower "bandwidth". These multiple signaling round-trip would affect the session establishment time and in turn would negatively impact the end user experience.

The objective of this document is to avoid such roundtrips as well as allow applications to successfully receive some level of bandwidth allotment that it can use for its sessions.

While the ADSPEC provides an indication of the bandwidth available along the path and can be used by the receiver in creating the FLOWSPEC, it does not prevent failures or multiple round-trips as described above. The intermediary routers provide a best attempt estimate of available bandwidth in the ADSPEC object. However, it does not take into account external policy considerations (RFC 2215). In addition, the available bandwidth at the time of creating the ADSPEC may not be available at the time of an actual request in an RESV message. These reasons may cause the RESV message to be rejected. Therefore, the ADSPEC object cannot, by itself, satisfy the requirements of the current generations of real-time applications.

It needs to be noted that the ADSPEC is unchanged by this new mechanism. If ADSPEC is included in the PATH message, it is suggested that the receiver use this object in determining the flow specification.

This document creates a means for conveying more than one "bandwidth" within the same RSVP reservation set-up (both PATH and RESV) messages to optimize the determination of an agreed upon bandwidth for this reservation. Allowing multiple traffic specifications within the same PATH message allows the sender to communicate to the receiver multiple "bandwidths" that match the different sending rates that the sender is capable of transmitting at. This allows the receiver to convey this multiple "bandwidths" in the RESV so those can be considered when RSVP makes the actual reservation admission into the network. This allows the applications to dynamically adapt their data stream to available network resources.

The concept of RSVP signaling is shown in a single direction below, in Figure 1. Although the TSPEC is opaque to RSVP, it is shown along with the RSVP messages for completeness. The RSVP messages themselves need not be the focus of the reader. Instead, the
number of round trips it takes to establish a reservation is the focus here.

Sender       Rtr-1       Rtr-2 ... Rtr-N       Receiver
|              |          |          |               |
| PATH (with a TSPEC & ADSPEC) | ----/----> | ----/----> |
| <----/----< | <----/----< | <----/----< |

Figure 1. Concept of RSVP in a Single Direction

Figure 1 shows a successful one-way reservation using RSVP and IntServ.

Figure 2 shows a scenario where the RESV message, containing a FLOWSPEC, which is generated by the Receiver, after considering both the Sender TSPEC and the ADSPEC, is rejected by an intermediary router.

Sender       Rtr-1       Rtr-2 ... Rtr-N       Receiver
|              |          |          |               |
| PATH (with 1 TSPEC wanting 12Mbps) | ----/----> | ----/----> |
| RESV (with 1 FLOWSPEC wanting 12Mbps) | X <----/----< |
| ResvErr (with Admission control Error=2) | ----/----> |

Figure 2. Concept of RSVP Rejection due to Limited Bandwidth

The scenario above is where multiple TSPEC and multiple FLOWSPEC optimization helps. The Sender may support multiple bandwidths for a given application (i.e., more than one codec for voice or video) and therefore might want to establish a reservation with the highest (or best) bandwidth that the network can provide for a particular codec.

For example, bandwidths of:

12Mbps,
4Mbps, and
1.5Mbps

for the three video codecs the Sender supports.
This document will discuss the overview of the proposal to include multiple TSPECs and FLOWSPECs RSVP in section 2. In section 3, the overview of the entire solution is provided. This section also contains the new parameters which are defined in this document. The multiple TSPECs in a PATH message and the multiple FLOWSPEC in a RESV message, both for controlled load and guaranteed service are described in this section. Section 4 will cover the rules of usage of this IntServ extension. This section contains how this document needs to extend the scenario of when a router in the middle of a reservation cannot accept a preferred bandwidth (i.e., FLOWSPEC), meaning previous routers that accepted that greater bandwidth now have too much bandwidth reserved. This requires an extension to RFC 4495 (RSVP Bandwidth Reduction) to cover reservations being established, as well as existing reservations. Section 4 also includes the merging rules.

2. Overview of Proposal for Including Multiple TSPECs and FLOWSPECs

Presently, this is the format of a PATH message [RFC2205]:

\[
\text{<PATH Message> ::= <Common Header> [ <INTEGRITY> ]}
\]

\[
\text{<SESSION> <RSVP_HOP>}
\]

\[
\text{<TIME_VALUES>}
\]

\[
\text{[ <POLICY_DATA> ... ]}
\]

\[
\text{[ <sender descriptor> ]}
\]

\[
\text{<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>}
\]

\[
\text{[ <ADSPEC> ]}
\]

where the SENDER_TSPEC object contains a single traffic specification.

For the PATH message, the focus of this document is to modify the <sender_descriptor> in such a way to include more than one traffic specification. This solution does this by retaining the existing SENDER_TSPEC object above, highlighted by the ‘^^^^’ characters, and complementing it with a new optional MULTI_TSPEC object to convey additional traffic specifications in this PATH message. No other object within the PATH message is affected by this IntServ extension.

This extension modifies the sender descriptor by specifically augmenting it to allow an optional <MULTI_TSPEC> object after the optional <ADSPEC>, as shown below.
<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
[ <ADSPEC> ] [ <MULTI_TSPEC> ]

As can be seen above, the MULTI_TSPEC is in addition to the SENDER_TSPEC – and is only to be used, per this extension, when more than one TSPEC is to be included in the PATH message.

Here is another way of looking at the proposal choices:

```
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing TSPEC</td>
</tr>
<tr>
<td>+--------------+</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+--------------+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional TSPECs</td>
</tr>
<tr>
<td>+----------------+</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+----------------+</td>
</tr>
</tbody>
</table>
```

Figure 3. Encoding of Multiple Traffic Specifications in the TSPEC and MULTI_TSPEC objects

This solution is backwards compatible with existing implementations of [RFC2205] and [RFC2210], as the multiple TSPECs and FLOWSPECs are inserted as optional objects and such objects do not need to be processed, especially if they are not understood.

This solution defines a similar approach for encoding multiple flow specifications in the RESV message. Flow specifications beyond the first one can be encoded in a new "MULTI_FLOWSPEC" object contained
In this proposal, the original SENDER_TSPEC and the FLOWSPEC are left untouched, allowing routers not supporting this extension to process the PATH and the RESV message without issue. Two new additional objects are defined in this document. They are the MULTI_TSPEC and the MULTI_FLOWSPEC for the PATH and the RESV message, respectively. The additional TSPECs (in the new MULTI_TSPEC Object) are included in the PATH and the additional FLOWSPECS (in the new MULTI_FLOWSPEC Object) are included in the RESV message as new (optional) objects. These additional objects will have a class number of 11bbbbbb, allowing older routers to ignore the object(s) and forward each unexamined and unchanged, as defined in section 3.10 of [RFC 2205].

NOTE: it is important to emphasize here that including more than one FLOWSPEC in the RESV message does not cause more than one FLOWSPEC to be granted. This document requires that the receiver arrange these multiple FLOWSPECs in the order of preference according to the order remaining from the MULTI_TSPECs in the PATH message. The benefit of this arrangement is that RSVP does not have to process the rest of the FLOWSPEC if it can admit the first one.

3. MULTI_TSPEC and MULTI_FLOWSPEC Solution

For the Sender Descriptor within the PATH message, the original TSPEC remains where it is, and is untouched by this IntServ extension. What is new is the use of a new <MULTI_TSPEC> object inside the sender descriptor as shown here:

```
<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC> 
[ <ADSPEC> ] [ <MULTI_TSPEC> ] ^^^^^^^^^^^
```

The preferred order of TSPECs sent by the sender is this:

- preferred TSPEC is in the original SENDER_TSPEC
- the next in line preferred TSPEC is the first TSPEC in the MULTI_TSPEC object
- the next in line preferred TSPEC is the second TSPEC in the MULTI_TSPEC object
- and so on...

The composition of the flow descriptor list in a Resv message depends upon the reservation style. Therefore, the following shows
the inclusion of the MULTI_FLOWSPEC object with each of the styles:

WF Style:
<flow descriptor list> ::=  <WF flow descriptor>
<WF flow descriptor> ::= <FLOWSPEC> [MULTI_FLOWSPEC]

FF style:
<flow descriptor list> ::= 
  <FLOWSPEC>  <FILTER_SPEC>  [MULTI_FLOWSPEC] |
  <flow descriptor list> <FF flow descriptor>
<FF flow descriptor> ::= 
  [ <FLOWSPEC> ] <FILTER_SPEC> [MULTI_FLOWSPEC]

SE style:
<flow descriptor list> ::= <SE flow descriptor>
<SE flow descriptor> ::= 
  <FLOWSPEC>  <filter spec list> [MULTI_FLOWSPEC]
<filter spec list> ::= <FILTER_SPEC> 
  | <filter spec list> <FILTER_SPEC>

3.1 New MULTI_TSPEC and MULTI-RSPEC Parameters

This extension to Integrated Services defines two new parameters. They are:

1. <parameter name> Multiple_Token_Bucket_Tspec, with a parameter number of 125.

2. <parameter name> Multiple_Guaranteed_Service_RSpec with a parameter number of 124

These are IANA registered in this document.

The original SENDER_TSPEC and FLOWSPEC for Controlled Service maintain the <parameter name> of Token_Bucket_Tspec with a parameter number of 127. The original FLOWSPEC for Guaranteed Service maintains the <parameter name> of Guaranteed_Service_RSpec with a parameter number of 130.

3.2 Multiple TSPEC in a PATH Message
Here is the object from [RFC2210]. It is used as a SENDER_TSPEC in a PATH message:

```
31           24 23           16 15            8 7             0
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
1   | 0 (a) |    reserved           |            7 (b)             |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
2   |    X  (c)     |0| reserved    |            6 (d)             |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
3   |   127 (e)     |    0 (f)      |            5 (g)             |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
4   |  Token Bucket Rate [r] (32-bit IEEE floating point number) |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
5   |  Token Bucket Size [b] (32-bit IEEE floating point number) |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
6   |  Peak Data Rate [p] (32-bit IEEE floating point number)    |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
7   |  Minimum Policed Unit [m] (32-bit integer)                  |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
8   |  Maximum Packet Size [M] (32-bit integer)                   |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
```

Figure 4. SENDER_TSPEC in PATH

(a) - Message format version number (0)
(b) - Overall length (7 words not including header)
(c) - Service header, service number
   - '1′ (Generic information) if in a PATH message;
(d) - Length of service data, 6 words not including per-service header
(e) - Parameter ID, parameter 127 (Token Bucket TSpec)
(f) - Parameter 127 flags (none set)
(g) - Parameter 127 length, 5 words not including per-service header

For completeness, Figure 4 is included in its original form for backwards compatibility reasons, as if there were only 1 TSPEC in the PATH. What is new when there are more than one TSPEC in this reservation message is the new MULTI_TSPEC object in Figure 5 containing, for example, 3 (Multiple_Token_Bucket_Tspec) TSPECs in a PATH message.

```
31           24 23           16 15            8 7             0
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
1   | 0 (a) |    reserved           |            19 (b)             |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
2   |    5  (c)     |0| reserved    |            18 (d)             |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
3   |   125 (e)     |    0 (f)      |            5 (g)             |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
4   |  Token Bucket Rate [r] (32-bit IEEE floating point number) |
+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+−+
```
Figure 5. MULTI_TSPEC Object

(a) - Message format version number (0)
(b) - Overall length (19 words not including header)
(c) - Service header, service number 5 (Controlled-Load)
(d) - Length of service data, 18 words not including per-service header
(e) - Parameter ID, parameter 125 (Multiple Token Bucket TSpec)
(f) - Parameter 125 flags (none set)
(g) - Parameter 125 length, 5 words not including per-service header

Figure 5 shows the 2nd through Nth TSPEC in the PATH in the preferred order. The message format (a) remains the same for a second TSPEC and for other additional TSPECs.

The Overall Length (b) includes all the TSPECs within this object, plus the 2nd Word (containing fields (c) and (d)), which MUST NOT be repeated. The service header fields (e), (f) and (g) are repeated for
Each TSPEC is six 32-bit Words long (the per-service header plus the 5 values that are 1 Word each in length), therefore the length is in 6 Word increments for each additional TSPEC. Case in point, from the above Figure 5, Words 3-8 are the first TSPEC (2nd preferred), Words 9-14 are the next TSPEC (3rd preferred), and Words 15-20 are the final TSPEC (and 4th preferred) in this example of 3 TSPECs in this MULTI_TSPEC object. There is no limit placed on the number of TSPECs a MULTI_TSPEC object can have. However, it is RECOMMENDED to administratively limit the number of TSPECs in the MULTI_TSPEC object to 9 (making for a total of 10 in the PATH message).

The TSPECs are included in the order of preference by the message generator (PATH) and MUST be maintained in that order all the way to the Receiver. The order of TSPECs that are still grantable, in conjunction with the ADSPEC at the Receiver, MUST retain that order in the FLOWSPEC and MULTI_FLOWSPEC objects.

3.3 Multiple FLOWSPEC for Controlled-Load service

The format of an RSVP FLOWSPEC object requesting Controlled-Load service is the same as the one used for the SENDER_TSPEC given in Figure 4.

The format of the new MULTI_FLOWSPEC object is given below:

```
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
|     31             |      24           |       23          |       16          |       15          |        8          |        7          |        0          |                   |
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| 0 (a) | reserved        | 19 (b)           |                   |                   |                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| 5 (c) | 0 reserved      | 18 (d)           |                   |                   |                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| 125 (e) | 0 (f) | 5 (g)           |                   |                   |                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| Token Bucket Rate [r] (32-bit IEEE floating point number) |                   |                   |                   |                   |                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| Token Bucket Size [b] (32-bit IEEE floating point number) |                   |                   |                   |                   |                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| Peak Data Rate [p] (32-bit IEEE floating point number) |                   |                   |                   |                   |                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| Minimum Policed Unit [m] (32-bit integer) |                   |                   |                   |                   |                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| Maximum Packet Size [M] (32-bit integer) |                   |                   |                   |                   |                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| 125 (e) | 0 (f) | 5 (g)           |                   |                   |                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
```
<table>
<thead>
<tr>
<th></th>
<th>Token Bucket Rate ([r]) (32-bit IEEE floating point number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>+----------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Token Bucket Size ([b]) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td></td>
<td>+----------------------------------------------------------------</td>
</tr>
<tr>
<td>11</td>
<td>Peak Data Rate ([p]) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td></td>
<td>+----------------------------------------------------------------</td>
</tr>
<tr>
<td>12</td>
<td>Minimum Policed Unit ([m]) (32-bit integer)</td>
</tr>
<tr>
<td></td>
<td>+----------------------------------------------------------------</td>
</tr>
<tr>
<td>13</td>
<td>Maximum Packet Size ([M]) (32-bit integer)</td>
</tr>
<tr>
<td></td>
<td>+----------------------------------------------------------------</td>
</tr>
<tr>
<td>14</td>
<td>125 (e)</td>
</tr>
<tr>
<td></td>
<td>+----------------------------------------------------------------</td>
</tr>
<tr>
<td>15</td>
<td>Token Bucket Rate ([r]) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td></td>
<td>+----------------------------------------------------------------</td>
</tr>
<tr>
<td>16</td>
<td>Token Bucket Size ([b]) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td></td>
<td>+----------------------------------------------------------------</td>
</tr>
<tr>
<td>17</td>
<td>Peak Data Rate ([p]) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td></td>
<td>+----------------------------------------------------------------</td>
</tr>
<tr>
<td>18</td>
<td>Minimum Policed Unit ([m]) (32-bit integer)</td>
</tr>
<tr>
<td></td>
<td>+----------------------------------------------------------------</td>
</tr>
<tr>
<td>19</td>
<td>Maximum Packet Size ([M]) (32-bit integer)</td>
</tr>
<tr>
<td></td>
<td>+----------------------------------------------------------------</td>
</tr>
</tbody>
</table>

Figure 5. Multiple FLOWSPEC for Controlled-Load service

(a) − Message format version number (0)
(b) − Overall length (19 words not including header)
(c) − Service header, service number 5 (Controlled-Load)
(d) − Length of controlled-load data, 18 words not including per-service header
(e) − Parameter ID, parameter 125 (Multiple Token Bucket TSPEC)
(f) − Parameter 125 flags (none set)
(g) − Parameter 125 length, 5 words not including per-service header

This is for the 2nd through Nth TSPEC in the RESV, in the preferred order.

The message format (a) remains the same for a second TSPEC and for additional TSPECs.

The Overall Length (b) includes the TSPECs, plus the 2nd Word fields (c) and (d), which MUST NOT be repeated. The service header fields (e),(f) and (g), which are repeated for each TSPEC.

The Service header, here service number 5 (Controlled-Load) MUST remain the same for the RESV message. The services, Controlled-Load and Guaranteed MUST NOT be mixed within the same RESV message. In other words, if one TSPEC is a Controlled Load service TSPEC, the remaining TSPECs MUST be Controlled Load service. This same rule also is true for Guaranteed Service - if one TSPEC is for Guaranteed
Service, the rest of the TSPECs in this PATH or RESV MUST be for Guaranteed Service.

The Length of controlled-load data (d) also increases to account for the additional TSPECs.

Each FLOWSPEC is six 32-bit Words long (the per-service header plus the 5 values that are 1 Word each in length), therefore the length is in 6 Word increments for each additional TSPEC. Case in point, from the above Figure 5, Words 3-8 are the first TSPEC (2nd preferred), Words 9-14 are the next TSPEC (3rd preferred), and Words 15-20 are the final TSPEC (and 4th preferred) in this example of 3 TSPECs in this FLOWSPEC. There is no limit placed on the number of TSPECs a particular FLOWSPEC can have.

Within the MULTI_FLOWSPEC, any SENDER_TSPEC that cannot be reserved - based on the information gathered in the ADSPEC, is not placed in the RESV or based on other information available to the receiver. Otherwise, the order in which the TSPECs were in the PATH message MUST be in the same order they are in the FLOWSPEC in the RESV. This is the order of preference of the sender, and MUST be maintained throughout the reservation establishment, unless the ADSPEC indicates one or more TSPECs cannot be granted, or the receiver cannot include any TSPEC due to technical or administrative constraints or one or more routers along the RESV path cannot grant a particular TSPEC. At any router that a reservation cannot honor a TSPEC, this TSPEC MUST be removed from the RESV, or else another router along the RESV path might reserve that TSPEC. This rule ensures this cannot happen.

Once one TSPEC has been removed from the RESV, the next in line TSPEC becomes the preferred TSPEC for that reservation. That router MUST generate a ResvErr message, containing an ERROR_SPEC object with a Policy Control Failure with Error code = 2 (Policy Control Failure), and an Error Value sub-code 102 (ERR_PARTIAL_PREEMPT) to the previous routers, clearing the now over allocation of bandwidth for this reservation. The difference between the previously accepted TSPEC bandwidth and the currently accepted TSPEC bandwidth is the amount this error identifies as the amount of bandwidth that is no longer required to be reserved. The ResvErr and the RESV messages are independent, and not normally sent by the same router. This aspect of this document is the extension to RFC 2205 (RSVP).

If a RESV cannot grant the final TSPEC, normal RSVP rules apply with regard to the transmission of a particular ResvErr.

3.4 Multiple FLOWSPEC for Guaranteed service

The FLOWSPEC object, which is used to request guaranteed service contains a TSPEC and RSpec. Here is the FLOWSPEC object from [RFC2215] when requesting Guaranteed service:
### Figure 6. FLOWSPEC for Guaranteed service

(a) - Message format version number (0)  
(b) - Overall length (9 words not including header)  
(c) - Service header, service number 2 ( Guaranteed)  
(d) - Length of per-service data, 9 words not including per-service header  
(e) - Parameter ID, parameter 127 (Token Bucket TSpec)  
(f) - Parameter 127 flags (none set)  
(g) - Parameter 127 length, 5 words not including parameter header  
(h) - Parameter ID, parameter 130 (Guaranteed Service RSpec)  
(i) - Parameter xxx flags (none set)  
(j) - Parameter xxx length, 2 words not including parameter header  

The difference in structure between the Controlled-Load FLOWSPEC and Guaranteed FLOWSPEC is the RSPEC, defined in [RFC2212].  

For completeness, Figure 6 is included in its original form for backwards compatibility reasons, as if there were only 1 FLOWSPEC in the RESV. What is new when there is more than one TSPEC in the FLOWSPEC in a RESV message is the new MULTI_FLOWSPEC object in Figure 7 containing, for example, 3 FLOWSPECs requesting Guaranteed Service.
<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>Unused</td>
</tr>
<tr>
<td>24</td>
<td>Unused</td>
</tr>
<tr>
<td>23</td>
<td>Unused</td>
</tr>
<tr>
<td>16</td>
<td>Unused</td>
</tr>
<tr>
<td>15</td>
<td>Unused</td>
</tr>
<tr>
<td>8</td>
<td>Unused</td>
</tr>
<tr>
<td>7</td>
<td>Unused</td>
</tr>
<tr>
<td>0</td>
<td>Unused</td>
</tr>
<tr>
<td>2</td>
<td>Unused</td>
</tr>
<tr>
<td>1</td>
<td>Unused</td>
</tr>
<tr>
<td></td>
<td>Unused</td>
</tr>
<tr>
<td>125</td>
<td>Token Bucket Rate (r) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Token Bucket Size (b) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Peak Data Rate (p) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Minimum Policed Unit (m) (32-bit integer)</td>
</tr>
<tr>
<td>1</td>
<td>Maximum Packet Size (M) (32-bit integer)</td>
</tr>
<tr>
<td>1</td>
<td>Rate (R) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Slack Term (S) (32-bit integer)</td>
</tr>
<tr>
<td>1</td>
<td>Token Bucket Rate (r) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Token Bucket Size (b) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Peak Data Rate (p) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Minimum Policed Unit (m) (32-bit integer)</td>
</tr>
<tr>
<td>1</td>
<td>Maximum Packet Size (M) (32-bit integer)</td>
</tr>
<tr>
<td>1</td>
<td>Rate (R) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Slack Term (S) (32-bit integer)</td>
</tr>
<tr>
<td>1</td>
<td>Token Bucket Rate (r) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Token Bucket Size (b) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Peak Data Rate (p) (32-bit IEEE floating point number)</td>
</tr>
<tr>
<td>1</td>
<td>Minimum Policed Unit (m) (32-bit integer)</td>
</tr>
</tbody>
</table>
Figure 7. Multiple FLOWSPECs for Guaranteed service

(a) - Message format version number (0)
(b) - Overall length (9 words not including header)
(c) - Service header, service number 2 (Guaranteed)
(d) - Length of per-service data, 9 words not including per-service header
(e) - Parameter ID, parameter 125 (Token Bucket TSpec)
(f) - Parameter 125 flags (none set)
(g) - Parameter 125 length, 5 words not including parameter header
(h) - Parameter ID, parameter 124 (Guaranteed Service RSpec)
(i) - Parameter 124 flags (none set)
(j) - Parameter 124 length, 2 words not including parameter header

There MUST be 1 RSpec per TSPEC for Guaranteed Service. Therefore, there are 5 words for Receiver TSPEC and 3 words for the RSpec. Therefore, for Guaranteed Service, the TSPEC/RSPEC combination occurs in increments of 8 words.

4. Rules of Usage

The following rules apply to nodes adhering to this specification:

4.1 Backward Compatibility

If the recipient does not understand this extension, it ignores this MULTI_TSPEC object, and operates normally for a node receiving this RSVP message.

4.2 Applies to Only a Single Session

When there is more than one TSPEC object or more than one FLOWSPEC object, this MUST NOT be considered for more than one flow created. These are OR choices for the same flow of data. In order to attain three reservations between two endpoints, three different reservation requests are required, not one reservation request with 3 TSPECs.
4.3 No Special Error Handling for PATH Message

If a problem occurs with the PATH message - regardless of this extension, normal RSVP procedures apply (i.e., there is no new PathErr code created within this extension document) - resulting in a PathErr message being sent upstream towards the sender, as usual.

4.4 Preference Order to be Maintained

When more than one TSPEC is in a PATH message, the order of TSPECs is decided by the Sender and MUST be maintained within the SENDER_TSPEC. The same order MUST be carried to the FLOWSPECs by the receiver. No additional TSPECs can be introduced by the receiver or any router processing these new objects. The deletion of TSPECs from a PATH message is not permitted. The deletion of the TSPECs when forming the FLOWSPEC is allowed by the receiver in the following cases:

- If one or more preferred TSPECs cannot be granted by a router as discovered during processing of the ADSPEC by the receiver, then they can be omitted when creating the FLOWSPEC(s) from the TSPECs.

- If one or more TSPECs arriving from the sender is not preferred by the receiver, then the receiver MAY omit any while creating the FLOWSPEC. A good reason to omit a TSPEC is if, for example, it does not match a codec supported by the receiver’s application(s).

The deletion of the TSPECs in the router during the processing of this MULTI_FLOWSPEC object is allowed in the following cases:

- If the original FLOWSPEC cannot be granted by a router then the router may discard that FLOWSPEC and replace it with the topmost FLOWSPEC from the MULTI_FLOWSPEC project. This will cause the topmost FLOWSPEC in the MULTI_FLOWSPEC object to be removed. The next FLOWSPECs becomes the topmost FLOWSPEC.

- If the router merges multiple RESV into a single RESV message, then the FLOWSPEC and the multiple FLOWSPEC may be affected

The preferred order of the remaining TSPECs or FLOWSPECs MUST be kept intact both at the receiver as well as the router processing these objects.

4.5 Bandwidth Reduction in Downstream Routers

If there are multiple FLOWSPECs in a single RESV message, it is quite possible that a higher bandwidth is reserved at a previous downstream device. Thus, any device that grants a reservation that is not the highest will have to inform the previous downstream routers to reduce the bandwidth reserved for this particular
The bandwidth reduction RFC [RFC4495] does not address the need that this document addresses. RFC 4495 defines an ability to preempt part of an existing reservation so as to admit a new incoming reservation with a higher priority, in lieu of tearing down the whole reservation having a lower priority. It does not specify the capability to reduce the bandwidth a RESV set up along the data path before the reservation is realized (from source to destination), when a subsequent router cannot support a more preferred FLOWSPEC contained in that RESV. This document extends the RFC 4495 defined partial teardown error to reduce bandwidth from previous downstream hops while a reservation is being established.

For example, if a 12Mbps TSPEC were granted for a reservation on previous hops, but could not be granted at the current hop, while the 4Mbps TSPEC could be granted (provided there is a MULTI_TSPEC with a 4Mbps TSPEC), this modification to the bandwidth reduction function would work by having the 4Mbps granting node send a reduction error to the downstream routers that installed 12Mbps for this reservation, thus clearing bandwidth that is now unnecessarily installed for a 4Mbps reservation.

4.6 Merging Rules

RFC 2205 defines the rules for merging as combining more than one FLOWSPEC into a single FLOWSPEC. In the case of MULTI_FLOWSPECs, merging of the two (or more) MULTI_FLOWSPEC MUST be done to arrive at a single MULTI_FLOWSPEC. The merged MULTI_FLOWSPEC will contain all the flow specification components of the individual MULTI_FLOWSPECs in descending orders of bandwidth. In other words, the merged FLOWSPEC MUST maintain the relative order of each of the individual FLOWSPECs. For example, if the individual FLOWSPEC order is 1,2,3 and another FLOWSPEC is a,b,c, then this relative ordering cannot be altered in the merged FLOWSPEC.

A byproduct of this is the ordering between the two individual FLOWSPECs cannot be signaled with this extension. If two (or more) FLOWSPECs have the same bandwidth, they are to be merged into one FLOWSPEC using the rules defined in RFC 2205. It is RECOMMENDED that the following rules are used for determining ordering (in TSPEC and FLOWSPEC):

For Controlled Load − in descending order of BW based on the Token Bucket Rate ‘r’ parameter value

For Guaranteed Service − in descending order of BW based on the RSPEC Rate ‘R’ parameter value

The resultant FLOWSPEC is added to the MULTI_FLOWSPEC based on its bandwidth in descending orders of bandwidth.
As a result of such merging, the number of FLOWSPECs in a MULTI_FLOWSPEC object should be the sum of the number of FLOWSPECs from individual MULTI_FLOWSPEC that have been merged *minus* the number of duplicates.

### 4.7 Applicability to Multicast

An RSVP message with a MULTI_TSPEC works just as well in a multicast scenario as it does in a unicast scenario. In a multicast scenario, the bandwidth allotted in each hop is the lowest bandwidth that can be admitted along the various path. For example:

```
+------|sender|======|Router-1|=====>|Router-2|=====>|Receiver-A|
+------|------|------|-------|-------|-------|-------|
                    V
                      +----|Receiver-C|
                      +----|
                      V
                      +----|Receiver-B|
                      +----|
```

Figure 8. MULTI_TSPEC and Multicast

If the sender (in Figure 8) sends 3 TSPECs (i.e., 1 TSPEC Object, and 2 in the MULTI_TSPEC Object) of 12Mbps, 5Mbps and 1.5Mbps. Let us say the path from Receiver-B to Router-1 admitted 5Mbps, Receiver-C to Router-2 admitted 1.5Mbps and Receiver-A to Router-2 admitted 12Mbps.

When the Resv message is send upstream from Router-2, the combining of 1.5Mbps (to Receiver-C) and 12Mbps (to Receiver-A) will be resolved to 1.5Mbps (lowest that can be admitted). Only a Resv with 1.5Mbps will be sent upstream from Router-2. Likewise, at Router-1, the combining of 1.5Mbps (to Router-2) and 5Mbps (to Receiver-B) will be resolved to 1.5Mbps units.

This is to allow the sender to transmit the flow at a rate that can be accepted by all devices along the path. Without this, if Router-2 receives a flow of 12Mbps, it will not know how to create a flow of 1.5Mbps down to Receiver-B. A differentiated reservation for the various paths along a multicast path is only possible with a Media-aware network device (MANE). The discussion of MANE and how it relates to admission control is outside the scope of this draft.
4.8 MULTI_TSPEC Specific Error

Since this mechanism is backward compatible, it is possible that a router without support for this MULTI_TSPEC extension will reject a reservation because the bandwidth indicated in the primary FLOWSPECs is not available. This means that an attempt with a lower bandwidth might have been successful, if one were included in a MULTI_TSPEC Object. Therefore, one should be able to differentiate between an admission control error where there is insufficient bandwidth when all the FLOWSPECs are considered and insufficient bandwidth when only the primary FLOWSPEC is considered.

This requires the definition of an error code within the ERROR_SPEC Object. When a router does not have sufficient bandwidth even after considering all the FLOWSPEC provided, it issues a new "MULTI_TSPEC bandwidth unavailable" error. This will be an Admission Control Failure (error #1), with a subcode of 6. A router that does not support this MULTI_TSPEC extension will return the "requested bandwidth unavailable" error as defined in RFC 2205 as if there was no MULTI_TSPEC in the message.

4.9 Other Considerations

- RFC 4495 articulates why a ResvErr is more appropriate to use for reducing the bandwidth of an existing reservation vs. a ResvTear.

- Refreshes only include the TSPECs that were accepted. One SHOULD be sent immediately upon the Sender receiving the RESV, to ensure all routers in this flow are synchronized with which TSPEC is in place.

- Periodically, it might be appropriate to attempt to increase the bandwidth of an accepted reservation with one of the TSPECs that were not accepted by the network when the reservation was first installed. This SHOULD NOT occur too regularly. This document currently offers no guidance on the frequency of this bump request for a rejected TSPEC from the PATH.

4.10 Known Open Issues

Here are the know open issues within this document:

- Need to ensure the cap on the number of TSPECs and FLOWSPECs is viable, yet controlled.
5. Security considerations

The security considerations for this document do not exceed what is already in RFC 2205 (RESV) or RFC 2210 (IntServ), as nothing in either of those documents prevent a node from requesting a lot of bandwidth in a single TSPEC. This document merely reduces the signaling traffic load on the network by allowing many requests that fall under the same policy controls to be included in a single round-trip message exchange.

Further, this document does not increase the security risk(s) to that defined in RFC 4495, where this document creates additional meaning to the RFC 4495 created error code 102.

A misbehaving Sender can include too many TSPECs in the MULTI_TSPEC object, which can lead to an amplification attack. That said, a bad implementation can create a reservation for each TSPEC received from within the Resv message. The number of TSPECs in the new MULTI_TSPEC object is limited, and the spec clearly states that only a single reservation is to be set up per Resv message.

To ensure the integrity of RSVP, the RSVP Authentication mechanisms defined in [RFC2747] and [RFC3097] SHOULD be used. Those protect RSVP message integrity hop-by-hop and provide node authentication as well as replay protection, thereby protecting against corruption and spoofing of RSVP messages.

6. IANA considerations

This document IANA registers the following new parameter name in the Integ-serv assignments at [IANA]:

Registry Name: Parameter Names
Registry:
Value     Description                                   Reference
--------     -------------------------------------------      -------
125       Multiple-Token_Bucket_Tspec                   [RFCXXXX]
124       Multiple_Guaranteed_Service_RSpec             [RFCXXXX]

Where RFCXXXX is replaced with the RFC number assigned to this Document.

This document IANA registers the following new error subcode in the Error code section, under the Admission Control Failure (error=1), of the rsvp-parameters assignments at [IANA]:

Registry Name: Error Codes and Globally-Defined Error Value Sub-Codes
Registry:
"Admission Control Failure"
Error Subcode  meaning                                    Reference

6    =    MULTI_TSPEC bandwidth unavailable          [RFCXXXX]

7. Acknowledgments

The authors wish to thank Fred Baker, Joe Touch, Bruce Davie, Dave Oran, Ashok Narayanan, Lou Berger, Lars Eggert, Arun Kudur, Ken Carlberg and Janet Gunn for their helpful comments and guidance in this effort.

And to Francois Le Faucheur, who provided text in this version.

8. References

8.1. Normative References


8.2. Informative References

Appendix A: Alternatives for Sending Multiple TSPECs

This appendix describes the discussion within the TSVWG of which approach best fits the requirements of sending multiple TSPECs within a single PATH or RESV message. There were 3 different options proposed, of which − 2 were insufficient or caused more harm than other options.

Looking at the format of a PATH message [RFC2205] again:

\[
<\text{PATH Message}> ::= \langle \text{Common Header} \rangle \ [ \langle \text{INTEGRITY} \rangle \ ] \\
<\text{SESSION}> \ <\text{RSVP\_HOP}> \\
<\text{TIME\_VALUES}> \\
[ \ <\text{POLICY\_DATA} \ \ldots \ ] \\
[ \ <\text{sender descriptor}> \ ] \\
<\text{sender descriptor}> ::= \langle \text{SENDER\_TEMPLATE} \rangle \ <\text{SENDER\_TSPEC}> \\
\ ^^^^^^^^^^^^^ \\
[ \ <\text{ADSPEC}\rangle \ ]
\]

For the PATH message, the focus of this document is with what to do with respect to the <SENDER_TSPEC> above, highlighted by the ‘^^^^’ characters. No other object within the PATH message will be affected by this IntServ extension.

The ADSPEC is optional in IntServ; therefore it might or might not be in the RSVP PATH message. Presently, the SENDER_TSPEC is limited to one bandwidth associated with the session. This is changed in this extension to IntServ to multiple bandwidths for the same session. There are multiple options on how the additional bandwidths...
Option #1 - creating the ability to add one or more additional (and complete) SENDER_TSPECs,

or

Option #2 - create the ability for the one already allowed SENDER_TSPEC to carry more than one bandwidth amount for the same reservation.

or

Option #3 - create the ability for the existing SENDER_TSPEC to remain unchanged, but add an optional `<MULTI_TSPEC>` object to the `<sender descriptor>` such as this:

```
<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC> [ <ADSPEC> ] [ <MULTI_TSPEC> ]
```

Here is another way of looking at the option choices:

<table>
<thead>
<tr>
<th>Option#1</th>
<th>Option#2</th>
<th>Option#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSPEC1</td>
<td>MULTI_TSPEC</td>
<td>TSPEC1</td>
</tr>
<tr>
<td>TSPEC2</td>
<td>TSPEC1</td>
<td>TSPEC1</td>
</tr>
<tr>
<td>TSPEC3</td>
<td>TSPEC2</td>
<td>MULTI_TSPEC</td>
</tr>
<tr>
<td>TSPEC4</td>
<td>TSPEC3</td>
<td>TSPEC2</td>
</tr>
<tr>
<td></td>
<td>TSPEC4</td>
<td>TSPEC3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TSPEC4</td>
</tr>
</tbody>
</table>

Figure 3. Concept of Option Choice

Option #1 and #2 do not allow for backward compatibility. If the currently used SENDER_TSPEC and FLOWSPEC objects are changed, then unless all the routers requiring RSVP processing are upgraded, this
functionality cannot be realized. As it is unlikely that all routers along the path will have the necessary enhancements as per this extension at one given time, therefore, it is necessary this enhancement be made in a way that is backward compatible. Therefore, option #1 and option #2 has been discarded in favor of option #3, which had WG consensus in a recent IETF meeting.

Option #3: This option has the advantage of being backwards compatible with existing implementations of [RFC2205] and [RFC2210], as the multiple TSPECs and FLOWSPECs are inserted as optional objects and such objects do not need to be processed, especially if they are not understood.

Option #3 applies to the FLOWSPEC contained in the RESV message as well. In this option, the original SENDER_TSPEC and the FLOWSPEC are left untouched, allowing routers not supporting this extension to be able to process the PATH and the RESV message without issue. Two new additional objects are defined in this document. They are the MULTI_TSPEC and the MULTI_FLOWSPEC for the PATH and the RESV message, respectively. The additional TSPECs (in the new MULTI_TSPEC Object) are included in the PATH and the additional FLOWSPECs (in the new MULTI_FLOWSPEC Object) are included in the RESV message as new (optional) objects. These additional objects will have a class number of 11bbbbbb, allowing older routers to ignore the object(s) and forward each unexamined and unchanged, as defined in section 3.10 of [RFC 2205].

We state in the document body that the top most FLOWSPEC of the new MULTI_FLOWSPEC Object in the RESV message replaces the existing FLOWSPEC when it is determined by the receiver (perhaps along with the ADSPEC) that the original FLOWSPEC cannot be granted. Therefore, the ordering of preference issue is solved with Option#3 as well.

NOTE: it is important to emphasize here that including more than one FLOWSPEC in the RESV message does not cause more than one FLOWSPEC to be granted. This document requires that the receiver arrange these multiple FLOWSPECs in the order of preference according to the order remaining from the MULTI_TSPECs in the PATH message. The benefit of this arrangement is that RSVP does not have to process the rest of the FLOWSPEC if it can admit the first one.

Additional details of these options can be found in the draft-polk-tsvwg-...-01 version of this appendix (which includes the RSVP bit mapping of fields in the TSPECs, if the reader wishes to search for that doc.)
Stream Control Transmission Protocol (SCTP) Network Address Translation Support

draft-ietf-tsvwg-natsupp-02.txt

Abstract

Stream Control Transmission Protocol [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 for TCP that allows multiple hosts to reside behind a NAT and yet use only a single globally unique IPv4 address, even when two hosts (behind a NAT) choose the same port numbers for their connection. This additional code is sometimes classified as Network Address and Port Translation (NAPT). To date, specialized code for SCTP has not yet been added to most NATs so that only pure NAT is available. The end result of this is that only one SCTP capable host can be behind a NAT.

This document describes the protocol extensions required for the SCTP endpoints to help NAT’s provide similar features of NAPT in the single-point and multi-point traversal scenario.

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 13, 2012.
Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the
document authors. All rights reserved.

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

Table of Contents

1. Introduction .................................................. 3
2. Conventions .................................................. 4
3. Terminology .................................................. 4
4. Problem Space Overview ....................................... 5
5. Association Setup Considerations .............................. 5
6. Handling of Internal Port Number and Verification Tag
   Collisions .................................................. 6
7. Handling of Internal Port Number Collisions .................. 7
8. Handling of Missing State .................................... 9
9. Multi Point Traversal Considerations .......................... 11
10. Socket API Considerations ................................... 11
10.1. Get or Set the NAT Friendliness
      (SCTP_NAT_FRIENDLY) .................................. 11
11. IANA Considerations ......................................... 12
11.1. New Chunk Flags for Two Chunk Types ..................... 12
11.2. Three New Error Causes .................................. 13
11.3. Two New Chunk Parameter Types ............................ 13
12. Security Considerations ...................................... 14
13. Acknowledgments ............................................. 14
14. References .................................................. 14
14.1. Normative References .................................... 14
14.2. Informative References .................................. 14
Authors’ Addresses .............................................. 15
1. Introduction

Stream Control Transmission Protocol [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 for TCP that allows multiple hosts to reside behind a NAT using private addresses (see [RFC5735]) and yet use only a single globally unique IPv4 address, even when two hosts (behind a NAT) choose the same port numbers for their connection. This additional code is sometimes classified as Network Address and Port Translation (NAPT). To date, specialized code for SCTP has not yet been added to most NATs so that only true NAT is available. The end result of this is that only one SCTP capable host can be behind a NAT.

This document describes an SCTP specific chunks and procedures to help NAT’s provide similar features of NAPT in the single point and multi-point traversal scenario. An SCTP implementation supporting this extension will follow these procedures to assure that in both single homed and multi-homed cases a NAT will maintain the proper state without needing to change port numbers.

A NAT will need to follow these procedures for generating appropriate SCTP packet formats. NAT’s should refer to [I-D.ietf-behave-sctpnat] for the BCP in using these formats.

When considering this feature it is possible to have multiple levels of support. At each level, the Internal Host, External Host and NAT may or may not support the features described in this document. The following table illustrates the results of the various combinations of support and if communications can occur between two endpoints.

<table>
<thead>
<tr>
<th>Internal Host</th>
<th>NAT</th>
<th>External Host</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support</td>
<td>Support</td>
<td>Support</td>
<td>Yes</td>
</tr>
<tr>
<td>Support</td>
<td>Support</td>
<td>No Support</td>
<td>Limited</td>
</tr>
<tr>
<td>Support</td>
<td>No Support</td>
<td>Support</td>
<td>None</td>
</tr>
<tr>
<td>Support</td>
<td>No Support</td>
<td>No Support</td>
<td>None</td>
</tr>
<tr>
<td>No Support</td>
<td>Support</td>
<td>Support</td>
<td>Limited</td>
</tr>
<tr>
<td>No Support</td>
<td>Support</td>
<td>No Support</td>
<td>Limited</td>
</tr>
<tr>
<td>No Support</td>
<td>No Support</td>
<td>Support</td>
<td>None</td>
</tr>
<tr>
<td>No Support</td>
<td>No Support</td>
<td>No Support</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 1: Communication possibilities

From the table we can see that when a NAT does not support the
extension no communication can occur. This is for the most part the current situation i.e. SCTP packets sent externally from behind a NAT are discarded by the NAT. In some cases, where the NAT supports the feature but one of the two external hosts does not support the feature communication may occur but in a limited way. For example only one host may be able to have a connection when a collision case occurs.

2. Conventions

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

3. Terminology

This document uses the following terms, which are depicted in Figure 1.

Private-Address (Priv-Addr): The private address that is known to the internal host.

Internal-Port (Int-Port): The port number that is in use by the host holding the Private-Address.

Internal-VTag (Int-VTag): The Verification Tag that the internal host has chosen for its communication. The VTag is a unique 32-bit tag that must accompany any incoming SCTP packet for this association to the Private-Address.

External-Address (Ext-Addr): The address that an internal host is attempting to contact.

External-Port (Ext-Port): The port number of the peer process at the External-Address.

External-VTag (Ext-VTag): The Verification Tag that the host holding the External-Address has chosen for its communication. The VTag is a unique 32-bit tag that must accompany any incoming SCTP packet for this association to the External-Address.

Public-Address (Pub-Addr): The public address assigned to the NAT box which it uses as a source address when sending packets towards the External-Address.
4. Problem Space Overview

When an SCTP endpoint is behind a NAT which supports [I-D.ietf-behave-sctpnat] a number of problems may arise as it tries to communicate with its peer:

- More than one server behind a NAT may pick the same VTag and source port when talking to the same peer server. This creates a situation where the NAT will not be able to tell the two associations apart. This situation is discussed in Section 6.

- When an SCTP endpoint is a server and talking with multiple peers and the peers are behind the same NAT, to the server the two endpoints cannot be distinguished. This case is discussed in Section 7.

- A NAT could at one point during a conversation restart causing all of its state to be lost. This problem and its solution is discussed in Section 8.

- An SCTP endpoint may be behind two NAT’s giving it redundancy. The method to set up this scenario is discussed in Section 9.

Each of these solutions requires additional chunks and parameters, defined in this document, and possibly modified handling procedures from those specified in [RFC4960].

5. Association Setup Considerations

Every association MUST initially be set up single-homed. There MUST NOT be any IPv4 Address parameter, IPv6 Address parameter, or Supported Address Types parameter in the INIT-chunk. The INIT-ACK chunk MUST NOT contain any IPv4 Address parameter or IPv6 Address.
parameter.

If the association should finally be multi-homed, the procedure in Section 9 MUST be used.

The INIT and INIT-ACK chunk SHOULD contain the Disable Restart parameter defined in Section 7.

6. Handling of Internal Port Number and Verification Tag Collisions

Consider the case where two hosts in the Private-Address space want to set up an SCTP association with the same server running on the same host in the Internet. This means that the External-Port and the External-Address are the same. If they both choose the same Internal-Port and Internal-VTag, the NAT box cannot distinguish incoming packets anymore. But this is very unlikely. The Internal-VTags are chosen at random and if the Internal-Ports are also chosen from the ephemeral port range at random this gives a 46-bit random number which has to match. In the TCP like NAPT case the NAT box can control the 16-bit Natted Port.

The same can happen when the INIT-ACK is processed by the NAT.

However, in this unlikely event the NAT box MUST respond to the INIT chunk by sending an ABORT chunk with the M-bit set. 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. The source address of the packet containing the ABORT chunk MUST be the destination address of the SCTP packet containing the INIT chunk.

The sender of the packet containing the INIT chunk, upon reception of an ABORT with M-bit set SHOULD reinitiate the association setup procedure after choosing a new initiate tag. These procedures SHOULD be followed only if the appropriate error cause code for colliding NAT table state is included AND the association is in the COOKIE-WAIT state (i.e. it is awaiting a INIT-ACK). If the endpoint is in any other state an SCTP endpoint SHOULD NOT respond.

The ABORT chunk defined in [RFC4960] is therefore extended by using the following format:
The following error cause with cause code 0x00B0 (V-tag and Port Number Collision) MUST be included in the ABORT chunk:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Cause Code = 0x00B0 | Cause Length = Variable |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

V-tag and Port Number Collision error cause

FIXME: What to do when this is collision happens when processing an ASCONF chunk?

7. Handling of Internal Port Number Collisions

When two SCTP hosts are behind a NAT and using the recommendations in [I-D.ietf-behave-sctpnat] it is possible that two SCTP hosts in the Private-Address space will want to set up an SCTP association with the same server running on the same host in the Internet. For the NAT appropriate tracking may be performed by assuring that the VTags are unique between the two hosts as defined in [I-D.ietf-behave-sctpnat]. But for the external SCTP server on the Internet this means that the External-Port and the External-Address are the same. If they both have chosen the same Internal-Port the server cannot distinguish 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 which is defined as follows:
Disable Restart parameter

When the server receives this parameter it MUST do the following:

- Include in the INIT-ACK a Disable Restart parameter to inform the client that it will support the feature.
- 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) differentiated only by the VTags.

The NAT, when processing the INIT-ACK, should note in its internal table that the association supports the Disable Restart extension. This note is used when establishing future associations (i.e. when processing an INIT from an internal host) to decide if the connection should be allowed. The NAT MUST do the following when processing an INIT:

- If the INIT is destined to an external address and port for which the NAT has no outbound connection, allow the INIT creating an internal mapping table.
- If the INIT matches the external address and port of an already existing connection, validate that the external server supports the Disable Restart feature. If it does allow the INIT to be forwarded.
- If the external server does not support the Disable Restart extension the NAT MUST send an ABORT with the M-bit set.

The following error cause with cause code 0x00B2 (Port Number Collision) MUST be included in the ABORT chunk:
8. Handling of Missing State

If the NAT box receives a packet from the internal network for which the lookup procedure does not find an entry in the NAT table, a packet containing an ERROR chunk is sent back with the M-bit set. The source address of the packet containing the ERROR chunk MUST be the destination address of the incoming SCTP packet. The verification tag is reflected and the T-bit is set. Please note that such an packet containing an ERROR chunk SHOULD NOT be sent if the received packet contains an ABORT, SHUTDOWN-COMPLETE or INIT-ACK chunk.

The ERROR chunk defined in [RFC4960] is therefore extended by using the following format:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = 9    | Reserved  |M|T|           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                   zero or more Error Causes                   /
\                                                               \\n+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Extended ERROR chunk
```

The following error cause with cause code 0x00B1 (Missing State) SHOULD be included in the ERROR chunk:
Missing State error cause

Upon reception by an SCTP end-point with this ERROR chunk the receiver SHOULD take the following actions:

- Validate the verification tag is reflected by looking at the VTag that would have been included in the outgoing packet.
- Validate that the peer of the SCTP association supports the dynamic address extension, if it does not discard the incoming ERROR chunk.
- Generate a new ASCONF chunk containing the V-tags parameter as defined in Figure 2 and the Disable Restart parameter if the association is using the disabled restart feature. By processing this packet the NAT can recover the appropriate state. The procedures for generating an ASCONF chunk can be found in [RFC5061].

If the NAT box receives a packet for which it has no NAT table entry and the packet contains an ASCONF chunk with the V-tags parameter, the NAT box MUST update its NAT table according to the verification tags in the V-tags parameter and the optional Disable Restart parameter.
The peer SCTP endpoint receiving such an ASCONF chunk SHOULD either add the address and respond with an acknowledgment, if the address is new to the association (following all procedures defined in [RFC5061]). Or, if the address is already part of the association, the SCTP endpoint MUST NOT respond with an error, but instead should respond with an ASCONF-ACK chunk acknowledging the address but take no action (since the address is already in the association).

9. Multi Point Traversal Considerations

If a multi-homed SCTP end-point behind a NAT connects to a peer, it SHOULD first set up the association single-homed with only one address causing the first NAT to populate its state. Then it SHOULD add each IP address using ASCONF chunks sent via their respective NATs. The address to add is the wildcard address and the lookup address SHOULD also contain the V-tags parameter and optionally the Disable Restart parameter as illustrated above.

10. 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.

10.1. Get or Set the NAT Friendliness (SCTP_NAT_FRIENDLY)

This socket option can be used to set the NAT friendliness for future associations and and retrieve the value for future and current ones.

```c
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 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.

11. IANA Considerations

[NOTE to RFC-Editor:

"RFCXXXX" is to be replaced by the RFC number you assign this document.

]

[NOTE to RFC-Editor:

The suggested values for the chunk type and the chunk parameter types are tentative and to be confirmed by IANA.

]

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

11.1. New Chunk Flags for Two Chunk Types

As defined in [RFC6096] two chunk flags have to be assigned by IANA for the ERROR chunk. The suggested 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:

<table>
<thead>
<tr>
<th>Chunk Flag Value</th>
<th>Chunk Flag Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>T bit</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>0x02</td>
<td>M Bit</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>0x04</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>0x08</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>0x10</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>0x20</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>0x40</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>0x80</td>
<td>Unassigned</td>
<td></td>
</tr>
</tbody>
</table>

As defined in [RFC6096] one chunk flag has to be assigned by IANA for the ABORT chunk. The suggested 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

11.2. Three New Error Causes

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

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

Chunk Parameter Types

<table>
<thead>
<tr>
<th>Value</th>
<th>Cause Code</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>176</td>
<td>V-tag and Port Number Collision</td>
<td>[RFCXXXXX]</td>
</tr>
<tr>
<td>177</td>
<td>Missing State</td>
<td>[RFCXXXXX]</td>
</tr>
<tr>
<td>178</td>
<td>Port Number Collision</td>
<td>[RFCXXXXX]</td>
</tr>
</tbody>
</table>

11.3. Two New Chunk Parameter Types

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

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

Chunk Parameter Types

<table>
<thead>
<tr>
<th>ID Value</th>
<th>Chunk Parameter Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>49159</td>
<td>Disable Restart (0xC007)</td>
<td>[RFCXXXXX]</td>
</tr>
<tr>
<td>49160</td>
<td>V-tags (0xC008)</td>
<td>[RFCXXXXX]</td>
</tr>
</tbody>
</table>
12. Security Considerations

The document does not add any additional security considerations to the ones given in [RFC4960], [RFC4895], and [RFC5061].

13. Acknowledgments

The authors wish to thank Jason But, Bryan Ford, David Hayes, Alfred Hines, Henning Peters, Timo Voelker, Dan Wing, and Qiaobing Xie for their invaluable comments.

14. References

14.1. Normative References


14.2. Informative References


[I-D.ietf-behave-sctpnat]


Authors’ Addresses

Randall R. Stewart
Adara Networks
Chapin, SC  29036
US

Email: randall@lakerest.net

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

Email: tuexen@fh-muenster.de

Irene Ruengeler
Muenster University of Applied Sciences
Stegerwaldstrasse 39
48565 Steinfurt
DE

Email: i.ruengeler@fh-muenster.de
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/1id-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.
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?........................................7
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:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Octal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-63</td>
<td>0-77</td>
<td>Network Wide Standard Function</td>
</tr>
<tr>
<td>64-127</td>
<td>100-177</td>
<td>Hosts Specific Functions</td>
</tr>
<tr>
<td>128-223</td>
<td>200-337</td>
<td>Reserved for Future Use</td>
</tr>
<tr>
<td>224-255</td>
<td>340-377</td>
<td>Any Experimental Function</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Binary</th>
<th>Hex</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1023</td>
<td>0x0000-0x03FF</td>
<td>System (also Well-Known)</td>
</tr>
<tr>
<td>1024-49151</td>
<td>0x0400-0xBFFF</td>
<td>User (also Registered)</td>
</tr>
<tr>
<td>49152-65535</td>
<td>0xC000-0xFFFF</td>
<td>Dynamic (also Private)</td>
</tr>
</tbody>
</table>

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:

- Demultiplexing transport endpoint associations within an end host
- 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 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:

- Is this really a new service, or can an existing service suffice?

- Is this an experimental service [RFC3692]? If so, consider using the current experimental ports [RFC2780].

- 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.

- Can this service use a Dynamic port number that is coordinated out-of-band, e.g.:
  - By explicit configuration of both endpoints.
  - By internal mechanisms within the same host (e.g., a configuration file, indicated within a URI, or using interprocess communication).
  - Using information exchanged on a related service: FTP, SIP, etc. [RFC959] [RFC3261].
  - 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:

- 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.

- 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 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.

- 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.

- 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].
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:

- Use of a discovery service, either a shared service (mDNS), or a discovery service for a given system [RFC6762] [RFC6763].

- 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.
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 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


10.2. Informative References


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
Generic Aggregation of Resource ReSerVation Protocol (RSVP) for IPv4 And IPv6 Reservations over PCN domains
draft-ietf-tsvwg-rsvp-pcn-11

Abstract

This document specifies extensions to Generic Aggregated RSVP RFC 4860 for support of the PCN Controlled Load (CL) and Single Marking (SM) edge behaviors over a Diffserv cloud using Pre-Congestion Notification.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 6, 2015.
3.9. Initiation of New Aggregate Resv Message By Deaggregating Router 17
3.10. Handling of Aggregate Resv Message by Interior Routers . . . 18
3.11. Handling of E2E Resv Message by Aggregating Router . . . . 18
3.12. Handling of Aggregated Resv Message by Aggregating Router . . 18
3.13. Removal of E2E Reservation . . . . . . . . . . . . . . . . . . 19
3.14. Removal of Aggregate Reservation . . . . . . . . . . . . . . . 19
3.15. Handling of Data On Reserved E2E Flow by Aggregating Router . 19
3.16. Procedures for Multicast Sessions . . . . . . . . . . . . . . . 19
3.17. Misconfiguration of PCN node . . . . . . . . . . . . . . . . . 19
3.18. PCN based Flow Termination . . . . . . . . . . . . . . . . . . 19
4. Protocol Elements . . . . . . . . . . . . . . . . . . . . . . . . . 20
4.1 PCN object . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
5. Security Considerations . . . . . . . . . . . . . . . . . . . . . . . 23
6. IANA Considerations . . . . . . . . . . . . . . . . . . . . . . . . 24
7. Acknowledgments . . . . . . . . . . . . . . . . . . . . . . . . . 24
8. Normative References . . . . . . . . . . . . . . . . . . . . . . . . 24
9. Informative References . . . . . . . . . . . . . . . . . . . . . . . 25
10. Appendix A: Example Signaling Flow . . . . . . . . . . . . . . . 26
11. Authors’ Address . . . . . . . . . . . . . . . . . . . . . . . . . 29
1. Introduction

1.1 Objective

Pre-Congestion Notification (PCN) can support the quality of service (QoS) of inelastic flows within a Diffserv domain in a simple, scalable, and robust fashion. Two mechanisms are used: admission control and flow termination. Admission control is used to decide whether to admit or block a new flow request, while flow termination is used in abnormal circumstances to decide whether to terminate some of the existing flows. To support these two features, the overall rate of PCN-traffic is metered on every link in the domain, and PCN-packets are appropriately marked when certain configured rates are exceeded. These configured rates are below the rate of the link, thus providing notification to boundary nodes about overloads before any congestion occurs (hence "pre-congestion" notification). The PCN-egress-nodes measure the rates of differently marked PCN traffic in periodic intervals and report these rates to the Decision Points for admission control and flow termination; the Decision Points use these rates to make decisions. The Decision Points may be collocated with the PCN-ingress-nodes, or their function may be implemented in a another node. For more details see [RFC5559], [RFC6661], and [RFC6662].

The main objective of this document is to specify the signaling protocol that can be used within a Pre-Congestion Notification (PCN) domain to carry reports from a PCN-ingress-node to a PCN Decision point, considering that the PCN Decision Point and PCN-egress-node are collocated.

If the PCN Decision Point is not collocated with the PCN-egress-node then additional signaling procedures are required that are out of the scope of this document. Moreover, as mentioned above this architecture conforms with PBAC (Policy-Based Admission Control), when the Decision Point is located in a another node then the PCN-ingress-node [RFC2753].

Several signaling protocols can be used to carry information between PCN-boundary-nodes (PCN-ingress-node and PCN-egress-node). However, since (1) both PCN-egress-node and PCN-ingress-nodes are located on the data path and (2) the admission control procedure needs to be done at PCN-egress-node, a signaling protocol that follows the same path as the data path, like RSVP (Resource Reservation Protocol), is more suited for this purpose. In particular, this document specifies extensions to Generic Aggregated RSVP [RFC4860] for support of the PCN Controlled Load (CL) and Single Marking (SM) edge behaviors over a Diffserv cloud using Pre-Congestion Notification.

This draft is intended to be published as Experimental in order to:

1. validate industry interest by allowing implementation and deployment
2. gather operational experience, in particular around dynamic interactions of RSVP signaling and PCN notification and
Support for the techniques specified in this document involves RSVP functionality in boundary nodes of a PCN domain whose interior nodes forward RSVP traffic without performing RSVP functionality.

1.2 Overview and Motivation

Two main Quality of Service (QoS) architectures have been specified by the IETF. These are the Integrated Services (Intserv) [RFC1633] architecture and the Differentiated Services (DiffServ) architecture ([RFC2475]).

Intserv provides methods for the delivery of end-to-end Quality of Service (QoS) to applications over heterogeneous networks. One of the QoS signaling protocols used by the Intserv architecture is the Resource reServation Protocol (RSVP) [RFC2205], which can be used by applications to request per-flow resources from the network. These RSVP requests can be admitted or rejected by the network. Applications can express their quantifiable resource requirements using Intserv parameters as defined in [RFC2211] and [RFC2212]. The Controlled Load (CL) service [RFC2211] is a quality of service (QoS) closely approximating the QoS that the same flow would receive from a lightly loaded network element. The CL service is useful for inelastic flows such as those used for real-time media.

The DiffServ architecture can support the differentiated treatment of packets in very large scale environments. While Intserv and RSVP classify packets per-flow, Diffserv networks classify packets into one of a small number of aggregated flows or "classes", based on the Diffserv codepoint (DSCP) in the packet IP header. At each Diffserv router, packets are subjected to a "per-hop behavior" (PHB), which is invoked by the DSCP. The primary benefit of Diffserv is its scalability, since the need for per-flow state and per-flow processing, is eliminated.

However, DiffServ does not include any mechanism for communication between applications and the network. Several solutions have been specified to solve this issue. One of these solutions is Intserv over Diffserv [RFC2998] including resource-based admission control (RBAC), PBAC, assistance in traffic identification/classification, and traffic conditioning. Intserv over Diffserv can operate over a statically provisioned or a RSVP aware Diffserv region. When it is RSVP aware, several mechanisms may be used to support dynamic provisioning and topology-aware admission control, including aggregate RSVP reservations, per-flow RSVP, or a bandwidth broker. [RFC3175] specifies aggregation of Resource ReSerVation Protocol (RSVP) end-to-end reservations over aggregate RSVP reservations. In [RFC3175] the RSVP generic aggregated reservation is characterized by a RSVP SESSION object using the 3-tuple \langle source IP address, destination IP address, Diffserv Code Point \rangle.

Several scenarios require the use of multiple generic aggregate reservations that are established for a given PHB from a given source.
IP address to a given destination IP address, see [SIG-NESTED], [RFC4860]. For example, multiple generic aggregate reservations can be applied in the situation that multiple E2E reservations using different preemption priorities need to be aggregated through a PCN-domain using the same PHB. By using multiple aggregate reservations for the same PHB, it allows enforcement of the different preemption priorities within the aggregation region. This allows more efficient management of the Diffserv resources, and in periods of resource shortage, this allows sustainment of a larger number of E2E reservations with higher preemption priorities. In particular, [SIG-NESTED] discusses in detail how end-to-end RSVP reservations can be established in a nested VPN environment through RSVP aggregation.

[RFC4860] provides generic aggregate reservations by extending [RFC3175] to support multiple aggregate reservations for the same source IP address, destination IP address, and PHB (or set of PHBs). In particular, multiple such generic aggregate reservations can be established for a given PHB from a given source IP address to a given destination IP address. This is achieved by adding the concept of a Virtual Destination Port and of an Extended Virtual Destination Port in the RSVP SESSION object. In addition to this, the RSVP SESSION object for generic aggregate reservations uses the PHB Identification Code (PHB-ID) defined in [RFC3140], instead of using the Diffserv Code Point (DSCP) used in [RFC3175]. The PHB-ID is used to identify the PHB, or set of PHBs, from which the Diffserv resources are to be reserved.

The RSVP like signaling protocol required to carry (1) requests from a PCN-egress-node to a PCN-ingress-node and (2) reports from a PCN-ingress-node to a PCN-egress-node needs to follow the PCN signaling requirements defined in [RFC6663]. In addition to that the signaling protocol functionality supported by the PCN-ingress-nodes and PCN-egress-nodes needs to maintain logical aggregate constructs (i.e. ingress-egress-aggregate state) and be able to map E2E reservations to these aggregate constructs. Moreover, no actual reservation state is needed to be maintained inside the PCN domain, i.e., the PCN-interior-nodes are not maintaining any reservation state.

This can be accomplished by two possible approaches:

Approach (1):

  o) adapting the RFC 4860 aggregation procedures to fit the PCN requirements with as little change as possible over the RFC 4860 functionality

  o) hence performing aggregate RSVP signaling (even if it is to be ignored by PCN interior nodes)

  o) using this aggregate RSVP signaling procedures to carry PCN information between the PCN-boundary-nodes (PCN-ingress-node and PCN-egress-node).
Approach (2):

o) adapting the RFC 4860 aggregation procedures to fit the PCN requirements with more significant changes over RFC4860 (i.e. the aspect of the procedures that have to do with maintaining aggregate states and to do with mapping the E2E reservations to aggregate constructs are kept, but the procedures that have to do with the aggregate RSVP signaling and aggregate reservation establishment/maintenance are dropped).

o) hence not performing aggregate RSVP signaling

o) piggy-backing of the PCN information inside the E2E RSVP signaling.

Both approaches are probably viable, however, since the RFC 4860 operations have been thoroughly studied and implemented, it can be considered that the RFC 4860 solution can better deal with the more challenging situations (rerouting in the PCN domain, failure of an PCN-ingress-node, failure of an PCN-egress-node, rerouting towards a different edge, etc.). This is the reason for choosing Approach (1) for the specification of the signaling protocol used to carry PCN information between the PCN-boundary-nodes (PCN-ingress-node and PCN-egress-node).

In particular, this document specifies extensions to Generic Aggregated RSVP [RFC4860] for support of the PCN Controlled Load (CL) and Single Marking (SM) edge behaviors over a Diffserv cloud using Pre-Congestion Notification.

This document follows the PCN signaling requirements defined in [RFC6663] and specifies extensions to Generic Aggregated RSVP [RFC4860] for support of PCN edge behaviors as specified in [RFC6661] and [RFC6662]. Moreover, this document specifies how RSVP aggregation can be used to setup and maintain: (1) Ingress Egress Aggregate (IEA) states at Ingress and Egress nodes and (2) generic aggregation of RSVP end-to-end RSVP reservations over PCN (Congestion and Pre-Congestion Notification) domains.

To comply with this specification, PCN-nodes MUST be able to support the functionality specified in [RFC5670], [RFC5559], [RFC6660], [RFC6661], [RFC6662]. Furthermore, the PCN-boundary-nodes MUST support the RSVP generic aggregated reservation procedures specified in [RFC4860] which are augmented with procedures specified in this document.

1.3. Terminology

This document uses terms defined in [RFC4860], [RFC3175], [RFC5559], [RFC5670], [RFC6661], [RFC6662].

For readability, a number of definitions from [RFC3175] as well as definitions for terms used in [RFC5559], [RFC6661], and [RFC6662] are provided here, where some of them are augmented with new meanings:
Aggregator

This is the process in (or associated with) the router at the ingress edge of the aggregation region (with respect to the end-to-end RSVP reservation) and behaving in accordance with [RFC4860]. In this document, it is also the PCN-ingress-node. It is important to notice that in the context of this document the Aggregator must be able to determine the Deaggregator using the procedures specified in Section 4 of [RFC4860] and in Section 1.4.2 of [RFC3175].

Congestion level estimate (CLE):

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:

A single instance of an application-to-application flow of packets which is identified by source address, destination address, protocol id, and source port, destination port (where applicable).

PCN-domain:

A PCN-capable domain; a contiguous set of PCN-enabled nodes that perform Diffserv scheduling [RFC2474]; the complete set of PCN-nodes that in principle can, through PCN-marking packets, influence decisions about flow admission and termination within the domain; includes the PCN-egress-nodes, which measure these PCN-marks, and the PCN-ingress-nodes.

PCN-boundary-node: a PCN-node that connects one PCN-domain to a node either in another PCN-domain or in a non-PCN-domain.
PCN-interior-node: a node in a PCN-domain that is not a PCN-boundary-node.

PCN-node: a PCN-boundary-node or a PCN-interior-node.

PCN-egress-node: a PCN-boundary-node in its role in handling traffic as it leaves a PCN-domain. In this document the PCN-egress-node operates also as a Decision Point and Deaggregator.

PCN-ingress-node: a PCN-boundary-node in its role in handling traffic as it enters a PCN-domain. In this document the PCN-ingress-node operates also as an 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:

- the IPv4 DestAddress, IPv6 DestAddress should be set to the IPv4 or IPv6 destination addresses, respectively, of the Deaggregator (PCN-egress-node)
- PHB-ID (Per Hop Behavior Identification Code) should be set equal to PCN-compatible Diffserv codepoint(s).
- Extended vDstPort should be set to the IPv4 or IPv6 destination addresses, of the Aggregator (PCN-ingress-node)

VDstPort (Virtual Destination Port)

A 16-bit identifier used in the SESSION that remains constant over the life of the generic aggregate reservation.

1.4. Organization of This Document

This document is organized as follows. Section 2 gives an overview of RSVP extensions and operations. The elements of the used procedures are specified in Section 3. Section 4 describes the protocol elements. The security considerations are given in section 5 and the IANA considerations are provided in Section 6.

2. Overview of RSVP extensions and Operations

2.1 Overview of RSVP Aggregation Procedures in PCN domains

The PCN-boundary-nodes, see Figure 1, can support RSVP SESSIONS for generic aggregated reservations [RFC4860], which are depending on ingress-egress-aggregates. In particular, one RSVP generic aggregated reservation matches to only one ingress-egress-aggregate.

However, one ingress-egress-aggregate matches to either one, or more than one, RSVP generic aggregated reservations. In addition, to comply with this specification, the PCN-boundary nodes need to distinguish and process (1) RSVP SESSIONS for generic aggregated sessions and their messages according to [RFC4860], (2) E2E RSVP sessions and messages according to [RFC2205].

This document locates all RSVP processing for a PCN domain at PCN-Boundary nodes. PCN-interior-nodes do not perform any RSVP functionality or maintain RSVP-related state information. Rather, PCN-interior nodes forward all RSVP messages (for both generic aggregated reservations[RFC4860] and end to end reservations [RFC2205]) as if they were ordinary network traffic.
Moreover, each Aggregator and Deaggregator (i.e., PCN-boundary-nodes) need to support policies to initiate and maintain for each pair of PCN-boundary-nodes of the same PCN-domain one ingress-egress-aggregate.

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:

1. Unlike [RFC4860], the generic aggregate RSVP reservations need not be admitted in the PCN core.
2. Unlike [RFC4860], the RSVP aggregated traffic does not need to be tunneled between Aggregator and Deaggregator, see Section 2.3.
3. Unlike [RFC4860], the Deaggregator need not perform admission control of E2E reservations over the aggregate RSVP reservations.
4. 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:

- 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 an 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:
  - 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.
  - PHB-ID (Per Hop Behavior Identification Code) SHOULD be set equal to PCN-compatible Diffserv codepoint(s).
  - 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 used as the ones described in Section 4 of [RFC4860].

3. Elements of Procedure

This section describes the procedures used to implement the aggregated RSVP procedure over PCN. It is considered that the procedures for aggregation of E2E reservations over generic aggregate RSVP reservations are same as the procedures specified in Section 4 of [RFC4860] except where a departure from these procedures is explicitly described in the present section. Please refer to [RFC4860] for all the below error cases:

- Incomplete message
- 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:

- The Deaggregator MUST NOT perform the RSVP-TTL vs IP TTL-check and MUST NOT update the ADspec Break bit. This is because the whole PCN-domain is effectively handled by E2E RSVP as a virtual link on which integrated service is indeed supported (and admission control performed) so that the Break bit MUST NOT be set, see also [draft-lefaucheur-rsvp-ecn-01].

The Deaggregator forwards the E2E Path message towards the receiver.

3.4. Initiation of new Aggregate Path Message by Aggregating Router

To comply with this specification, for the initiation of the new RSVP generic aggregated Path message by the Aggregator (also PCN-ingress-node in this document), the same methods MUST be used as the ones described in [RFC4860].

3.5. Handling Of Aggregate Path Message By Interior Routers

The Aggregate Path messages traverse zero or more PCN-interior-nodes. The PCN-interior-nodes receive the Aggregated Path message on an interior interface and forward it on another interior interface. It is considered that, by configuration, the PCN-interior-nodes ignore the Aggregated Path signaling messages. Therefore, the Aggregated Path messages are simply forwarded as normal IP datagrams.

3.6. Handling Of Aggregate Path Message By Deaggregating Router

When receiving the Aggregated Path message, the Deaggregator (also PCN-egress-node and Decision Point in this document) performs the regular [RFC4860] procedures, augmented with the following rules:

- 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:

- 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.

- 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:

- 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.
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.

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.

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:

- RSVP-AGGREGATE-IPv4-PCN-request
- 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:

- 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].


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:

- The DSCP value included in the SESSION object, SHOULD be set equal to a PCN-compatible Diffserv codepoint.
- Extended vDstPort SHOULD be set to the IPv4 or IPv6 destination addresses, of the Aggregator (also PCN-ingress-node in this document), see [RFC4860].
- 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:
  - RSVP-AGGREGATE-IPv4-PCN-request
  - RSVP-AGGREGATE-IPv6-PCN-request.
- When the Aggregator receives an Aggregate Resv message which carries one of the following PCN objects:
  - RSVP-AGGREGATE-IPv4-PCN-request
  - 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:
  - RSVP-AGGREGATE-IPv4-PCN-response
  - RSVP-AGGREGATE-IPv6-PCN-response.

4.1 PCN objects

This section describes four types of PCN objects that can be carried by the (refresh) Aggregate Path or the (refresh) Aggregate Resv messages specified in [RFC4860].
These objects are:
  o RSVP-AGGREGATE-IPv4-PCN-request,
  o RSVP-AGGREGATE-IPv6-PCN-request,
  o RSVP-AGGREGATE-IPv4-PCN-response,
  o RSVP-AGGREGATE-IPv6-PCN-response.

o) RSVP-AGGREGATE-IPv4-PCN-request: PCN request object, when IPv4 addresses are used:
Class = 248 (PCN)
C-Type = 1 (RSVP-AGGREGATE-IPv4-PCN-request)
+-------------+-------------+-------------+-------------+
|     IPv4 PCN-ingress-node Address (4 bytes)           |
|-------------|-------------|-------------+-------------+
|     IPv4 PCN-egress-node Address (4 bytes)            |
|-------------|-------------|-------------|-------------+
|     IPv4 Decision Point Address (4 bytes)             |
|-------------|-------------|-------------+-------------|
|                    R|     Reserved                                        |
+-------------+-------------|-------------+-------------+

o) RSVP-AGGREGATE-IPv6-PCN-request: PCN object, when IPv6 addresses are used:
Class = 248 (PCN)
C-Type = 2 (RSVP-AGGREGATE-IPv6-PCN-request)
+-------------+-------------+-------------+-------------+
|     IPv6 PCN-ingress-node Address (16 bytes)          |
|-------------|-------------|-------------+-------------+
|     IPv6 PCN-egress-node Address (16 bytes)           |
|-------------|-------------|-------------|-------------|
|     Decision Point Address (16 bytes)                 |
|-------------|-------------+-------------|-------------|
|                    R|     Reserved                                        |
+-------------+-------------+-------------+-------------+
o) RSVP-AGGREGATE-IPv4-PCN-response: PCN object, IPv4 addresses are used:
Class = 248 (PCN)
C-Type = 3 (RSVP-AGGREGATE-IPv4-PCN-response)

+-------------+-------------+-------------+-------------+
|     IPv4 PCN-ingress-node Address (4 bytes)           |
|     IPv4 PCN-egress-node Address (4 bytes)            |
|     IPv4 Decision Point Address (4 bytes)             |
| PCN-sent-rate                                         |
+-------------+-------------+-------------+-------------+

o) RSVP-AGGREGATE-IPv6-PCN-response: PCN object, IPv6 addresses are used:
Class = 248 (PCN)
C-Type = 4 (RSVP-AGGREGATE-IPv6-PCN-response)

+-------------+-------------+-------------+-------------+
|     IPv6 PCN-ingress-node Address (16 bytes)          |
|     IPv6 PCN-egress-node Address (16 bytes)           |
|     Decision Point Address (16 bytes)                 |
| PCN-sent-rate                                         |
+-------------+-------------+-------------+-------------+

The fields carried by the PCN object are specified in
[RFC6663], [RFC6661] and [RFC6662]:

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);

- 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.

- "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.

- "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.

- PCN-sent-rate: the PCN-sent-rate for the given ingress-egress-aggregate. It is expressed in octets/second; its format is a 32-bit IEEE floating point number; The PCN-sent-rate is specified in [RFC6661] and [RFC6662] and it represents the estimate of the rate at which the PCN-ingress-node (Aggregator) is receiving PCN-traffic that is destined for the given ingress-egress-aggregate.

5. Security Considerations

The security considerations specified in [RFC2205], [RFC4860] and [RFC5559] apply to this document. In addition, [RFC4230] and [RFC6411] provide useful guidance on RSVP security mechanisms.

Security within a PCN domain is fundamentally based on the controlled environment trust assumption stated in Section 6.3.1 of [RFC5559], in particular that all PCN-nodes are PCN-enabled and are trusted to perform accurate PCN-metering and PCN-marking.

In the PCN domain environments addressed by this document, Generic Aggregate Resource ReSerVation Protocol (RSVP) messages specified in [RFC4860] are used for support of the PCN Controlled Load (CL) and Single Marking (SM) edge behaviors over a Diffserv cloud using Pre-Congestion Notification. Hence the security mechanisms discussed in [RFC4860] are applicable. Specifically, the INTEGRITY object [RFC2747][RFC3097] can be used to provide hop-by-hop RSVP message integrity, node authentication and replay protection, thereby protecting against corruption and spoofing of RSVP messages and PCN feedback conveyed by RSVP messages.

For these reasons, this document does not introduce significant additional security considerations beyond those discussed in
IANA has modified the RSVP parameters registry, ‘Class Names, Class Numbers, and Class Types’ subregistry, to add a new Class Number and assign 4 new C-Types under this new Class Number, as described below, see Section 4.1:

<table>
<thead>
<tr>
<th>Class Number</th>
<th>Class Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>248</td>
<td>PCN</td>
<td>this document</td>
</tr>
</tbody>
</table>

Class Types or C-Types:
1 RSVP-AGGREGATE-IPv4-PCN-request this document
2 RSVP-AGGREGATE-IPv6-PCN-request this document
3 RSVP-AGGREGATE-IPv4-PCN-response this document
4 RSVP-AGGREGATE-IPv6-PCN-response this document

When this draft is published as an RFC, IANA should update the reference for the above 5 items to that published RFC (and the RFC Editor should remove this sentence).

7. Acknowledgments

We would like to thank the authors of [draft-lefaucheur-rsvp-ecn-01.txt], since some ideas used in this document are based on the work initiated in [draft-lefaucheur-rsvp-ecn-01.txt]. Moreover, we would like to thank Bob Briscoe, David Black, Ken Carlberg, Tom Taylor, Philip Eardley, Michael Menth, Toby Moncaster, James Polk, Scott Bradner, Lixia Zhang and Robert Sparks for the provided comments. In particular, we would like to thank Francois Le Faucheur for contributing in addition to comments also to a significant amount of text.

8. Normative References


9. Informative References


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
(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 (GAp cn) 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 (GAp cn). 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 (GAp cn). Otherwise, assuming that the generic aggregate reservation for the PCN (GAp cn) had been established with sufficient bandwidth to support the E2E Resv, the Deaggregator adjusts its counter, tracking the unused bandwidth on the generic aggregate reservation. Then it forwards the E2E Resv to the Aggregator including a SESSION-OF-INTEREST
object conveying the selected mapping onto GApcn (and hence onto
the PCN PHB-ID).

(9) The Aggregator records the mapping of the E2E Resv onto GApcn
(and onto the PCN PHB-ID). The Aggregator removes the SOI object
and forwards the E2E Resv towards the sender.

11. Authors’ Address

Georgios Karagiannis
Huawei Technologies
Hansaallee 205,
40549 Dusseldorf,
Germany
Email: Georgios.Karagiannis@huawei.com

Anurag Bhargava
Cisco Systems, Inc.
7100-9 Kit Creek Road
PO Box 14987
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709-4987
USA
Email: anuragb@cisco.com
Abstract

This document describes a simple method of encapsulating SCTP Packets into UDP packets and its limitations. This allows the usage of SCTP in networks with legacy NAT not supporting SCTP. It can also be used to implement SCTP on hosts without directly accessing the IP-layer, for example implementing it as part of the application without requiring special privileges.

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 12, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of
Table of Contents

1. Introduction ............................................. 3
2. Conventions ........................................... 3
3. Use Cases ............................................... 3
   3.1. Portable SCTP Implementations ..................... 3
   3.2. Legacy NAT Traversal ............................... 4
4. SCTP over UDP .......................................... 4
   4.1. Architectural Considerations ...................... 4
   4.2. Packet Format ...................................... 4
   4.3. Encapsulation Procedure ............................ 6
   4.4. Decapsulation Procedure ............................ 6
   4.5. ICMP Considerations ................................. 6
   4.6. Path MTU Considerations ............................ 7
   4.7. Handling of Embedded IP-addresses ................. 7
   4.8. ECN Considerations ................................ 7
5. Socket API Considerations ............................. 7
   5.1. Get or Set the Remote UDP Encapsulation Port Number
        (SCTP_REMOTE_UDP_ENCAPS_PORT) .................... 7
6. IANA Considerations .................................... 8
7. Security Considerations ............................... 8
8. Acknowledgments ........................................ 8
9. References ............................................. 8
   9.1. Normative References ............................... 8
   9.2. Informative References ............................. 9
Authors’ Addresses ..................................... 10
1. Introduction

This document describes a simple method of encapsulating SCTP packets into UDP packets. SCTP as defined in [RFC4960] runs directly over IPv4 or IPv6. There are two main reasons for encapsulating SCTP packets:

- Allow SCTP traffic to pass legacy NATs, which do not provide native SCTP support as specified in [I-D.ietf-behave-sctpnat] and [I-D.ietf-tsvwg-natsupp].
- Allow SCTP to be implemented on hosts which do not provide direct access to the IP-layer. In particular, applications can use their own SCTP implementation if the operating system does not provide one.

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. Use Cases

This section discusses two important use cases for encapsulating SCTP into UDP.

3.1. Portable SCTP Implementations

Some operating systems support SCTP natively. For other operating systems implementations are available, but require special privileges to install and/or use them. In some cases no kernel implementation might be available at all. When proving an SCTP implementation as part of a user process, most operating systems require special privileges to access the IP layer directly.

Using UDP encapsulation makes it possible to provide an SCTP implementation as part of a user process which does not require any special privileges.

A crucial point for implementing SCTP in user-land is controlling the source address of outgoing packets. This is not an issue when using all available addresses. However, this is not the case when also using the address management required for NAT traversal described in Section 4.7.
3.2. Legacy NAT Traversal

Using UDP encapsulation allows SCTP communication when traversing legacy NATs (i.e., those NATs not supporting SCTP as described in [I-D.ietf-behave-sctpnat] and [I-D.ietf-tsvwg-natsupp]). It is important to realize that for single homed associations it is only necessary that no IP addresses are listed in the INIT and INIT-ACK chunks. To use multiple addresses, the dynamic address reconfiguration extension described in [RFC5061] MUST be used with wildcard addresses in combination with [RFC4895].

For multi-homed SCTP association the address management as described in Section 4.7 MUST be performed.

4. SCTP over UDP

4.1. Architectural Considerations

An SCTP implementation supporting UDP encapsulation MUST store a remote UDP encapsulation port number per destination address for each SCTP association.

Each SCTP stack uses a single local UDP encapsulation port number as the destination port for all its incoming SCTP packets. The IANA-assigned value of 9989 MAY be used as this port number. If there is only a single SCTP implementation on a host (for example, a kernel implementation being part of the operating system), using a single UDP encapsulation port number per host can be advantageous (e.g., this reduces the number of mappings in firewalls and NATs, among other things). However, this is not possible if the SCTP stack is implemented as part of an application.

4.2. Packet Format

To encapsulate an SCTP packet, a UDP header as defined in [RFC0768] is inserted between the IP header as defined in [RFC0791] and the SCTP common header as defined in [RFC4960].

Figure 1 shows the packet format of an encapsulated SCTP packet when IPv4 is used.
The packet format for an encapsulated SCTP packet when using IPv6 as defined in [RFC2460] is shown in Figure 2. Please note the the number m of IPv6 extension headers can be 0.

The UDP checksum MUST NOT be zero.
4.3. Encapsulation Procedure

When inserting the UDP header, the source port is the local UDP encapsulation port number of the SCTP stack, the destination port is the remote UDP encapsulation port number stored for the destination address the packet is sent to (see Section 4.1).

The length of the UDP packet is the length of the SCTP packet plus the size of the UDP header.

The UDP checksum and the SCTP checksum MUST be computed.

4.4. Decapsulation Procedure

When an encapsulated packet is received, the UDP header is removed. Then a lookup is performed to find the association the received SCTP packet belongs to. The UDP source port is stored as the encapsulation port for the destination address the SCTP packet is received from (see Section 4.1).

Please note that when a non-encapsulated SCTP packet is received, the encapsulation of outgoing packets belonging to the same association and the corresponding destination address is disabled.

4.5. ICMP Considerations

When receiving ICMP or ICMPv6 response packets, there might not be enough bytes in the payload to identify the SCTP association which the SCTP packet triggering the ICMP or ICMPv6 packet belongs to. If a received ICMP or ICMPv6 packet can not be related to a specific SCTP association, it MUST be discarded silently. This means in particular that the SCTP stack MUST NOT rely on receiving ICMP or ICMPv6 messages. There MAY be implementation constraints not allowing to process received ICMP or ICMPv6 messages at all.

If received ICMP or ICMPv6 messages are processed, the following mapping SHOULD apply:

1. ICMP messages with type 'Destination Unreachable' and code 'Port Unreachable' SHOULD be treated as ICMP messages with type 'Protocol Unreachable' and code 'Destination Port unreachable. See [RFC0792] for more details.

2. ICMPv6 messages with type 'Destination Unreachable' and code 'Port unreachable' SHOULD be treated as ICMPv6 messages with type 'Parameter Problem' and code 'Unrecognized Next Header type encountered’. See [RFC4443] for more details.
4.6. Path MTU Considerations

If an SCTP endpoint starts to encapsulate the packets of a path, it MUST decrease the path MTU of that path by the size of the UDP header. If it stops encapsulating them, the path MTU SHOULD be increased by the size of the UDP header.

When performing path MTU discovery as described in [RFC4820] and [RFC4821] it MUST be taken into account that one cannot rely on the feedback provided by ICMP or ICMPv6 due to the limitation laid out in Section 4.5.

4.7. Handling of Embedded IP-addresses

When using UDP encapsulation for legacy NAT traversal, IP addresses that might require translation MUST NOT be put into any SCTP packet.

This means that a multi homed SCTP association is setup initially as a singled homed one and the protocol extension [RFC5061] in combination with [RFC4895] is used to add the other addresses. Only wildcard addresses are put into the SCTP packet.

When addresses are changed during the lifetime of an association [RFC5061] MUST be used with wildcard addresses only.

4.8. ECN Considerations

During encapsulation and decapsulation the ECN bits MUST NOT be changed.

5. Socket API Considerations

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

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.

5.1. Get or Set the Remote UDP Encapsulation Port Number (SCTP_REMOTE_UDP_ENCAPS_PORT)

This socket option can be used to set and retrieve the UDP encapsulation port number. This allows an endpoint to encapsulate initial packets.
struct sctp_udpencaps {
    sctp_assoc_t sue_assoc_id;
    struct sockaddr_storage sue_address;
    uint16_t sue_port;
};

sue_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 for this query. It is an error to use SCTP_{CURRENT|ALL}_ASSOC in sue_assoc_id.

sue_address: This specifies which address is of interest. If a wildcard address is provided it applies only to future paths.

sue_port: The UDP port number in network byte order used as the destination port number for UDP encapsulation. Providing a value of 0 disables UDP encapsulation.

6. IANA Considerations

This document does not require any actions from IANA.

7. Security Considerations

Encapsulating SCTP into UDP does not add any additional security considerations to the ones given in [RFC4960] and [RFC5061].

8. Acknowledgments

The authors wish to thank Irene Ruengeler and Dan Wing for their invaluable comments.

9. References

9.1. Normative References


9.2. Informative References


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
Deprecation of ICMP Source Quench messages

draft-ietf-tsvwg-source-quench-06.txt

Abstract

This document formally deprecates the use of ICMP Source Quench messages by transport protocols, formally updating RFC 792, RFC 1122, and RFC 1812.

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 28, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

1. Introduction .......................... 3
2. ICMP Source Quench messages ................. 3
3. Updating RFC 1122 .......................... 4
4. Updating RFC 1812 .......................... 4
5. Clarification for UDP, SCTP, and DCCP .......... 5
6. General Advice to Transport Protocols .......... 5
8. Security Considerations ..................... 5
9. IANA Considerations ....................... 6
10. Acknowledgements ........................ 6
11. References ................................ 7
   11.1. Normative References .................. 7
   11.2. Informative References ................. 7
Appendix A. Survey of support of ICMP Source Quench in some popular TCP/IP implementations ........................ 8
Appendix B. Changes from previous versions of the draft (to be removed by the RFC Editor before publishing this document as an RFC) ................. 8
   B.1. Changes from draft-ietf-tsvwg-source-quench-05 .......... 8
   B.2. Changes from draft-ietf-tsvwg-source-quench-04 .......... 8
   B.3. Changes from draft-ietf-tsvwg-source-quench-03 .......... 8
   B.5. Changes from draft-ietf-tsvwg-source-quench-01 .......... 9
   B.6. Changes from draft-ietf-tsvwg-source-quench-00 .......... 9
   B.7. Changes from draft-gont-tsvwg-source-quench-01 .......... 9
   B.8. Changes from draft-gont-tsvwg-source-quench-00 .......... 9
Author's Address ................................ 9
1. Introduction

The ICMP specification [RFC0792] defined the ICMP Source Quench message (type 4, code 0), which was meant as a mechanism for congestion control. ICMP Source Quench has been known to be an ineffective (and unfair) antidote for congestion, and generation of ICMP Source Quench messages by routers has been formally deprecated by [RFC1812] since 1995. However, reaction to ICMP Source Quench messages in transport protocols has never been formally deprecated.

This document formally deprecates reaction to ICMP Source Quench messages by transport protocols such as TCP, formally updating [RFC0792], [RFC1122], and [RFC1812]. Additionally, it provides recommendation against the implementation of [RFC1016]. The rationale for these specification updates is:

- Processing of ICMP Source Quench messages by routers has been deprecated for more than 20 years [RFC1812].
- Virtually all popular host implementations have removed support for ICMP Source Quench messages since (at least) 2005 [RFC5927].
- Widespread deployment of ICMP filtering makes it impossible to rely on ICMP Source Quench messages for congestion control.
- The IETF has moved away from ICMP Source Quench messages for congestion control (note e.g. the development of ECN [RFC3168], and the fact that ICMPv6 [RFC4443] does not even specify a Source Quench message).

ICMP Source Quench messages are not normally seen in the deployed Internet and were considered rare at least as far back as 1994. [Floyd1994]

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].

2. ICMP Source Quench messages

The ICMP specification [RFC0792] defined the ICMP Source Quench message (type 4, code 0), which was meant to provide a mechanism for congestion control. The Host Requirements RFC [RFC1122] stated in Section 4.2.3.9 that hosts MUST react to ICMP Source Quench messages by slowing transmission on the connection, and further added that the RECOMMENDED procedure was to put the corresponding connection in the slow-start phase of TCP’s congestion control algorithm [RFC5681].
Internet-Draft        Deprecation of ICMP Source Quench        February 2012

[RFC1812] noted that research suggested that ICMP Source Quench was
an ineffective (and unfair) antidote for congestion, and formally
deprecated the generation of ICMP Source Quench messages by routers,
stating that routers SHOULD NOT send ICMP Source Quench messages in
response to congestion.

[RFC5927] discussed the use of ICMP Source Quench messages for
performing "blind throughput-reduction" attacks, and noted that most
TCP implementations silently ignore ICMP Source Quench messages.

We note that TCP implements its own congestion control mechanisms
[RFC5681] [RFC3168], that do not depend on ICMP Source Quench
messages.

It is interesting to note that ICMPv6 [RFC4443] does not specify a
"Source Quench" message.

3. Updating RFC 1122

This document hereby updates Section 3.2.2.3 of [RFC1122] as follows:

A host MUST NOT send ICMP Source Quench messages.

If a Source Quench message is received, the IP layer MAY silently
discard it.

Section 4.2.3.9 of [RFC1122] is updated as follows:

TCP MUST silently discard any received ICMP Source Quench
messages.

The consensus of the TSV WG was that there are no valid reasons for a
host to generate or react to an ICMP Source Quench message in the
current Internet. The recommendation that a sender "MUST NOT" send
an ICMP Source Quench message is because there is no known valid
reason for a host to generate this message. The only known impact of
a sender ignoring this requirement is that it may necessarily consume
network and endpoint resources. Discarding ICMP Source Quench
messages at the internet-layer (rather than at the transport layer)
is a performance optimization that is permitted by this update.

4. Updating RFC 1812

This document hereby updates Section 4.3.3.3 of [RFC1812] as follows:
A router MUST ignore any ICMP Source Quench messages it receives.

The consensus of the TSV WG was that there are no valid reasons for a router to react to ICMP Source Quench messages in the current Internet.

5. Clarification for UDP, SCTP, and DCCP

UDP did not explicitly specify support for ICMP Source Quench messages. Hereby we clarify that UDP end-points MUST silently discard received ICMP Source Quench messages.

It is understood that SCTP and DCCP did not specify support for processing received ICMP Source Quench messages. Hereby we clarify that DCCP and SCTP end-points MUST silently discard received ICMP Source Quench messages.

6. General Advice to Transport Protocols

If a Source Quench message is received by any other transport-protocol instance, it MUST be silently ignored.

The TSV WG is not aware of any use that requires processing of these messages, and therefore expects other transports to follow the recommendations in Section 3. Note that for IETF-specified transports, this document formally deprecates reaction to ICMP Source Quench messages, and that generation of ICMP Source Quench messages has been deprecated for both hosts and routers. Therefore, future applications can not expect to receive these messages.

7. Recommendation Regarding RFC 1016

RFC 1016 [RFC1016] described an experimental approach to ICMP Source Quench message handling in hosts that was being thought about in 1987. The IETF notes that RFC 1016 has never been on the IETF standards-track, but for clarity and avoidance of doubt, we note that the approach described in RFC 1016 [RFC1016] MUST NOT be implemented.

8. Security Considerations

ICMP Source Quench messages could be leveraged for performing blind throughput-reduction attacks against TCP and similar protocols. This attack vector, along with possible countermeasures, has been discussed in great detail in [RFC5927] and [CPNI-TCP]. Silently
ignoring ICMP Source Quench messages, as specified in this document, eliminates the aforementioned attack vector.

For current TCP implementations, receipt of an ICMP Source Quench message should not result in security issues because, as noted in [RFC5927] and [CPNI-TCP], virtually all current versions of popular TCP implementations already silently ignore ICMP Source Quench messages. This is also the case for SCTP and DCCP implementations.

Hosts, security gateways, and firewalls MUST silently discard received ICMP Source Quench packets and SHOULD log such drops as a security fault with at least minimal details (IP Source Address, IP Destination Address, ICMP message type, and date/time the packet was seen).

We note that security devices such as the Snort Network Intrusion Detection System (NIDS) has logged ICMP Source Quench messages as such for more than ten years. [Anderson2002].

9. IANA Considerations

IANA is requested to mark ICMP type 4 (Source Quench) as "Deprecated" in the ICMP Parameters registry [ICMPPARREG] with a reference to this document.

10. Acknowledgements

The author of this document would like to thank Ran Atkinson, who contributed text that was incorporated into this document and also provided valuable feedback on earlier versions of this document.

The author of this document would like to thank (in alphabetical order) Fred Baker, David Black, Scott Bradner, James Carlson, Antonio De Simone, Wesley Eddy, Gorry Fairhurst, Alfred Hoenes, Mahesh Jethanandani, Kathleen Moriarty, Carlos Pignataro, James Polk, Anantha Ramaiyah, Randall Stewart, Dan Wing, and Andrew Yourtchenko, for providing valuable feedback on earlier versions of this document.

This document has benefited from discussions within the TCPM Working Group while working on [RFC5927].

11. References
11.1. Normative References


11.2. Informative References


[OpenBSD] The OpenBSD Project, "http://www.openbsd.org".
Appendix A. Survey of support of ICMP Source Quench in some popular TCP/IP implementations

A large number of implementations completely ignore ICMP Source Quench messages meant for TCP connections. This behavior has been implemented in, at least, Linux [Linux] since 2004, and in FreeBSD [FreeBSD], NetBSD [NetBSD], OpenBSD [OpenBSD], and Solaris 10 since 2005. Additionally, OpenSolaris [OpenSolaris] has always shipped with support for ICMP Source Quench messages disabled.

Appendix B. Changes from previous versions of the draft (to be removed by the RFC Editor before publishing this document as an RFC)

B.1. Changes from draft-ietf-tsvwg-source-quench-05
   o Fixes minor writeo in Section 7.

B.2. Changes from draft-ietf-tsvwg-source-quench-04
   o Removes request to move RFC 1016 to "Historic" status.
   o Updates the Security Considerations section.

B.3. Changes from draft-ietf-tsvwg-source-quench-03
   o Added 'Obsoletes' metadata, and moved the reference to [RFC1016] from the 'Normative References' to the 'Informative References'.
B.4. Changes from draft-ietf-tsvwg-source-quench-02

- Clarifies the requirements language.

B.5. Changes from draft-ietf-tsvwg-source-quench-01

- Changes deprecation of ICMP SQ from "SHOULD NOT" to "MUST NOT" in response of feedback from Scott Bradner and the TSV WG.

B.6. Changes from draft-ietf-tsvwg-source-quench-00

- Discusses the motivation for deprecating ICMP Source Quench messages (as suggested by Anantha Ramaiah).
- Incorporates IANA considerations such that ICMP Source Quench messages are deprecated in the corresponding registry.

B.7. Changes from draft-gont-tsvwg-source-quench-01

- Addresses nits and editorial changes suggested by Gorry Fairhurst.
- Added the status of Solaris and OpenSolaris to Appendix A.
- Document resubmitted as draft-ietf.

B.8. Changes from draft-gont-tsvwg-source-quench-00

- This revision reflects the recent discussion about ICMP Source Quench messages on the tsvwg mailing-list. A detailed list of the changes is available at:
  http://www.ietf.org/mail-archive/web/tsvwg/current/msg10407.html

Author’s Address

Fernando Gont
UTN-FRH / SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: http://www.si6networks.com
New Differentiated Services Code Point Assignments
for Rich Media Traffic
draft-polk-tsvwg-new-dscp-assignments-00.txt

Abstract

This document requests five new Differentiated Services Code Point (DSCP) values from the Internet Assigned Numbers Authority (IANA) for new classes of rich media traffic and one additional DSCP value for the signaling of multimedia sessions.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 5, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

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

This document requests five new Differentiated Services Code Point (DSCP) values (DSCP) from the Internet Assigned Numbers Authority (IANA) for new classes of rich media traffic and one additional DSCP value for the signaling of multimedia sessions. Four of the six new DSCP values are for traffic classes that are admitted by the network using an additional Capacity-Admission signaling procedure to the normal signaling that occurs between multiple endpoints establishing a traffic flow between endpoints. The additional capacity-admission signaling procedure is offered in RFC 5865 [RFC5865], which defined the Voice-Admit per hop behavior (PHB) DSCP. Each of these four traffic classes can conform to the Expedited Forwarding Per-Hop Behavior, if configured to do so, using the Priority Queuing system such as that defined in Section 1.4.1.1 of [ID-4594-UP].

It is expected that voice and video media samples will be carried using the Real-time Transport Protocol (RTP) [RFC3550], thus making voice by itself indistinguishable from video to routers and switches, unless one of two things occurs:

- Deep packet inspection (DPI) at the ingress of each DiffServ edge node to determine that the packet is an RTP packet with a certain codec that properly identifies it as either a voice or video packet, or
- have a separate marking for the packets (i.e., a different DSCP).

It is certainly the case that voice samples/frames can be in the same packet as video frames, thus making the packet marked either voice or video, but that will have to be left to the application to decide if that is a good idea. For what it is worth, most current implementations of mixing the media types have the packets marked as a video.
This effort is based on the work started in RFC 5865 [RFC5865], a Differentiated Services Code Point for Capacity-Admitted Traffic voice only traffic, which recommends the classes created within RFC 4594 [RFC4594] be extended for video traffic flows of different types. Nearly all of what is requested and referenced here is based on what started in RFC 4594, but with video as the dominant application as RFC 5865 recommends. Presently, RFC 4594 is being updated by [ID-4594-UP] for many reasons, including the inclusion of these six new DSCPs.

These four new video classes differ from their existing counterparts in behavior by not being subjected to capacity admission. All of the mentioned traffic classes and subsequent DSCPs within RFC 4594 are non-binding, given that it is a non-normative RFC. RFC 4594 also did not recommend the need for capacity admission traffic classes (aka with associated DSCP values). This document is symbiotic with [ID-4594-UP] which intends to replace RFC 4594 as a standards track update which includes the new DSCP assignments created within this document.

Thus, RFC 4594 defined the need for application assignment of certain DSCPs, but only non-normatively. RFC 5865 defined updated DSCP values for a capacity-admitted voice traffic class that is normative. This document takes what was in RFC 4594, creates 4 new capacity-admitted traffic classes and associated DSCPs. This document also moves one non-capacity-admitted traffic class as well as moves the recommended audio/video signaling DSCP value to another value.

Within RFC 5865, there is the specific call for additional DSCPs for capacity-admitted traffic flows of real-time rich media (video) flows in Section 3 of that document under the heading "Summary: Changes from RFC 4594".

It should be noted here that these flows are typically video flows, and frequently include the audio with the adjoining video traffic within that flow. The details of how that gets sorted out are outside the scope of this document. DiffServ is a known and proven mechanism. This document does not change or challenge the idea that Differentiated Services is a Per Hop Behavior (PHB) mechanism, and does not create a service. Here we merely want to add new DSCP assignments because of how at least some of the world is (or wants to) differentiate video from other traffic, including other video traffic.

Section 3 will discuss some of the evolution of DSCP assignments, focusing on those aspects pertinent to the creation of these six new DSCP values. Section 4 describes and defines each of the six DSCP values being requested. Heavy reliance exists on the text of RFC 5865 for its diagrams and charts. Those were not brought into this document at this time, but could be in the future.
2. Terminology

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

CAC - defined in RFC 5865

PHB - defined in RFC 5865

DSCP - defined in RFC 5865

Queue - defined in RFC 5865

3. Evolution of the Proposed DSCPs

First of all, full consideration of PHBs and DSCPs needs to originate with RFC 2474. Section 6 of that document states the following:

"The DSCP field within the DS field is capable of conveying 64 distinct codepoints. The codepoint space is divided into three pools for the purpose of codepoint assignment and management: a pool of 32 RECOMMENDED codepoints (Pool 1) to be assigned by Standards Action as defined in [CONS], a pool of 16 codepoints (Pool 2) to be reserved for experimental or Local Use (EXP/LU) as defined in [CONS], and a pool of 16 codepoints (Pool 3) which are initially available for experimental or local use, but which should be preferentially utilized for standardized assignments if Pool 1 is ever exhausted. The pools are defined in the following table (where 'x' refers to either '0' or '1'):

<table>
<thead>
<tr>
<th>Pool</th>
<th>Codepoint space</th>
<th>Assignment Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>xxxxx0</td>
<td>Standards Action</td>
</tr>
<tr>
<td>2</td>
<td>xxxx11</td>
<td>EXP/LU (*)</td>
</tr>
<tr>
<td>3</td>
<td>xxxx01</td>
<td>EXP/LU (*)</td>
</tr>
</tbody>
</table>

(*) may be utilized for future Standards Action allocations as Necessary"

The key part of the above quote is

"... which should be preferentially utilized for standardized assignments if Pool 1 is ever exhausted..."

which we here take to mean ‘SHOULD NOT use unless you have a really good reason to use’. We propose what we consider a really good
reason to use some of the assignments from Pool 3 before Pool 1 is exhausted. One reason for assigning out of Pool 3 is to get similar marking from layer 2 technologies that only have 3 bits to use for their value, not 6 bits. Technologies such as 802.3 Ethernet, 802.11 Wireless Ethernet, and MPLS are 3 examples of technologies that only have 3 bits to use.

[Editor’s Note: If this aspect of assigning DSCPs from Pool 3 before Pool 1 is exhausted requires an update to RFC 2474, please let the authors know so we can point this out to the community for additional feedback.]

Just as RFC 5865 matched the first 3 (or 4) bits with EF for Voice-Admit (101110 and 101100), we RECOMMEND the admitted DSCP for an existing value be its XXXX01 version of the non-admitted DSCP (XXXXX0). We note that the last two bits MUST NOT be x11 because that would mean the value is a Pool 2 value, which is forbidden currently by RFC 2474.

Thus, a DSCP value commonly traverses a layer 2 device by ignoring the last 3 bits of the DSCP value, i.e., taking EF, which is 101110, and reducing it to 101 only, and transmitting this over the layer 2 infrastructure.

RFC 4954, and its intended replacement document [ID-4594-UP], create several service classes primarily intended for video traffic with slightly different characteristics. It was stated there that not all video DSCP values from RFC 4594 are expected to be within the same network, but that could be the case.

RFC 4594 listed these voice and video services classes:

- "Telephony" using the EF DSCP
- "Realtime Interactive" using the CS4 DSCP
- "Multimedia Conferencing" using the AF4X DSCP
- "Multimedia Streaming" using the AF3X DSCP
- "Broadcast Video" using the CS3 DSCP

Plus, for Telephony Signaling

- "Signaling" using the CS5 DSCP

[ID-4594-UP] lists these ‘non-admitted’ voice and video services classes (some with changed service names, as well as some DSCPs changed):

- Audio using the EF DSCP
Internet-Draft        New Rich Media (Video) DSCPs             Mar 2012

- Video using the AF4X DSCP
- Hi-Res using the CS4 DSCP
- Realtime-Interactive using the CS5 DSCP
- Multimedia Streaming using the AF3X DSCP
- Broadcast using the CS3 DSCP

The Multimedia Conferencing purpose and meaning has been changed within [ID-DSCP-UP], as has its DSCPs, which will be listed in the next set of bullets and defined within this document.

RFC 5865 created the new capacity-admitted Voice-Admit, which mentions specifically that a reservation protocol, "such as RSVP" is used to establish those sessions or traffic flows.

This document creates six additional services classes that are incorporated into [ID-4594-UP]:

- Hi-Res-Admit using the CS4-Admit (100001) DSCP
- Realtime-Interactive-Admit using the CS5-Admit (101001) DSCP
- Multimedia Conferencing using the MC (011101) DSCP
- Multimedia Conferencing-Admit using the MC-Admit (100101) DSCP
- Broadcast-Admit using the CS3-Admit (011001) DSCP

Plus, for Conversational Signaling (a term described in [ID-4594-UP]), which is no longer to use the CS5 DSCP,

- "A/V-Sig" using the 010001 DSCP

The results of this are that the

- CS4-Admit is the xxxxx1 version of CS4.
- CS5-Admit is the xxxxx1 version of CS5.
- CS3-Admit is the xxxxx1 version of CS3.

MC-Admit is not the xxxxxx1 version of the new MC DSCP value (100101), because there are no more 100xxx values that are available, outside of the two x11 values from Pool 2, which cannot be assigned for public use.

[Editor’s Note: The author is open to suggestions from the community for how to resolve this issue, if anyone considers it an issue.]
The new goal for the signaling service class is to not be starved. It has been shown that mission critical voice and video call set-up does not require expedited forwarding as a PHB. However, this service class MUST NOT be starved, and so it is RECOMMENDED to use a codepoint similar in characteristics to the RFC 4594 (and [ID-4594-UP] defined Low-Latency Data service class of 010xxx.

4. New DSCP Assignments

4.1 The CS5-Admit PHB

'CS5-Admit' MUST be used with a capacity-admission signaling procedure similar to what is required of 'Voice-Admit' [RFC5865]. RSVP [RFC2205] and NSIS [RFC4080] are two good examples for data-path signaling for capacity-admission. Neither is mandatory, but one of them SHOULD be used.

CS5-Admit has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for CS5-Admit is 101001.

4.2 The CS4-Admit DSCP

'CS4-Admit' MUST be used with a capacity-admission signaling procedure similar to what is required of 'Voice-Admit' [RFC5865]. RSVP [RFC2205] and NSIS [RFC4080] are two good examples for data-path signaling for capacity-admission. Neither is mandatory, but one of them SHOULD be used.

CS4-Admit has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for CS4-Admit is 100001.

4.3 The CS3-Admit DSCP

'CS3-Admit' MUST be used with a capacity-admission signaling procedure similar to what is required of 'Voice-Admit' [RFC5865]. RSVP [RFC2205] and NSIS [RFC4080] are two good examples for data-path signaling for capacity-admission. Neither is mandatory, but one of them SHOULD be used.

CS3-Admit has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for CS3-Admit is 011001.
4.4 The MC DSCP

'MC' SHOULD NOT use a capacity-admission signaling procedure. Rather, the MC-Admit is used with a capacity-admission signaling procedure if needed. This PHB MUST be non-admitted.

MC has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for MC is 011001.

4.5 The MC-Admit DSCP

'MC-Admit' MUST be used with a capacity-admission signaling procedure similar to what is required of 'Voice-Admit' [RFC5865]. RSVP [RFC2205] and NSIS [RFC4080] are two good examples for data-path signaling for capacity-admission. Neither is mandatory, but one of them SHOULD be used.

MC-Admit has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for MC-Admit is 100101.

4.6 The Conversational Signaling (A/V-Sig) DSCP

'A/V-Sig' MUST be used with a capacity-admission signaling procedure similar to what is required of 'Voice-Admit' [RFC5865]. RSVP [RFC2205] and NSIS [RFC4080] are two good examples for data-path signaling for capacity-admission. Neither is mandatory, but one of them SHOULD be used.

A/V-Sig has traffic characteristics described in [ID-4594-UP].

The DSCP value requested for A/V-Sig is 010001.

5. Acknowledgements

The author would like to thank Paul Jones, Glen Lavers, Mo Zanaty, David Benham, Michael Ramalho for their comments and questions about this effort that ultimately helped shape this document.

6. IANA Considerations

IANA is requested to make the following registry assignments from Pool 1 and Pool 3 from the dscp-parameters section within IANA. Justification for assigning from Pool 3 is in Section 3 of this document, and are the only possible parallel assignments to existing assignments of similar registries - very much for the reason Voice-Admit [RFC5865] was assigned a codepoint similar to EF. That
aspect is the main point of this document.

6.1 DSCP Assignments from Pool 1

The code points described in this document is referred to as the following from Pool 1 and has been requested as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Space</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/V-Sig</td>
<td>010010</td>
<td>[this document]</td>
</tr>
</tbody>
</table>

6.2 DSCP Assignments from Pool 3

The code points described in this document is referred to as the following from Pool 3 and has been requested as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Space</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS5-Admit</td>
<td>101001</td>
<td>[this document]</td>
</tr>
<tr>
<td>CS4-Admit</td>
<td>100001</td>
<td>[this document]</td>
</tr>
<tr>
<td>CS3-Admit</td>
<td>011001</td>
<td>[this document]</td>
</tr>
<tr>
<td>MC-Admit</td>
<td>100101</td>
<td>[this document]</td>
</tr>
<tr>
<td>MC</td>
<td>011001</td>
<td>[this document]</td>
</tr>
</tbody>
</table>

7. Security Considerations

The Security Considerations are identical to those of RFC 5865.

Every newly proposed DSCP (save A/V-Sig) serves the same security risk and properties of the Voice-Admit DSCP. Section 3 of this document discusses why these DSCP values are should be parallel to their non-admitted counterparts, just as Voice-Admit states in RFC 5865 it is parallel to the existing (at the time) EF.

The A/V-Sig merely has a new DSCP name, RFC 4594 currently has this service class called "Signaling", serving the same purpose.

8. References

8.1 Normative References
8.2 Informative References


Author’s Addresses

James Polk
3913 Treemont Circle
Colleyville, Texas  76034

Phone: +1.817.271.3552
Email: jmpolk@cisco.com
Abstract

This document describes service classes configured with Diffserv and identifies how they are used and how to construct them using Differentiated Services Code Points (DSCPs), traffic conditioners, Per-Hop Behaviors (PHBs), and Active Queue Management (AQM) mechanisms. There is no intrinsic requirement that particular DSCPs, traffic conditioners, PHBs, and AQM be used for a certain service class, but for consistent behavior under the same network conditions, configuring networks as described here is appropriate.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 5, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in
Table of Contents

1. Introduction ...................................................3
   1.1. Requirements Notation .....................................6
   1.2. Expected Use in the Network ...............................6
   1.3. Service Class Definition ..................................7
   1.4. Key Differentiated Services Concepts ......................8
       1.4.1. Queuing ............................................8
       1.4.1.1. Priority Queuing ..........................9
       1.4.1.2. Rate Queuing ..............................9
   1.4.2. Active Queue Management ................................9
   1.4.3. Traffic Conditioning ................................10
   1.4.4. Differentiated Services Code Point (DSCP) ..........11
   1.4.5. Per-Hop Behavior (PHB) ................................11
   1.5. Key Service Concepts .....................................11
       1.5.1. Default Forwarding (DF) ...........................12
       1.5.2. Assured Forwarding (AF) ...........................12
       1.5.3. Expedited Forwarding (EF) ..........................12
       1.5.4. Class Selector (CS) ...............................13
       1.5.5. Admission Control ..................................14
2. Service Differentiation .......................................14
   2.1. Service Classes ..........................................15
   2.2. Categorization of User Oriented Service Classes ........17
   2.3. Service Class Characteristics ............................20
3. Network Control Traffic .......................................26
   3.1. Current Practice in the Internet .........................26
   3.2. Network Control Service Class ............................26
   3.3. OAM Service Class ........................................28
4. User Oriented Traffic .........................................30
   4.1. Conversational Service Class Group .......................31
       4.1.1 Audio Service Class ................................31
       4.1.1 Video Service Class ................................34
       4.1.1 Hi-Res Service Class ................................38
   4.2. Realtime-Interactive Service Class ........................42
   4.3. Multimedia Conferencing Service Class .................44
   4.4. Multimedia Streaming Service Class ......................46
   4.5. Broadcast Video Service Class ............................49
   4.6. Low-Latency Data Service Class ..........................52
   4.7. Conversational Signaling Service Class .................54
   4.8. High-Throughput Data Service Class .......................56
   4.9. Standard Service Class ..................................59
   4.10. Low-Priority Data ......................................60
5. Additional Information on Service Class Usage .................61
   5.1. Mapping for NTP ..........................................61
   5.2. VPN Service Mapping .....................................61
6. Security Considerations .......................................62
7. Contributing Authors ..........................................63
1. Introduction

Differentiated Services [RFC2474][RFC2475] provides the ability to mark/label/classify IP packets differently to distinguish how individual packets need to be treated differently through (or throughout) a network on a per hop basis. Local administrators are who configure each router for which Differentiated Services Code Points (DSCP) are to be treated differently, which are to be ignored (i.e., no differentiated treatment), and which DSCPs are to have their packets remarked (to different DSCPs) as they pass through a router. Local administrators are also who assign which applications, or traffic types, should use which DSCPs to receive the treatment the administrators expect within their network.

What most people fail to understand is that DSCPs provide a per hop behavior (PHB) through that router, but not the previous or next router. In this way of understanding PHB markings, one can understand that Differentiated Services (Diffserv) is not a Quality of Service (QoS) mechanism, but rather a Classification of Service (CoS) mechanism.

For instance, there are 64 possible DSCP values, i.e., using 6 bits of the old Type of Service (TOS) byte [RFC0791]. Each can be configured locally to have greater or less treatment relative to any other DSCP with two exceptions*.  

* Expedited Forwarding (EF) [RFC3246] DSCPs have a treatment requirement that any packet marked within an EF class has to be the next packet transmitted out its egress interface. If there are more than one EF marked packet in the queue, obviously the queue sets the order they are transmitted. Further, if there are more than one EF DSCP, local configuration determines if each are treated the same or differently relate to each other EF DSCP. Currently, there are two Expedited Forwarding DSCPs: EF (101110) [RFC3246] and VOICE-ADMIT (101100) [RFC5865].

* Class Selector 6 (CS6) [RFC2474] is for routing protocol traffic. There are deemed important because if the network does not transmit and receive its routing protocol traffic in a timely manner, the network stops operating properly.

Not all are configured to mean anything other than best effort forwarding by local administrators of a network. Let us say there
are 5 DSCPs configured within network A. Network A’s administrator chooses and configures which order (obeying the two exceptions noted above) which application packets are treated differently than any other packets within that network (A). The DSCPs are not fixed to a linear order for relative priority on a per hop basis. Further, and this is often the case, there might be packets with the same DSCP arriving at multiple interfaces of a node, each egressing that node out the same interface. At ingress to this node, everything was fine, with no poor behavior or noticeably excessive amount of packets with the same DSCP. However, at the egress interface, there might not be enough capacity to satisfy the load, thus the departing packets transmit at their maximum rate for that DSCP, but have additional latency due to the overload within that one node. This is called fan-in (congestion or problem). By itself, DiffServ will not remedy this problem for the application that is intolerant to added latency because DiffServ only functions within 1 node at a time.

An additional mechanism is needed to ensure each flow or session receives the amount of packets at its destination that the application requires to perform properly; a mechanism such as IntServ, by way of RSVP [RFC2205] or NSIS [RFC4080]. With this added capability to be session aware, something DiffServ is not, the packets transmitted within a single session have a very good probability of arriving in such a way the receiving application can make full use of each. That said, signaling reservations for each session or flow adds complexity, which creates more work for those who maintain and administer such a network. Adding bandwidth and using DiffServ marking is an easier pill to swallow. The deployment of not few, but more and more audio and (particularly bandwidth hogging) video codecs and their respective application rigidity has caused some to conclude that throwing bandwidth at the problem is no longer acceptable.

With this in mind, this document incorporates five of the six new DSCPs from [ID-DSCP] identified as capacity-admitted DSCPs for most of the service classes in this document. As explained in [ID-DSCP], the five new capacity-admitted DSCPs are from Pool 3. [ID-DSCP] goes further to explain that may layer 2 technologies fewer bits for marking and prioritization. Instead of six bits like DiffServ, they have three bits, which yields a maximum of 8 values, which tend to line up quite well with the TOS field values. Thus, aggregation of DSCPs is typically accomplished by simply ignoring or reducing the number of bits used to the most significant ones available, such as EF is 101110, at layer 2 this is merely 101;

Broadcast is 011000, at layer 2 this is merely 011.

This document is originally built upon the RFC 4594 effort, while updating some of the usages and expanding the scope for newer applications that are in use today.
Service class definitions are based on the different traffic characteristics and required performance of the applications/services. There are a greater number of service classes in this document than there were when RFC 4594 [RFC4594] was published (the RFC this document intends to obsolete). The required performance of applications/services has also changed since the publication of RFC 4594, specifically in the area of conversational real time communications. As a result, this document has a greater number of real time applications with more granular set of DSCPs due to their different required performances. Like RFC 4594 before, this approach allows those applications with similar traffic characteristics and performance requirements to be placed in the same service class.

The notion of traffic characteristics and required performance is a per application concept, therefore the label name of each service class remains the same on an end-to-end basis, even if we understand that Diffserv is only a PHB and cannot guarantee anything, even packet delivery at the intended destination node. That said, several applications can be configured to have the same DSCP, or each have different DSCPs that have the same treatment per hop within a network.

Since RFC 4594 was first published, a new concept has been introduced that will appear throughout this document, including DSCP assignments -- the idea of "admitted" traffic, initially introduced into Diffserv within RFC 5865 [RFC5865]. The VOICE-ADMIT Expedited Forwarding class differentiates itself from the EF Expedited Forwarding by having the packets marked be for admitted traffic. This concept of "admitted" traffic is spread throughout the real time traffic classes.

Thus, the document flow is as follows:

- maintain the general format of RFC 4594;
- augment the content with the concept of capacity-admission;
- incorporate much more video into this document, as it has become a dominant application in enterprises and other managed networks, as well as on the open public Internet;
- reduce the discussion on voice and its examples;
- articulate the subtle differences learned since RFC 4594 was published.

The goal here is to provide a standard configuration for DiffServ DSCP assignments and expected PHBs for enterprises and other managed networks, as well as towards the public Internet with specific traffic characteristics per Service class/DSCP, and example applications shown for each.
This document describes service classes configured with Diffserv and recommends how they can be used and how to construct them using Differentiated Services Code Points (DSCPs), traffic conditioners, Per-Hop Behaviors (PHBs), and Active Queue Management (AQM) mechanisms. There is no intrinsic requirement that particular DSCPs, traffic conditioners, PHBs, and AQM be used for a certain service class, but as a policy and for interoperability it is useful to apply them consistently.

We differentiate services and their characteristics in Section 2. Network control traffic, as well as user oriented traffic are discussed in Sections 3 and 4, respectively. We analyze the security considerations in Section 6. Section 7 offers a tribute to the authors of RFC 4594, from which this document is based. It is in its own section, and not part of the normal acknowledgements portion of each IETF document.

1.1. Requirements Notation

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

1.2. Expected Use in the Network

In the Internet today, corporate LANs and ISP WANs are increasingly utilized, to the point in which network congestion is affecting performance of applications. For this reason, congestion, loss, and variation in delay within corporate LANs and ISP backbones is becoming known to the users collectively as "the network is slow for this application" or just "right now" or "for today". Users do not directly detect network congestion. They react to applications that run slow, or to downloads that take too long in their mind(s). The explosion of video traffic on the internet recently has cause much of this, and is often the application the user is using when they have this slowness.

In the past, application slowness occurred for three very good reasons.

- the networks the user oriented traffic traverses moves through cycles of bandwidth boom and bandwidth bust, the latter of which become apparent with the periodic deployment of new bandwidth-hungry applications.
- In access networks, the state is often different. This may be because throughput rates are artificially limited or over-subscribed, or because of access network design trade-offs.
- Other characteristics, such as database design on web servers.
that may create contention points, e.g., in filestore) and configuration of firewalls and routers, often look externally like a bandwidth limitation.

The intent of this document is to provide a standardized marking, plus a conditioning and packet treatment strategy so that it can be configured and put into service on any link that is itself congested.

1.3. Service Class Definition

A "service class" represents a similar set of traffic characteristics for delay, loss, and jitter as packets traverse routers in a network. For example, "High-Throughput Data" service class for store-and-forward applications, or a "Broadcast" service class for minimally time-shifted IPTV or Internet radio broadcasts. Such a service class may be defined locally in a Differentiated Services (DS) domain, or across multiple DS domains, possibly extending end to end. A goal of this document is to have most/all networks assign the same type of traffic the same for consistency.

A service class is a naming convention which is defined as a word, phrase or initialism/acronym representing a set of necessary traffic characteristics of a certain type of data flow. The necessary characteristics of these traffic flows can be realized by the use of defined per-hop behavior that started with [RFC2474]. The actual specification of the expected treatment of a traffic aggregate within a domain may also be defined as a per-domain behavior (PDB) [RFC3086].

Each domain will locally choose to

- implement one or more service classes with traffic characteristics as defined here, or
- implement one or more service classes with similar traffic characteristics as defined here, or
- implement one or more service classes with similar traffic characteristics as defined here and to aggregate one or more service classes to reduce the number of unique DSCPs within their network, or
- implement one or more non-standard service classes with traffic characteristics not as defined here, or
- not use Diffserv within their domain.

For example, low delay, low loss, and minimal jitter may be realized using the EF PHB, or with an over-provisioned AF PHB. This must be done with care as it may disrupt the end-to-end performance required
by the applications/services. If the packet sizes are similar within an application, but different between two applications, say small voice packets and large video packets, these two applications may not realize optimum results if merged into the same aggregate if there are any bottlenecks in the network. We provide for this flexibility on a per hop or per domain basis within this document.

This document provides standardized markings for traffic with similar characteristics, and usage expectations for PHBs for specific service classes for their consistent implementation.

The Default Forwarding "Standard" service class is REQUIRED; all other service classes are OPTIONAL. That said, each service class lists traffic characteristics that are expected when using that type of traffic. It is RECOMMENDED that applications and protocols that fit a certain traffic characteristic use the appropriate service class mark, i.e., the DSCP, for consistent behavior. It is expected that network administrators will base their endpoint application and router configuration choices on the level of service differentiation they require to meet the needs of their customers (i.e., their end-users).

1.4. Key Differentiated Services Concepts

In order to fully understand this document, a reader needs to familiarize themselves with the principles of the Differentiated Services Architecture [RFC2474]. We summarize some key concepts here only to provide convenience for the reader, the referenced RFCs providing the authoritative definitions.

1.4.1. Queuing

A queue is a data structure that holds packets that are awaiting transmission. A router interface can only transmit one packet at a time, however fast the interface speed is. If there is only 1 queue at an interface, the packets are transmitted in the order they are received into that queue - called FIFO, or "first in, first out". Sometimes there is a lag in the time between a packets arrives in the queue and when it is transmitted. This delay might be due to lack of bandwidth, or if there are multiple queues on that interface, because a packet is low in priority relative to other packets that are awaiting to transmit. The scheduler is the system entity that chooses which packet is next in line for transmission when more than one packet are awaiting transmission out the same router interface.

1.4.1.1 Priority Queuing

A priority queuing system is a combination of a set of queues and a
scheduler that empties the queues (of packets) in priority sequence. When asked for a packet, the scheduler inspects the highest priority queue and, if there is data present, returns a packet from that queue. Failing that, it inspects the next highest priority queue, and so on. A freeway onramp with a stoplight for one lane that allows vehicles in the high-occupancy-vehicle lane to pass is an example of a priority queuing system; the high-occupancy-vehicle lane represents the "queue" having priority.

In a priority queuing system, a packet in the highest priority queue will experience a readily calculated delay. This is proportional to the amount of data remaining to be serialized when the packet arrived plus the volume of the data already queued ahead of it in the same queue. The technical reason for using a priority queue relates exactly to this fact: it limits delay and variations in delay and should be used for traffic that has that requirement.

A priority queue or queuing system needs to avoid starvation of lower-priority queues. This may be achieved through a variety of means, such as admission control, rate control, or network engineering.

1.4.1.2. Rate Queuing

Similarly, a rate-based queuing system is a combination of a set of queues and a scheduler that empties each at a specified rate. An example of a rate-based queuing system is a road intersection with a stoplight. The stoplight acts as a scheduler, giving each lane a certain opportunity to pass traffic through the intersection.

In a rate-based queuing system, such as Weighted Fair Queuing (WFQ) or Weighted Round Robin (WRR), the delay that a packet in any given queue will experience depends on the parameters and occupancy of its queue and the parameters and occupancy of the queues it is competing with. A queue whose traffic arrival rate is much less than the rate at which it lets traffic depart will tend to be empty, and packets in it will experience nominal delays. A queue whose traffic arrival rate approximates or exceeds its departure rate will tend not to be empty, and packets in it will experience greater delay. Such a scheduler can impose a minimum rate, a maximum rate, or both, on any queue it touches.

1.4.2 Active Queue Management

Active Queue Management, or AQM, is a generic name for any of a variety of procedures that use packet dropping or marking to manage the depth of a queue. The canonical example of such a procedure is Random Early Detection (RED), in that a queue is assigned a minimum and maximum threshold, and the queuing algorithm maintains a moving average of the queue depth. While the mean queue depth exceeds the maximum threshold, all arriving traffic is dropped. While the mean queue depth exceeds the minimum threshold but not the maximum...
threshold, a randomly selected subset of arriving traffic is marked or dropped. This marking or dropping of traffic is intended to communicate with the sending system, causing its congestion avoidance algorithms to kick in. As a result of this behavior, it is reasonable to expect that TCP’s cyclic behavior is desynchronized and that the mean queue depth (and therefore delay) should normally approximate the minimum threshold.

A variation of the algorithm is applied in Assured Forwarding PHB [RFC2597], in that the behavior aggregate consists of traffic with multiple DSCP marks, which are intermingled in a common queue. Different minima and maxima are configured for the several DSCPs separately, such that traffic that exceeds a stated rate at ingress is more likely to be dropped or marked than traffic that is within its contracted rate.

1.4.3 Traffic Conditioning

In addition, at the first router in a network that a packet crosses, arriving traffic may be measured and dropped or marked according to a policy, or perhaps shaped on network ingress, as in "A Rate Adaptive Shaper for Differentiated Services" [RFC2963]. This may be used to bias feedback loops, as is done in "Assured Forwarding PHB" [RFC2597], or to limit the amount of traffic in a system, as is done in "Expedited Forwarding PHB" [RFC3246]. Such measurement procedures are collectively referred to as "traffic conditioners". Traffic conditioners are normally built using token bucket meters, for example with a committed rate and burst size, as in Section 1.5.3 of the DiffServ Model [RFC3290]. The Assured Forwarding PHB [RFC2597] uses a variation on a meter with multiple rate and burst size measurements to test and identify multiple levels of conformance.

Multiple rates and burst sizes can be realized using multiple levels of token buckets or more complex token buckets; these are implementation details. The following are some traffic conditioners that may be used in deployment of differentiated services:

- For Class Selector (CS) PHBs, a single token bucket meter to provide a rate plus burst size control.
- For Expedited Forwarding (EF) PHB, a single token bucket meter to provide a rate plus burst size control.
- For Assured Forwarding (AF) PHBs, usually two token bucket meters configured to provide behavior as outlined in "Two Rate Three Color Marker (trTCM)" [RFC2698] or "Single Rate Three Color Marker (srTCM)" [RFC2697]. The two-rate, three-color marker is used to enforce two rates, whereas the single-rate, three-color marker is used to enforce a committed rate with two burst lengths.
1.4.4 Differentiated Services Code Point (DSCP)

The DSCP is a number in the range 0..63 that is placed into an IP packet to mark it according to the class of traffic it belongs in. These are divided into 3 groups, or pools, defined in RFC 2474, arranged as follows:

- Pool-1 has 32 values designated for standards assignment (of the form ‘xxxxx0’).
- Pool-2 has 16 values designated for experimental or local use only (EXP/LU) assignment (of the form ‘xxxx11’).
- Pool-3 has 16 values designated for experimental or local use (EXP/LU) assignment (of the form ‘xxxx01’).

However, pool-3 is allowed to be assigned for one of two reasons,

#1 - if the values in pool-1 are exhausted, or
#2 - if there is a justifiable reason for assigning a pool-3 DSCP prior to pool-1’s exhaustion.

1.4.5 Per-Hop Behavior (PHB)

In the end, the mechanisms described above are combined to form a specified set of characteristics for handling different kinds of traffic, depending on the needs of the application. This document seeks to identify useful traffic aggregates and to specify what PHB should be applied to them.

1.5 Key Service Concepts

While Differentiated Services is a general architecture that may be used to implement a variety of services, three fundamental forwarding behaviors have been defined and characterized for general use. These are basic Default Forwarding (DF) behavior for elastic traffic, the Assured Forwarding (AF) behavior, and the Expedited Forwarding (EF) behavior for real-time (inelastic) traffic. The facts that four code points are recommended for AF and that one code point is recommended for EF are arbitrary choices, and the architecture allows any reasonable number of AF and EF classes simultaneously. The choice of four AF classes and one EF class in the current document is also arbitrary, and operators MAY choose to operate more or fewer of either.

The terms "elastic" and "real-time" are defined in [RFC1633], Section 3.1, as a way of understanding broad-brush application requirements. This document should be reviewed to obtain a broad understanding of the issues in quality of service, just as [RFC2475]
should be reviewed to understand the data plane architecture used in today’s Internet.

1.5.1 Default Forwarding (DF)

The basic forwarding behaviors applied to any class of traffic are those described in [RFC2474] and [RFC2309]. Best-effort service may be summarized as "I will accept your packets" and is typically configured with some bandwidth guarantee. Packets in transit may be lost, reordered, duplicated, or delayed at random. Generally, networks are engineered to limit this behavior, but changing traffic loads can push any network into such a state.

Application traffic in the internet that uses default forwarding is expected to be "elastic" in nature. By this, we mean that the sender of traffic will adjust its transmission rate in response to changes in available rate, loss, or delay.

For the basic best-effort service, a single DSCP value is provided to identify the traffic, a queue to store it, and active queue management to protect the network from it and to limit delays.

1.5.2 Assured Forwarding (AF)

The Assured Forwarding PHB [RFC2597] behavior is explicitly modeled on Frame Relay's Discard Eligible (DE) flag or ATM's Cell Loss Priority (CLP) capability. It is intended for networks that offer average-rate Service Level Agreements (SLAs) (as FR and ATM networks do). This is an enhanced best-effort service; traffic is expected to be "elastic" in nature. The receiver will detect loss or variation in delay in the network and provide feedback such that the sender adjusts its transmission rate to approximate available capacity.

For such behaviors, multiple DSCP values are provided (two or three, perhaps more using local values) to identify the traffic, a common queue to store the aggregate, and active queue management to protect the network from it and to limit delays. Traffic is metered as it enters the network, and traffic is variously marked depending on the arrival rate of the aggregate. The premise is that it is normal for users occasionally to use more capacity than their contract stipulates, perhaps up to some bound. However, if traffic should be marked or lost to manage the queue, this excess traffic will be marked or lost first.

1.5.3. Expedited Forwarding (EF)

The intent of Expedited Forwarding PHB [RFC3246] is to provide a building block for low-loss, low-delay, and low-jitter services. It can be used to build an enhanced best-effort service: traffic remains subject to loss due to line errors and reordering during
routing changes. However, using queuing techniques, the probability of delay or variation in delay is minimized. For this reason, it is generally used to carry voice and for transport of data information that requires "wire like" behavior through the IP network. Voice is an inelastic "real-time" application that sends packets at the rate the codec produces them, regardless of availability of capacity. As such, this service has the potential to disrupt or congest a network if not controlled. It also has the potential for abuse.

To protect the network, at minimum one SHOULD police traffic at various points to ensure that the design of a queue is not overrun, and then the traffic SHOULD be given a low-delay queue (often using priority, although it is asserted that a rate-based queue can do this) to ensure that variation in delay is not an issue, to meet application needs.

1.5.4 Class Selector (CS)

Class Selector, those DSCPs that end in zeros (xxx000), provide support for historical codepoint definitions and PHB requirement. The CS fields provide a limited backward compatibility with legacy practice, as described in [RFC2474], Section 4. Backward compatibility is addressed in two ways,

- First, there are per-hop behaviors that are already in widespread use (e.g., those satisfying the IPv4 Precedence queuing requirements specified in [RFC1812]), and

- this document will continue to permit their use in DS-compliant networks.

In addition, there are some DSCPs that correspond to historical use of the IP Precedence field,

- CS0 (000000) will remain ‘Default Forwarding’ (also know as ‘Best Effort’)

- 11xxxx will remain for routing traffic

and will map to PHBs that meet the general requirements specified in [RFC2474], Section 4.2.2.2.

No attempt is made to maintain backward compatibility with the "DTR" or Type of Service (TOS) bits of the IPv4 TOS octet, as defined in [RFC0791] and [RFC1349].

A DS-compliant network can be deployed exclusively by using one or more CS-compliant PHB groups. Thus, for example, codepoint ‘011000’ would map to the same PHB as codepoint ‘011010’.
1.5.5 Admission Control

Admission control (including refusal when policy thresholds are crossed) can ensure high-quality communication by ensuring the availability of bandwidth to carry a load. Inelastic real-time flows such as Voice over Internet Protocol (VoIP) (audio) or video conferencing services can benefit from use of an admission control mechanism, as generally the audio or video service is configured with over-subscription, meaning that some users may not be able to make a call during peak periods.

For VoIP (audio) service, a common approach is to use signaling protocols such as SIP, H.323, H.248, MEGACO, along with Resource Reservation Protocol (RSVP) to negotiate admittance and use of network transport capabilities. When a user has been authorized to send voice traffic, this admission procedure has verified that data rates will be within the capacity of the network that it will use. Many RTP voice and video payloads are inelastic and cannot react to loss or delay in any substantive way. For these payload types, the network needs to police at ingress to ensure that the voice traffic stays within its negotiated bounds. Having thus assured a predictable input rate, the network may use a priority queue to ensure nominal delay and variation in delay.

1.5.5.1 Capacity Admitted (*-Admit)

This is a newer group of traffic types that started with RFC 5865 and the Voice-Admit service type. Voice-Admit is an EF class marking but has capacity-admission always applied to it to ensure each of these flows are managed through a network, though not necessarily on an end-to-end basis. This depends on how many networks each flow transits and the load on each transited network. There are a series of new DSCPs proposed in [ID-DSCP], each specifying unique characteristics necessitating a separate marking from what existing before that document.

This document will import in four new ‘*-Admit’ DSCPs from [ID-DSCP], 2 others that are new but not capacity-admitted, one from RFC 5865, and change the existing usage of 2 DSCPs from RFC 4594. This is discussed throughout the rest of this document.

2. Service Differentiation

There are practical limits on the level of service differentiation that should be offered in the IP networks. We believe we have defined a practical approach in delivering service differentiation by defining different service classes that networks may choose to support in order to provide the appropriate level of behaviors and performance needed by current and future applications and services.
The defined structure for providing services allows several applications having similar traffic characteristics and performance requirements to be grouped into the same service class. This approach provides a lot of flexibility in providing the appropriate level of service differentiation for current and new, yet unknown applications without introducing significant changes to routers or network configurations when a new traffic type is added to the network.

2.1 Service Classes

Traffic flowing in a network can be classified in many different ways. We have chosen to divide it into two groupings, network control and user/subscriber traffic. To provide service differentiation, different service classes are defined in each grouping. The network control traffic group can further be divided into two service classes (see Section 3 for detailed definition of each service class):

- "Network Control" for routing and network control function.
- "OAM" (Operations, Administration, and Management) for network configuration and management functions.

The user/subscriber traffic group is broken down into ten service classes to provide service differentiation for all the different types of applications/services (see Section 4 for detailed definition of each service class):

- Conversational service group consists of five service classes:
  - Audio, which includes both ‘admitted’ and ‘unadmitted’ audio service classes, is for non-one way (i.e., generally bidirectional) audio media packets between human users of smaller size and at a constant delivery rate.
  - Hi-Res Video, which includes both ‘admitted’ and ‘unadmitted’ Hi-Res Video service classes, is for video traffic from higher end endpoints between human users necessitating different treatment that from desktop or video phone endpoints. This has a clearly business differentiation, and not a technical differentiation – as both Hi-Res-Video and Video will be treated similarly on the wire when no congestion occurs.
  - Video, which includes both ‘admitted’ and ‘unadmitted’ video service classes, is for video traffic from lower end endpoints between human users necessitating different treatment that from higher end (i.e., Telepresence) endpoints. This has a clearly business differentiation, and not a technical differentiation – as both Hi-Res-Video and Video will be treated similarly on the wire when no congestion occurs.
o Conversational Signaling service class is for peer-to-peer and client-server signaling and control functions using protocols such as SIP, H.323, H.248, and Media Gateway Control Protocol (MGCP). This traffic needs to not be starved on the network.

Editor’s note: RFC 4594 had this DSCP marking as CS5, but with clearly different characteristics (i.e., no sensitivity to jitter or (unreasonable) delay), this DSCP has been moved to a more appropriate (new) value, defined in [ID-DSCP].

o Real-Time Interactive, which includes both ‘admitted’ and ‘unadmitted’ Realtime-Interactive service class, is for bidirectional variable rate inelastic applications that require low jitter and loss and very low delay, such as interactive gaming applications that use RTP/UDP streams for game control commands, and Virtualized Desktop applications between the user and content source, typically in a centralized data center.

o Multimedia Conferencing, which includes both ‘admitted’ and ‘unadmitted’ multimedia conferencing service class, is for applications that require minimal delay, but not like those of realtime application requirements. This service class can be bursty in nature, as well as not transmit packets for some time. Applications such as presentation data or collaborative application sharing will use this service class.

o Multimedia Streaming, which includes both ‘admitted’ and ‘unadmitted’ multimedia streaming service class, is for one-way bufferable streaming media applications such as Video on Demand (VOD) and webcasts.

o Broadcast, which includes both ‘admitted’ and ‘unadmitted’ broadcast service class, is for inelastic streaming media applications that may be of constant or variable rate, requiring low jitter and very low packet loss, such as broadcast TV and live events, video surveillance, and security.

o Low-Latency Data service class is for data processing applications such as client/server interactions or Instant Messaging (IM) and Presence data.

o Conversational Signaling (A/V-Sig) service class is for all signaling messages, whether in-band (i.e., along the data path) or out-of-band (separate from the data path), for the purposes of setting up, maintaining, managing and terminating bi- or multi-directional realtime sessions.

o High-Throughput Data service class is for store and forward applications such as FTP and billing record transfer.
2.2 Categorization of User Oriented Service Classes

The ten defined user/subscriber service classes listed above can be grouped into a small number of application categories. For some application categories, it was felt that more than one service class was needed to provide service differentiation within that category due to the different traffic characteristic of the applications, control function, and the required flow behavior. Figure 1 provides a summary of service class grouping into four application categories.

Application Control Category

- The Conversational Signaling service class is intended to be used to control applications or user endpoints. Examples of protocols that would use this service class are SIP, XMPP or H.323 for voice and/or video over IP services. User signaling flows have similar performance requirements as Low-Latency Data, they require a separate DSCP to be distinguished other traffic and allow for a treatment that is unique.

Media-Oriented Category

Due to the vast number of new (in process of being deployed) and already-in-use media-oriented services in IP networks, seven service classes have been defined.

- Audio service class is intended for Voice-over-IP (VoIP) services. It may also be used for other applications that meet the defined traffic characteristics and performance requirements.

- Video service class is intended for Video over IP services. It may also be used for other applications that meet the defined traffic characteristics and performance requirements.

- Hi-Res service class is intended for higher end video services that have the same traffic characteristics as the video service class, but have a business requirement(s) to be treated differently. One example of this is Telepresence video applications.

- Realtime-Interactive service class is intended for inelastic applications such as desktop virtualization applications and for interactive gaming.
o Multimedia Conferencing service class is for everything about or within video conferencing solutions that does not include the voice or (human) video components. Several examples are
- the presentation data part of an IP conference (call).
- the application sharing part of an IP conference (call).
- the whiteboarding aspect of an IP conference (call).

Each of the above can be part of a lower end web-conferencing application or part of a higher end Telepresence video conference. Each also has the ability to reduce their transmission rate on detection of congestion. These flows can therefore be classified as rate adaptive and most often more elastic than their voice and video counterparts.

o Broadcast Video service class is to be used for inelastic traffic flows specifically with minimal buffering expected by the source or destination, which are intended for broadcast HDTV service, as well as for transport of live video (sports or concerts) and audio events.

o Multimedia Streaming service class is to be used for elastic multimedia traffic flows where buffering is expected. This is the fundamental difference between the Broadcast and multimedia streaming service classes. Multimedia streaming content is typically stored before being transmitted. It is also buffered at the receiving end before being played out. The buffering is sufficiently large to accommodate any variation in transmission rate that is encountered in the network. Multimedia entertainment over IP delivery services that are being developed can generate both elastic and inelastic traffic flows; therefore, two service classes are defined to address this space, respectively: Multimedia Streaming and Broadcast Video.

Data Category

The data category is divided into three service classes.

o Low-Latency Data for applications/services that require low delay or latency for bursty but short-lived flows.

o High-Throughput Data for applications/services that require good throughput for long-lived bursty flows. High Throughput and Multimedia Streaming are close in their traffic flow characteristics with High Throughput being a bit more bursty and not as long-lived as Multimedia Streaming.

o Low-Priority Data for applications or services that can tolerate short or long interruptions of packet flows. The Low-Priority
Data service class can be viewed as "don’t care" to some degree.

Best-Effort Category

- All traffic that is not differentiated in the network falls into this category and is mapped into the Standard service class. If a packet is marked with a DSCP value that is not supported in the network, it SHOULD be forwarded using the Standard service class.

Figure 1, below, provides a grouping of the defined user/subscriber service classes into four categories, with indications of which ones use an independent flow for signaling or control; type of flow behavior (elastic, rate adaptive, or inelastic); and the last column provides end user Class of Service (CoS) rating as defined in ITU-T Recommendation G.1010.

<table>
<thead>
<tr>
<th>Application Categories</th>
<th>Service Class</th>
<th>Signaled Flow Behavior</th>
<th>G.1010 Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/V Sig Control</td>
<td>Real-Time Interactive</td>
<td>Yes</td>
<td>Inelastic</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>Yes</td>
<td>Inelastic</td>
</tr>
<tr>
<td></td>
<td>Video</td>
<td>Yes</td>
<td>Inelastic</td>
</tr>
<tr>
<td>Media-Oriented</td>
<td>Hi-Res</td>
<td>Yes</td>
<td>Inelastic</td>
</tr>
<tr>
<td></td>
<td>Multimedia Conferencing</td>
<td>Yes</td>
<td>Rate Adaptive</td>
</tr>
<tr>
<td></td>
<td>Broadcast</td>
<td>Yes</td>
<td>Inelastic</td>
</tr>
<tr>
<td></td>
<td>Multimedia Streaming</td>
<td>Yes</td>
<td>Elastic</td>
</tr>
<tr>
<td>Data</td>
<td>Low-Latency Data</td>
<td>No</td>
<td>Elastic</td>
</tr>
<tr>
<td></td>
<td>High-Throughput Data</td>
<td>No</td>
<td>Elastic</td>
</tr>
<tr>
<td></td>
<td>Low-Priority Data</td>
<td>No</td>
<td>Elastic</td>
</tr>
<tr>
<td>Best Effort</td>
<td>Standard</td>
<td>Not Specified</td>
<td>Non-critical</td>
</tr>
</tbody>
</table>
Here is a short explanation of the end user CoS category as defined in ITU-T Recommendation G.1010. User oriented traffic is divided into four different categories, namely, interactive, responsive, timely, and non-critical. An example of interactive traffic is between two humans and is most sensitive to delay, loss, and jitter. Another example of interactive traffic is between two servers where very low delay and loss are needed. Responsive traffic is typically between a human and a server but can also be between two servers. Responsive traffic is less affected by jitter and can tolerate longer delays than interactive traffic. Timely traffic is either between servers or servers and humans and the delay tolerance is significantly longer than responsive traffic. Non-critical traffic is normally between servers/machines where delivery may be delay for period of time.

2.3. Service Class Characteristics

This document specifies what network administrators are to expect when configuring service classes identified by their differing characteristics. Figure 2 identifies these service classes along with their characteristics, as well as the tolerance to loss, delay and jitter for each service class. Properly engineered networks to these PHBs will achieve expected results. That said, not all of the identified service classes are expected in each operator’s network.

<table>
<thead>
<tr>
<th>Service Class Name</th>
<th>Traffic Characteristics</th>
<th>Tolerance to Loss</th>
<th>Delay</th>
<th>Jitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Control</td>
<td>Variable size packets, mostly inelastic short messages, but traffic can also burst (BGP)</td>
<td>Low</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>Real-Time Interactive</td>
<td>Inelastic, mostly variable rate</td>
<td>Low</td>
<td>Very Low</td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td>Fixed-size small packets, inelastic</td>
<td>Very Low</td>
<td>Very Low</td>
<td>Very Low</td>
</tr>
<tr>
<td>Video</td>
<td>Fixed-size small-large packets, inelastic</td>
<td>Very Low</td>
<td>Very Low</td>
<td>Very Low</td>
</tr>
<tr>
<td>Hi-Res A/V</td>
<td>Fixed-size small-large packets, inelastic</td>
<td>Very Low</td>
<td>Very Low</td>
<td>Very Low</td>
</tr>
</tbody>
</table>
### Figure 2. Service Class Characteristics

Notes for Figure 2: A "Yes" in the jitter-tolerant column implies that received data is buffered at the endpoint and that a moderate level of server or network-induced variation in delay is not expected to affect the application. Applications that use TCP or SCTP as a transport are generally good examples. Routing protocols and peer-to-peer signaling also fall in this class; although loss can create problems in setting up calls, a moderate level of jitter merely makes call placement a little less predictable in duration.

Service classes indicate the required traffic forwarding treatment in order to meet user, application, and/or network expectations. Section 3 defines the service classes that MAY be used for forwarding network control traffic, and Section 4 defines the service classes that MAY be used for forwarding user oriented traffic with examples of intended application types mapped into each service class. Note that the application types are only examples and are not meant to be all-inclusive or prescriptive. Also, note that the service-class naming or ordering does not imply any priority ordering. They are simply reference names that are used in this document with associated QoS behaviors that are optimized for the particular application types they support. Network administrators MAY choose to assign different service class names to

<table>
<thead>
<tr>
<th>Service Class</th>
<th>Characteristics</th>
<th>Jitter-Tolerant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimedia Conferencing</td>
<td>Variable size packets, constant transmit interval, rate adaptive, reacts to loss</td>
<td>Low, Medium, Low</td>
</tr>
<tr>
<td>Multimedia Streaming</td>
<td>Variable size packets, elastic with variable rate</td>
<td>Low, Medium, High</td>
</tr>
<tr>
<td>Broadcast</td>
<td>Constant and variable rate, inelastic, non-bursty flows</td>
<td>Very, Low, Low</td>
</tr>
<tr>
<td>Low-Latency Data</td>
<td>Variable rate, bursty short-lived elastic flows</td>
<td>Low, Low, Yes</td>
</tr>
<tr>
<td>Conversational Signaling</td>
<td>Variable size packets, some what bursty short-lived flows</td>
<td>Low, Low, Yes</td>
</tr>
<tr>
<td>OAM</td>
<td>Variable size packets, elastic &amp; inelastic flows</td>
<td>Low, Medium, Yes</td>
</tr>
<tr>
<td>High-Throughput Data</td>
<td>Variable rate, bursty long-lived elastic flows</td>
<td>Low, Medium, Yes</td>
</tr>
<tr>
<td>Standard</td>
<td>A bit of everything</td>
<td>Not Specified</td>
</tr>
<tr>
<td>Low-Priority Data</td>
<td>Non-real-time and elastic</td>
<td>High, High, Yes</td>
</tr>
</tbody>
</table>

Service class characteristics indicate the required traffic forwarding treatment in order to meet user, application, and/or network expectations. Section 3 defines the service classes that MAY be used for forwarding network control traffic, and Section 4 defines the service classes that MAY be used for forwarding user oriented traffic with examples of intended application types mapped into each service class. Note that the application types are only examples and are not meant to be all-inclusive or prescriptive. Also, note that the service-class naming or ordering does not imply any priority ordering. They are simply reference names that are used in this document with associated QoS behaviors that are optimized for the particular application types they support. Network administrators MAY choose to assign different service class names to
the service classes that they will support. Figure 3 defines the RECOMMENDED relationship between service classes and DS codepoint assignment with application examples. It is RECOMMENDED that this relationship be preserved end to end.

<table>
<thead>
<tr>
<th>Service Class Name</th>
<th>DSCP Name</th>
<th>DSCP Value</th>
<th>Application Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Control</td>
<td>CS6&amp;CS7</td>
<td>11xxxx</td>
<td>Network routing</td>
</tr>
<tr>
<td>Real-Time Interactive</td>
<td>CS5, CS5-Admit</td>
<td>101000, 101001</td>
<td>Remote/Virtual Desktop and Interactive gaming</td>
</tr>
<tr>
<td>Audio</td>
<td>EF</td>
<td>101110</td>
<td>Voice bearer</td>
</tr>
<tr>
<td>Hi-Res A/V</td>
<td>CS4, CS4-Admit</td>
<td>100000, 100001</td>
<td>Conversational Hi-Res Audio/Video bearer</td>
</tr>
<tr>
<td>Video</td>
<td>AF41,AF42</td>
<td>100010,100100</td>
<td>Audio/Video conferencing</td>
</tr>
<tr>
<td>Multimedia Conferencing</td>
<td>MC, MC-Admit</td>
<td>011101, 100101</td>
<td>Presentation Data and App Sharing/Whiteboarding</td>
</tr>
<tr>
<td>Multimedia Streaming</td>
<td>AF31,AF32</td>
<td>011010,011100</td>
<td>Streaming video and audio on demand</td>
</tr>
<tr>
<td>Broadcast</td>
<td>CS3, CS3-Admit</td>
<td>011000, 011001</td>
<td>Broadcast TV, live events &amp; video surveillance</td>
</tr>
<tr>
<td>Low-Latency Data</td>
<td>AF21,AF22</td>
<td>010010,010100</td>
<td>Client/server trans., Web-based ordering, IM/Pres</td>
</tr>
<tr>
<td>Conversational Signaling</td>
<td>A/V-Sig</td>
<td>010001</td>
<td>Conversational signaling</td>
</tr>
<tr>
<td>OAM</td>
<td>CS2</td>
<td>010000</td>
<td>OAM&amp;P</td>
</tr>
<tr>
<td>High-Throughput Data</td>
<td>AF11,AF12</td>
<td>001010,001100</td>
<td>Store and forward applications</td>
</tr>
<tr>
<td>Low-Priority Data</td>
<td>CS1</td>
<td>001000</td>
<td>Any flow that has no BW assurance</td>
</tr>
<tr>
<td>Best Effort</td>
<td>CS0</td>
<td>000000</td>
<td>Undifferentiated applications</td>
</tr>
</tbody>
</table>

Figure 3. DSCP to Service Class Mapping
Notes for Figure 3:

- Default Forwarding (DF) and Class Selector 0 (CS0) (i.e., Best Effort) provide equivalent behavior and use the same DS codepoint, ‘000000’.

- RFC 2474 identifies any DSCP with a value of 11xxxxx to be for network control. This remains true, while it removes 12 DSCPs from the overall pool of 64 available DSCP values (the 4 that are x11 from this group are within pool 2 of RFC 2474, and remain as only experimentally assignable/useable).

- All PHB names that say "-Admit" are to be used only when a capacity-admission protocol is utilized for that or each traffic flow.

Changes from table 3 of RFC 4594 are as follows:

- The old term "Signaling" was using CS5 (101000), now is exclusively for the "Conversational Signaling" service group using the DSCP name of "A/V-Sig" (010001), which is newly defined in [ID-DSCP]. This is because CS5 aggregates into the 101xxx aggregate when using layer 2 technologies such as 802.3 Ethernet, 802.11 Wireless Ethernet MPLS, etc – each of which only have 3 bits to mark with. A traffic type that can have very large packets and is not delay sensitive (within reason) is not appropriate for have a 101xxx marking. A REQUIRED behavior for this PHB is that it not be starved in any node.

- "Conversational" is a new term to include all interactive audio and video. The Conversational service group consists of the audio service class, the video service class and the new Hi-Res service class.

- "Audio" obsoletes the term "Telephony", which has generally not retained the "video" aspect within the IETF, where video is still commonly called out as a separate thing. Audio retains the nonadmitted traffic PHB of EF (101110), while capacity-admitted audio has been added via the RFC 5865 defined PHB Voice-Admit.

- "Video" now is AF4x, with AF41 specifically for capacity-admitted video traffic, while AF42 and AF43 are nonadmitted video traffic.

- "Hi-Res A/V", part of the Conversational service group, is created by [ID-DSCP] for an additional business differentiation interactive video marking for higher end traffic. It is within the 100xxx as CS4 (for nonadmitted traffic) and CS4-Admit (100001) (for capacity-admitted traffic).

- "Realtime Interactive" is now using CS5 (for nonadmitted traffic), but adds a capacity-admitted DSCP CS5-Admit (101001).
"Multimedia Conferencing" is no longer using the AF4x DSCPs, rather it will use the new PHB MC (100101) (for capacity-admitted) and MC-Admit (011101) (for nonadmitted traffic).

"Multimedia Streaming" retains using AF3x, however, AF31 is now used for capacity-admitted traffic, while AF32/33 are nonadmitted.

"Broadcast" replaces "Broadcast Video" using CS3 (for nonadmitted traffic), and adds a capacity-admitted PHB CS3-Admit (011001).

It is expected that network administrators will base their choice of the service classes that they will support on their need.

Figure 4 provides a summary of DiffServ CoS mechanisms that MUST be used for the defined service classes that are further detailed in Sections 3 and 4 of this document. According to what applications/services need to be differentiated, network administrators MAY choose the service class(es) that need to be supported in their network.

<table>
<thead>
<tr>
<th>Service Class</th>
<th>DSCP</th>
<th>Conditioning at DS Edge</th>
<th>PHB Used</th>
<th>Queuing</th>
<th>AQM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Control</td>
<td>CS6/CS7</td>
<td>See Section 3.1</td>
<td>RFC2474</td>
<td>Rate</td>
<td>Yes</td>
</tr>
<tr>
<td>Real-Time Interactive</td>
<td>CS5, CS5-Admit*</td>
<td>Police using sr+bs</td>
<td>RFC2474</td>
<td>Rate</td>
<td>No</td>
</tr>
<tr>
<td>Real-Time Audio</td>
<td>EF, Voice-Admit*</td>
<td>Police using sr+bs</td>
<td>RFC3246</td>
<td>Priority</td>
<td>No</td>
</tr>
<tr>
<td>Hi-Res A/V</td>
<td>CS4, CS4-Admit*</td>
<td>Police using sr+bs</td>
<td>RFC2474</td>
<td>Priority</td>
<td>No</td>
</tr>
<tr>
<td>Video</td>
<td>AF41*, AF42, AF43</td>
<td>Using two-rate, three-color marker (such as RFC 2698)</td>
<td>RFC2597</td>
<td>Rate per DSCP</td>
<td>Yes</td>
</tr>
<tr>
<td>Multimedia Conferencing</td>
<td>MC, MC-Admit*</td>
<td>Police using sr+bs</td>
<td>[ID-DSCP]</td>
<td>Rate</td>
<td>No</td>
</tr>
<tr>
<td>Multimedia Streaming</td>
<td>AF31*, AF32, AF33 (such as RFC 2698)</td>
<td>Using two-rate, three-color marker</td>
<td>RFC2597</td>
<td>Rate per DSCP</td>
<td>Yes</td>
</tr>
<tr>
<td>Service Class</td>
<td>cos</td>
<td>PHB Description</td>
<td>RFC</td>
<td>Rate</td>
<td>Priority</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------</td>
<td>------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>Broadcast</td>
<td>CS3</td>
<td>Police using sr+bs (ID-DSCP)</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>Low-Latency Data</td>
<td>AF21</td>
<td>Using single-rate, three-color marker (such as RFC 2697)</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>AF22</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AF23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conversational Signaling</td>
<td>AV-Sig</td>
<td>Police using sr+bs (ID-DSCP)</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>OAM</td>
<td>CS2</td>
<td>Police using sr+bs (ID-DSCP)</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>High-Throughput Data</td>
<td>AF11</td>
<td>Using two-rate, three-color marker (such as RFC 2698)</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>AF12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AF13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard Data</td>
<td>DF</td>
<td>Not applicable</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Low-Priority Data</td>
<td>CS1</td>
<td>Not applicable</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>

Figure 4. Summary of CoS Mechanisms Used for Each Service Class

* denotes each DSCP identified for capacity-admission traffic only.

Notes for Figure 4:

- Conditioning at DS edge means that traffic conditioning is performed at the edge of the DiffServ network where untrusted user devices are connected to two different administrative DiffServ networks.

- "sr+bs" represents a policing mechanism that provides single rate with burst size control.

- The single-rate, three-color marker (srTCM) behavior SHOULD be equivalent to RFC 2697, and the two-rate, three-color marker (trTCM) behavior SHOULD be equivalent to RFC 2698.

- The PHB for Realtime-Interactive service class SHOULD be configured to provide high bandwidth assurance. It MAY be configured as another EF PHB (one capacity-admitted and one non-capacity-admitted, if both are to be used) that uses relaxed performance parameters and a rate scheduler.

- The PHB for Multimedia Conferencing service class SHOULD be configured to provide high bandwidth assurance. It MAY be configured as another EF PHB (one capacity-admitted and one non-capacity-admitted, if both are to be used) that uses relaxed...
performance parameters and a rate scheduler.

- The PHB for Broadcast service class SHOULD be configured to provide high bandwidth assurance. It MAY be configured as another EF PHB (one capacity-admitted and one non-capacity-admitted, if both are to be used) that uses relaxed performance parameters and a rate scheduler.

3. Network Control Traffic

Network control traffic is defined as packet flows that are essential for stable operation of an administered network, as well as the information exchanged between neighboring networks across a peering point where SLAs are in place. Network control traffic is different from user application control (signaling) that may be generated by some applications or services. Network control traffic is mostly between routers and network nodes (e.g., routing or mgmt protocols) that are used for operating, administering, controlling, or managing whole networks, network parts or just network segments. Network Control Traffic may be split into two service classes, i.e., Network Control and OAM.

3.1. Current Practice in the Internet

Based on today’s routing protocols and network control procedures that are used in the Internet, we have determined that CS6 DSCP value SHOULD be used for routing and control and that CS7 DSCP value SHOULD be reserved for future use, specifically if needed for future routing or control protocols. Network administrators MAY use a Local/Experimental DSCP, any value that contains 11xx11; therefore, they may use a locally defined service class within their network to further differentiate their routing and control traffic.

RECOMMENDED Network Edge Conditioning for CS7 DSCP marked packets:

- Drop or remark 111xxx packets at ingress to DiffServ network domain.

- 111xxx marked packets SHOULD NOT be sent across peering points. Exchange of control information across peering points SHOULD be done using CS6 DSCP and the Network Control service class.

- any internally defined 11xxx1 values, valid within that network domain, be remarked to CS6 upon egress at network peering points.

3.2. Network Control Service Class

The Network Control service class is used for transmitting packets between network devices (routers) that require control (routing) information to be exchanged between similar devices within the administrative domain, as well as across a peering point between
adjacent administrative domains. Traffic transmitted in this service class is very important as it keeps the network operational, and it needs to be forwarded in a timely manner.

The Network Control service class SHOULD be configured using the DiffServ CS6 PHB, defined in [RFC2474]. This service class MUST be configured so that the traffic receives a minimum bandwidth guarantee, to ensure that the packets always receive timely service. The configured forwarding resources for Network Control service class MUST be such that the probability of packet drop under peak load is very low. The Network Control service class SHOULD be configured to use a Rate Queuing system such as defined in Section 1.4.1.2 of this document.

The following are examples of protocols and applications that MUST use the Network Control service class if present in a network:

- Routing packet flows: OSPF, BGP, ISIS, RIP.
- Control information exchange within and between different administrative domains across a peering point where SLAs are in place.
- LSP setup using CR-LDP and RSVP-TE.

The following protocols and applications MUST NOT use the Network Control service class:

- User oriented traffic is not allowed to use this service class. By user oriented traffic, we mean packet flows that originate from user-controlled end points that are connected to the network.
  - even if originating from a server or a device acting on behalf of a user or endpoint,
  - even if it is application or in-band signaling to establish a connection wholly within a single network or across peering points of/to adjacent networks (e.g., creating a tunnel such as a VPN, or data path control signaling).

The following are traffic characteristics of packet flows in the Network Control service class:

- Mostly messages sent between routers and network servers.
- Variable size packets, normally one packet at a time, but traffic can also burst (BGP, OSPF, etc).
- IGMP, then is used only for the normal multicast routing purpose.

The REQUIRED DSCP marking is CS6 (Class Selector 6).
RECOMMENDED Network Edge Conditioning:

- At peering points (between two DiffServ networks) where SLAs are in place, CS6 marked packets MUST be policed, e.g., using a single rate with burst size (sr+bs) token bucket policer to keep the CS6 marked packet flows to within the traffic rate specified in the SLA.

- CS6 marked packet flows from untrusted sources (for example, end user devices) MUST be dropped or remarked at ingress to the DiffServ network. What a network admin remarks this user oriented traffic to if a matter of local policy, and inspection of the packets can determine which application is used for proper marking to a more appropriate DSCP, such as from table 3. of this document.

- Packets from users/subscribers are not permitted access to the Network Control service classes.

The fundamental service offered to the Network Control service class is enhanced best-effort service with high bandwidth assurance. Since this service class is used to forward both elastic and inelastic flows, the service SHOULD be engineered so that the Active Queue Management (AQM) [RFC2309] is applied to CS6 marked packets.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth, and the max-threshold specifies the queue depth above which all traffic is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- min-threshold CS6 < max-threshold CS6
- max-threshold CS6 <= memory assigned to the queue

Note: Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

3.3. OAM Service Class

The OAM (Operations, Administration, and Management) service class is RECOMMENDED for OAM&P (Operations, Administration, and Management and Provisioning) using protocols such as Simple Network Management Protocol (SNMP), Trivial File Transfer Protocol (TFTP), FTP, Telnet, and Common Open Policy Service (COPS). Applications using this service class require a low packet loss but are relatively not sensitive to delay. This service class is configured to provide good packet delivery for intermittent flows.

The OAM service class SHOULD use the Class Selector (CS) PHB defined in [RFC2474]. This service class SHOULD be configured to provide a
minimum bandwidth assurance for CS2 marked packets to ensure that they get forwarded. The OAM service class SHOULD be configured to use a Rate Queuing system such as defined in Section 1.4.1.2 of this document.

The following applications SHOULD use the OAM service class:

- Provisioning and configuration of network elements.
- Performance monitoring of network elements.
- Any network operational alarms.

The following are traffic characteristics:

- Variable size packets.
- Intermittent traffic flows.
- Traffic may burst at times.
- Both elastic and inelastic flows.
- Traffic not sensitive to delays.

**RECOMMENDED DSCP marking:**

- All flows in this service class are marked with CS2 (Class Selector 2).

Applications or IP end points SHOULD pre-mark their packets with CS2 DSCP value. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

**RECOMMENDED conditioning performed at DiffServ network edge:**

- Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods, defined in [RFC2475].
- Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the traffic stays within its negotiated or engineered bounds.
- Packet flows from trusted sources (routers inside administered network) MAY not require policing.
- Normally OAM&P CS2 marked packet flows are not allowed to flow across peering points. If that is the case, then CS2 marked
packets SHOULD be policed (dropped) at both egress and ingress peering interfaces.

The fundamental service offered to "OAM" traffic is enhanced best-effort service with controlled rate. The service SHOULD be engineered so that CS2 marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Since this service class is used to forward both elastic and inelastic flows, the service SHOULD be engineered so that Active Queue Management [RFC2309] is applied to CS2 marked packets.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth for each DSCP, and the max-threshold specifies the queue depth above which all traffic with such a DSCP is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- min-threshold CS2 < max-threshold CS2
- max-threshold CS2 <= memory assigned to the queue

Note: Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

4. User Oriented Traffic

User oriented traffic is defined as packet flows between different users or subscribers, or from servers/nodes on behalf of a user. It is the traffic that is sent to or from end-terminals and that supports a very wide variety of applications and services, to include traffic about a user or application that assists a user communicate. User oriented traffic can be classified in many different ways. What we have articulated throughout this document is a series of non-exhaustive list of categories for classifying user oriented traffic. We differentiated user oriented traffic that is real-time versus non-real-time, elastic or rate-adaptive versus inelastic, sensitive versus insensitive to loss as well as considering whether the traffic is interactive vs. one way communication, its responsiveness, whether it requires timely delivery, and critical verses non-critical. In the final analysis, we used all of the above for service differentiation, mapping application types that seemed to have different sets of performance sensitivities, and requirements to different service classes.

Network administrators can categorize their applications according to the type of behavior that they require and MAY choose to support all or a subset of the defined service classes. At the same time, we include a public facing default DSCP value, with its associated PHB, that is expected for each traffic type to ensure common or pervasive performance. Figure 3 provides some common applications and the forwarding service classes that best support them, based on their performance requirements.
4.1. Conversational Service Class Group

The Conversational Service Class Group consists of 3 different service classes, audio, video, and Hi-Res. We are describing the media sample, or bearer, packets for applications (e.g., RTP from [RFC3550]) that require bi-directional real-time, very low delay, very low jitter, and very low packet loss for relatively constant-rate traffic sources (inelastic traffic sources). It is RECOMMENDED that RTCP feedback use the same service class and be marked with the same DSCP as the bearer traffic for that (audio and/or video) call. This ensures comparable treatment within the network between endpoints.

The signaling to set-up these bearer flows is part of the Conversational Signaling service group that will be discussed later in Section 4. The following 3 subsections will detail what is expected within each bearer service class.

4.1.1 Audio Service Class

This service class MUST be used for IP Audio service.

The fundamental service offered to traffic in the Audio service class is minimum jitter, delay, and packet loss service up to a specified upper bound. There are two PHBs, both EF based, for the Audio service class:

Nonadmitted Audio traffic - MUST use the EF DSCP [RFC3246], and is for traffic that has not had any capacity admission signaling performed for that flow or session.

Capacity-Admitted Audio traffic - MUST use the Voice-Admit DSCP [RFC5865], and is for traffic that has had any capacity admission signaling performed for that flow or session, e.g., RSVP [RFC2205] or NSIS [RFC4080].

The capacity-admitted Audio traffic operation is similar to an ATM CBR service, which has guaranteed bandwidth and which, if it stays within the negotiated rate, experiences nominal delay and no loss.

The nonadmitted Audio traffic, on the other hand, has had no such explicit guarantee, but has a favorable PHB ensuring high probability of delivery as well as nominal delay and no loss - implicitly assuming there is not too much like marked traffic between users within a flow.

There are two typical scenarios in which audio calls are established, on the public open Internet using protocols such as SIP, XMPP or H.323, or in more managed networks like enterprises or certain service providers which offer a audio service with some feature benefits and take part in the call signaling. These SPs or
enterprises also use protocols like SIP, XMPP, H.323, but also use H.248/MEGACO and MGCP.

On the open Internet, typically there is no SP actively involved in the session set-up of calls, and therefore no servers providing assistance or features to help one user contact another user. Often, this traffic is marked or remarked with the DF (i.e., Best Effort) DSCP.

In more managed networks in which one of more operators have active servers aiding the audio call set-up, where DiffServ can be used and preserved to differentiate traffic, networks are offering a service, therefore need to do some, or a lot of engineering to ensure that capacity offered to one or more applications does not exceed the load to the network. Otherwise, the operator will have unhappy users, at least for that application’s usage. This is true for any application, but is especially true for inelastic applications in which the application is rigid in its delivery requirements. Audio bearer traffic is typically such an application, video is another such application, but we will get to video in the next subsection.

When a user in a managed network has been authorized to send Audio traffic (i.e., call initiation via the operator’s servers was not rejected), the call admission procedure should have verified that the newly admitted flow will be within the capacity of the Audio service class forwarding capability in the network. Capacity verification is a non-trivial thing, and can either be implicitly assumed by the call server(s) based on the operator’s network design, or it can be explicitly signaled from an in-data-path signaling mechanism that verifies the capacity is available now for this call, for each call made within that network. In the latter case, those that do not have verifiable network capacity along the data path are rejected. An in between means method is for call servers to count calls between two or more endpoints. By topologically understanding where the caller and called party is and have configured a known maximum it will allow between the two locations. This is especially true over WAN links that have far less capacity than LAN links or core parts of a network. Network operators will need to understand the topology between any two callers to ensure the appropriate amount of bandwidth is available for an expected number of simultaneous audio calls.

Once more than one bandwidth amount can be used for audio calls, for example – by allowing more than one codec with different bandwidths per codec for such calls, network engineering becomes more difficult. Since the inelastic nature of RTP payloads from this class do not react well to loss or significant delay in any substantive way, the Audio service class MUST forward packets as soon as possible.

The Audio service class that does not have capacity admission performed in the data path MUST use the Expedited Forwarding (EF)
PHB, as defined in [RFC3246], so that all packets are forwarded quickly. The Audio service class that does have capacity admission performed in the data path MUST use the Voice-Admit PHB, as defined in [RFC5865], so that all packets are forwarded quickly. The Audio service class SHOULD be configured to use a Priority Queuing system such as that defined in Section 1.4.1.1 of this document.

The following applications SHOULD use the Audio service class:

- VoIP (G.711, G.729, iLBC and other audio codecs).
- Voice-band data over IP (modem, fax).
- T.38 fax over IP.
- Circuit emulation over IP, virtual wire, etc.
- IP Virtual Private Network (VPN) service that specifies single-rate, mean network delay that is slightly longer than network propagation delay, very low jitter, and a very low packet loss.

The following are traffic characteristics:

- Mostly fixed-size packets for VoIP (30, 60, 70, 120 or 200 bytes in size).
- Packets emitted at constant time intervals.
- Admission control of new flows is provided by Audio call server, media gateway, gatekeeper, edge router, end terminal, access node or in-data-path signaling that provides flow admission control function.

Applications or IP end points SHOULD pre-mark their packets with EF or Voice-Admit DSCP value, whichever is appropriate. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

The RECOMMENDED DSCP marking is EF for nonadmitted audio flows, and Voice-Admit for capacity-admitted flows for the following applications:

- VoIP (G.711, G.729 and other codecs).
- Voice-band data over IP (modem and fax).
- T.38 fax over IP.
- Circuit emulation over IP, virtual wire, etc.
RECOMMENDED Network Edge Conditioning:

- Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods, defined in [RFC2475]. If untrusted, the network edge SHOULD know if capacity-admission has been applied, since the edge router will have taken part in the admission signaling; therefore will know whether EF or Voice-Admit is the proper marking for that flow.

- Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the Audio traffic stays within its negotiated bounds.

- Policing is OPTIONAL for packet flows from trusted sources whose behavior is ensured via other means (e.g., administrative controls on those systems).

- Policing of Audio packet flows across peering points where SLA is in place is OPTIONAL as Audio traffic will be controlled by admission control mechanism between peering points.

The fundamental service offered to "Audio" traffic is enhanced best-effort service with controlled rate, very low delay, and very low loss. The service MUST be engineered so that EF marked packet flows have sufficient bandwidth in the network to provide guaranteed delivery. Otherwise, the service will have in place an explicit capacity-admission signaling protocol such as RSVP or NSIS and thus mark the packets within the flow as Voice-Admit. Normally traffic in this service class does not respond dynamically to packet loss. As such, Active Queue Management [RFC2309] SHOULD NOT be applied to EF marked packet flows.

4.1.2 Video Service Class

The Video service class is for bidirectional applications that require real-time service for both constant and rate-adaptive traffic. SIP and H.323/V2 (and later) versions of video conferencing equipment with constant and dynamic bandwidth adjustment are such applications. The traffic sources in this service class either have a fixed bandwidth requirement (e.g., MPEG2, etc.), or have the ability to dynamically change their transmission rate (e.g., MPEG4/H.264, etc.) based on feedback from the receiver. This feedback SHOULD be accomplished using RTCP [RFC3550]. One approach for this downspeeding has the receiver detect packet loss, thus signaling in an RTCP message to the source the indication of lost (or delayed or out of order) packets in transit. When necessary the source then selects a lower rate encoding codec. When available, the source merely sends less data, resulting in lower resolution of the same visual display.
The Video service class is not for video downloads, webcasts, or single directional video or audio/video traffic of any kind. It is for human-to-human visual interaction between two users, or more if an MTP is used.

Typical video conferencing configurations negotiate the setup of audio/video session using protocols such as SIP and H.323. Just as with networks that have audio traversing them, video typically traverses the same two types of networks: the open big "I" Internet, in which most every type of traffic is best effort (DF), or on a more managed network such as an enterprise or SP’s managed network in which servers within either network take part in the call signaling, thereby offering the video service.

When a user in a managed network has been authorized to send video traffic (i.e., call initiation via the operator’s servers was not rejected), the call admission procedure should have verified that the newly admitted flow will be within the capacity of the video service class forwarding capability in the network. Capacity verification is a non-trivial thing, and can either be implicitly assumed by the call server(s) based on the operator’s network design, or it can be explicitly signaled from an in-data-path signaling mechanism that verifies the capacity is available now for this call, for each call made within that network. In the latter case, those that do not have verifiable network capacity along the data path are rejected. An in between means method is for call servers to count calls between two or more endpoints. By topologically understanding where the caller and called party is and have configured a known maximum it will allow between the two locations. Video is larger in bandwidth than audio, and the difference can be significant. For example, for a single G.711 audio call that is 80kbps, an associated video bandwidth for the same call can easily be 4Mbps. This is especially true over WAN links that have far less capacity than LAN links or core parts of a network. Network operators will need to understand the topology between any two callers to ensure the appropriate amount of bandwidth is available for an expected number of simultaneous video and/or audio/video calls.

Note that it is OPTIONALLY the case in these networks that the accompanying audio for the video call will be marked as the video is marked (i.e., using the same DSCP), but not always. One reason this has been done is for lip-sync.

The Video service class MUST use the Assured Forwarding (AF) PHB, defined in [RFC2597]. This service class MUST be configured to provide a bandwidth assurance for AF41, AF42, and AF43 marked packets to ensure that they get forwarded. The Video service class SHOULD be configured to use a Rate Queuing system for AF42 and AF43 traffic flows, such as that defined in Section 1.4.1.2 of this document. However, AF41 MUST be designated as the DSCP for use when
capacity-admission signaling has been used, such as RSVP or NSIS, to guarantee delivery through the network. AF42 and AF43 will be used for non-admitted video calls, as well as overflows from AF41 sources that send more packets than they have negotiated bandwidth for that call.

The following applications MUST use the Video service class:

- SIP and H.323/V2 (and later) versions of video conferencing applications (interactive video).
- Video conferencing applications with rate control or traffic content importance marking.
- Interactive, time-critical, and mission-critical applications.

NOTE with regards to the above bullet: this usage SHOULD be minimized, else the video traffic will suffer – unless this is engineered into the topology.

The following are traffic characteristics:

- Variable size packets (i.e., small to large in size).
- The higher the resolution or change rate between each image, the higher the duration of large packets.
- Usually constant inter-packet time interval.
- Can be Variable rate in transmission.
- Source is capable of reducing its transmission rate based on being told receiver is detecting packet loss (e.g., via RTCP).

Applications or IP end points SHOULD pre-mark their packets with DSCP values as shown below. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475] and mark all packets as AF4x. Note: In this case, the two-rate, three-color marker will be configured to operate in Color-Blind mode.

Mandatory DSCP marking when performed by router closest to source:

- AF41 = up to specified rate "A", which is dedicated to non-Hi-Res capacity-admitted video traffic.
  
  Note the audio of an A/V call can be marked AF41 as well.

- AF42 = all non-Hi-Res video traffic marked AF41 in excess of specified rate "A", or new non-admitted video traffic but below specified rate "B".
o AF43 = in excess of specified rate "B".
  o Where "A" < "B".

Note: One might expect "A" to approximate the peak rates of sum of all admitted video flows, plus the sum of the mean rates and "B" to approximate the sum of the peak rates of those same two flows.

Mandatory DSCP marking when performed by SIP or H.323/V2 videoconferencing equipment:

o AF41 = SIP or H.323 video conferencing audio stream RTP.

o AF41 = SIP or H.323 video conferencing video control RTCP.

o AF41 = SIP or H.323 video conferencing video stream up to specified rate "A".

o AF42 = SIP or H.323 video conferencing video stream in excess of specified rate "A" but below specified rate "B".

o AF42 = SIP or H.323 video conferencing video control RTCP, for those video streams that were generated using AF42.

o AF43 = SIP or H.323 video conferencing video stream in excess of specified rate "B".

o AF43 = SIP or H.323 video conferencing video control RTCP, for those video streams that were generated using AF43.

  o Where "A" < "B".

Mandatory conditioning performed at DiffServ network edge:

  o The two-rate, three-color marker SHOULD be configured to provide the behavior as defined in trTCM [RFC2698].

  o If packets are marked by trusted sources or a previously trusted DiffServ domain and the color marking is to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Aware mode.

  o If the packet marking is not trusted or the color marking is not to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Blind mode.

The fundamental service offered to nonadmitted "Video" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Video" traffic is a guaranteed service using in-data-path signaling to ensure expected
delivery in a timely manner. For a non-admitted video conferencing service, if a 1% packet loss detected at the receiver triggers an encoding rate change, thus dropping to the next lower provisioned video encoding rate then Active Queue Management [RFC2309] SHOULD be used primarily to switch the video encoding rate under congestion, changing from high rate to lower rate, i.e., 1472 kbps to 768 kbps. This rule applies to all AF42 and 43 flows. The probability of loss of AF41 traffic MUST NOT exceed the probability of loss of AF42 traffic, which in turn MUST NOT exceed the probability of loss of AF43 traffic.

Capacity-admitted video service should not result in packet loss. However, administratively this MAY be allowed to cause a purposeful downspeeding event (i.e., a change in resolution or a change in codec) to occur due to congestion.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth for each DSCP, and the max-threshold specifies the queue depth above which all traffic with such a DSCP is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

1. min-threshold AF43 < max-threshold AF43
2. max-threshold AF43 <= min-threshold AF42
3. min-threshold AF42 < max-threshold AF42
4. max-threshold AF42 <= min-threshold AF41
5. min-threshold AF41 < max-threshold AF41
6. max-threshold AF41 <= memory assigned to the queue

Note: This configuration tends to drop AF43 traffic before AF42 and AF42 before AF41. Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

4.1.3 Hi-Res Service Class

The Hi-Res service class is for higher end (i.e., deemed ‘more important’) bidirectional applications that require real-time service for both constant and rate-adaptive traffic. There are two PHBs, both EF based, for the Hi-Res video conferencing service class:

Nonadmitted Hi-Res traffic – MUST use the CS4 DSCP [RFC2474], and is for traffic that has not had any capacity admission signaling performed for that flow or session.

Capacity-Admitted Hi-Res traffic – MUST use the CS4-Admit DSCP
The capacity-admitted Hi-Res video conferencing traffic operation is similar to an ATM CBR service, which has guaranteed bandwidth and which, if it stays within the negotiated rate, experiences nominal delay and no loss.

SIP and H.323/V2 (and later) versions of video conferencing equipment with constant and dynamic bandwidth adjustment are such applications. The traffic sources in this service class either have a fixed bandwidth requirement (e.g., MPEG2), or have the ability to dynamically change their transmission rate (e.g., MPEG4/H.264) based on feedback from the receiver. This feedback SHOULD be accomplished using RTCP [RFC3550]. One approach for this downspeeding has the receiver detect packet loss, thus signaling in an RTCP message to the source the indication of lost (or delayed or out of order) packets in transit. When necessary the source then selects a lower rate encoding codec. When available, the source merely sends less data, resulting in lower resolution of the same visual display.

The Hi-Res service class, as with the Video service class, is not for video downloads, webcasts, or single directional video or audio/video traffic of any kind. It is for human-to-human visual interaction between two users, or more if an MTP is used.

Typical Hi-Res video conferencing configurations negotiate the setup of audio/video session using protocols such as SIP and H.323. Hi-Res video conferencing is generally not over the big "I" Internet, rather nearly exclusively over more managed networks such as an enterprise or special purpose SP’s managed network in which servers within either network take part in the call signaling, thereby offering the video service. In addition, typically this type of audio/video service has high business expectations for minimized packet loss, pixilation or other issues with the audio/video experience. In the recent past, entire T3s have been dedicated to a signal Hi-Res call; sometimes one T3 per site of a multi-site video conference.

Hi-Res video conferencing often has larger in bandwidth than the typical video call. The audio portion can be increased as well, as stereo capabilities are often necessary to provide an in-room experience from a distance. The difference can be significant (or another step up from just a typical video service). For example, for a single G.711 audio call that is 80kbps, a Hi-Res conference usually runs G.722 wideband audio at 256kbps. Typical video delivery is up to 4Mbps, whereas a Hi-Res conference can have three 1080p/30fps widescreen displays requiring at least 12Mbps, with a burst capability of much more.

If there were no congestion on the wire, the expected treatment
between a video service and a Hi−Res conference would be the same. However, it is typically the case that the Hi−Res conferencing flows have more rigid requirements for quality and business-wise, need to be experience far less errors than the regular video service on the same network.

Note that it is likely the case in these networks that the accompanying audio to the Hi−Res video call will be marked as the Hi−Res video is marked (i.e., using the same DSCP).

The Hi−Res service class MUST use the Class Selector 5 (CS4) PHB, defined in [RFC2474], for non-capacity-admitted conferences. While the capacity-admitted Hi−Res conferences MUST use the CS4−Admit PHB, defined in [ID-DSCP]. This service class MUST be configured to provide a bandwidth assurance for CS4 and CS4−Admit marked packets to ensure that they get forwarded. The Hi−Res service class SHOULD be configured to use a Priority Queuing system such as that defined in Section 1.4.1.1 of this document. Further, CS4−Admit will be designated as the DSCP for use when capacity-admission signaling has been used, such as RSVP or NSIS, to guarantee delivery through the network. CS4 will be used for non-admitted Hi−Res conferences, as well as overflows from CS4−Admit sources that send more packets than they have negotiated bandwidth for that call.

The following applications MUST use the Hi−Res service class:

- SIP and H.323/V2 (and later) versions of Hi−Res video conferencing applications (interactive Hi−Res video).
- Video conferencing applications with rate control or traffic content importance marking.

The following are traffic characteristics:

- Variable size packets.
- The higher the resolution or change rate between each image, the higher the duration of large packets.
- Usually constant inter-packet time interval.
- Can be Variable rate in transmission.
- Source is capable of reducing its transmission rate based on being told receiver is detecting packet loss.

Applications or IP end points SHOULD pre-mark their packets with DSCP values as shown below. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475] and mark all packets as AF4x.
Mandatory DSCP marking when performed by router closest to source:

- CS4-Admit = up to specified rate "A", which is dedicated to capacity-admitted Hi-Res traffic.
  
  Note the audio of an A/V call can be marked CS4-Admit as well.

- CS4 = all video traffic marked CS4-Admit in excess of specified rate "A", or new non-admitted video traffic but below specified rate "B".
  
  Where "A" < "B".

Note: One might expect "A" to approximate the peak rates of sum of all admitted video flows, plus the sum of the mean rates and "B" to approximate the sum of the peak rates of those same two flows.

Mandatory DSCP marking when performed by SIP or H.323/V2 videoconferencing equipment:

- CS4-Admit = SIP or H.323 video conferencing audio stream RTP/UDP.
  
- CS4-Admit = SIP or H.323 video conferencing video control RTCP/TCP.
  
- CS4-Admit = SIP or H.323 video conferencing video stream up to specified rate "A".
  
- CS4 = SIP or H.323 video conferencing video stream in excess of specified rate "A" but below specified rate "B".
  
  Where "A" < "B".

Mandatory conditioning performed at DiffServ network edge:

- The two-rate, three-color marker SHOULD be configured to provide the behavior as defined in trTCM [RFC2698].
  
- If packets are marked by trusted sources or a previously trusted DiffServ domain and the color marking is to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Aware mode.
  
- If the packet marking is not trusted or the color marking is not to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Blind mode.

The fundamental service offered to nonadmitted "Hi-Res" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Hi-Res" traffic is a guaranteed service using in-data-path signaling to ensure expected
or timely delivery. Capacity-admitted video service SHOULD NOT result in packet loss. However, administratively this MAY be allowed to cause a purposeful downspeeding event (i.e., a change in resolution or a change in codec) to occur.

4.2. Realtime-Interactive Service Class

The Realtime-Interactive service class is for bidirectional applications that require low loss and jitter and very low delay for constant or variable rate inelastic traffic sources. Interactive gaming applications that do not have the ability to change encoding rates or to mark packets with different importance indications is one good example of such an application. Another set of applications is virtualized desktop applications in which a remote user has a keyboard, mouse and display monitor, but the desktop is virtualized with the memory/processor/applications back in a common data center, requiring near instantaneous feedback on the user’s monitor of any changes caused by the application or an action by the user. Rich media protocols for voice and video MUST NOT use the Realtime-Interactive service class, but rather the appropriate service class from the Conversational service group discussed early in Section 4.1.

The Realtime-Interactive service class will use two PHBs:

Nonadmitted Realtime-Interactive traffic − MUST use the CS5 DSCP [RFC2474], and is for traffic that has not had any capacity admission signaling performed for that flow or session.

Capacity-Admitted Realtime-Interactive traffic − MUST use the CS5-Admit DSCP [ID-DSCP], and is for traffic that has had any capacity admission signaling performed for that flow or session, e.g., RSVP [RFC2205] or NSIS [RFC4080].

The capacity-admitted Realtime-Interactive traffic operation is similar to an ATM CBR service, which has guaranteed bandwidth and which, if it stays within the negotiated rate, experiences nominal delay and no loss.

Either of the above service classes can be configured as EF based by using a relaxed performance parameter and a rate scheduler.

When a user/endpoint has been authorized to start a new session (i.e., joins a networked game or logs onto a virtualized workstation), the admission procedure should have verified that the newly admitted data rates will be within the engineered capacity of the Realtime-Interactive service class. The bandwidth in the core network and the number of simultaneous Realtime-Interactive sessions that can be supported SHOULD be engineered to control traffic load for this service.
This service class SHOULD be configured to provide a high assurance for bandwidth for CS5 PHB, defined in [RFC2474], or CS5-Admit [ID-DSCP] for guaranteed service through a capacity-admission signaling protocol. The Realtime-Interactive service class SHOULD be configured to use a Rate Queuing system such as that defined in Section 1.4.1.2 of this document. Note that either Realtime-Interactive PHB MAY be configured as another EF PHB, specifically CS5-Admit, that uses a relaxed performance parameter and a rate scheduler, in the priority queue as defined in Section 1.4.1.1 of this document.

The following applications MUST use the Realtime-Interactive service class:

- Interactive gaming and control.
- Remote Desktop applications
- Virtualized Desktop applications.
- Application server-to-application server non-bursty data transfer requiring very low delay.
- Inelastic, interactive, time-critical, and mission-critical applications requiring very low delay.

The following are traffic characteristics:

- Variable size packets.
- Variable rate, though sometimes bursty, which will require engineering of the network to accommodate.
- Application is sensitive to delay variation between flows and sessions.
- Lost packets, if any, are usually ignored by application.

RECOMMENDED DSCP marking:

- All non-admitted flows in this service class are marked with CS5 (Class Selector 5).
- All capacity-admitted flows in this service class are marked with CS5-Admit.

Applications or IP end points SHOULD pre-mark their packets with CS5 or CS5-Admit DSCP value, depending on whether a capacity-admission signaling protocol is used for a flow. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].
RECOMMENDED conditioning performed at DiffServ network edge:

- Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods defined in [RFC2475].

- Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the traffic stays within its negotiated or engineered bounds.

- Packet flows from trusted sources (application servers inside administered network) MAY not require policing.

- Policing of packet flows across peering points MUST adhere to the Service Level Agreement (SLA).

The fundamental service offered to nonadmitted "Realtime-Interactive" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Realtime-Interactive" traffic is a guaranteed service using in-data-path signaling to ensure expected or timely delivery. Capacity-admitted Realtime-Interactive service SHOULD NOT result in packet loss. The service SHOULD be engineered so that CS5 marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Normally, traffic in this service class does not respond dynamically to packet loss. As such, Active Queue Management [RFC2309] SHOULD NOT be applied to CS5 marked packet flows.

4.3. Multimedia Conferencing Service Class

The Multimedia Conferencing service class is for applications that have a low to medium tolerance to delay, and are rate adaptive to lost packets in transit from sources. Presentation Data applications that are operational in conjunction with an audio/video conference is one good example of such an application. Another set of applications is application sharing or whiteboarding applications, also in conjunction to an A/V conference. In either case, the audio & video part of the flow MUST NOT use the Multimedia Conferencing service class, rather the more appropriate service class within the Conversational service group discussed earlier in Section 4.1.

The Multimedia Conferencing service class will use two PHBs:

- Nonadmitted Multimedia Conferencing traffic - MUST use the (new) MC DSCP [ID-DSCP], and is for traffic that has not had any capacity admission signaling performed for that flow or
Capacity-Admitted Multimedia Conferencing traffic - MUST use the (new) MC-Admit DSCP [ID-DSCP], and is for traffic that has had any capacity admission signaling performed for that flow or session, e.g., RSVP [RFC2205] or NSIS [RFC4080].

The capacity-admitted Multimedia Conferencing traffic operation is similar to an ATM CBR service, which has guaranteed bandwidth and which, if it stays within the negotiated rate, experiences nominal delay and no loss.

When a user/endpoint initiates a presentation data, application sharing or whiteboarding session, it will typically be part of an audio or audio/video conference such as web-conferencing or an existing Telepresence call. The authorization procedure SHOULD be controlled through the coordinated effort to bind the A/V call with the correct Multimedia Conferencing packet flow through some use of identifiers not in scope of this document. The managed network this flow traverse and the number of simultaneous Multimedia Conferencing sessions that can be supported SHOULD be engineered to control traffic load for this service.

The non-capacity admitted Multimedia Conferencing service class SHOULD use the new MC PHB, defined in [ID-DSCP]. This service class SHOULD be configured to provide a high assurance for bandwidth for CS5 marked packets to ensure that they get forwarded. The Multimedia Conferencing service class SHOULD be configured to use a Rate Queuing system such as that defined in Section 1.4.1.2 of this document. Note that this service class MAY be configured as another EF PHB that uses a relaxed performance parameter, a rate scheduler, and MC-Admit DSCP value, which MUST use the priority queue as defined in Section 1.4.1.1 of this document.

The following applications MUST use the Multimedia Conferencing service class:

- Presentation Data applications, which can utilize vector graphics, raster graphics or video delivery.
- Virtualized Desktop applications.
- Application server-to-application server non-bursty data transfer requiring very low delay.

The following are traffic characteristics:

- Variable size packets.
- Variable rate, though sometimes bursty, which will require engineering of the network to accommodate.
RECOMMENDED DSCP marking:

- All non-admitted flows in this service class are marked with the new MC DSCP.
- All capacity-admitted flows in this service class are marked with MC-Admit.

Applications or IP end points SHOULD pre-mark their packets with the MC DSCP value. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

RECOMMENDED conditioning performed at DiffServ network edge:

- Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods defined in [RFC2475].
- Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the traffic stays within its negotiated or engineered bounds.
- Packet flows from trusted sources (application servers inside administered network) MAY not require policing.
- Policing of packet flows across peering points MUST adhere to the Service Level Agreement (SLA).

The fundamental service offered to nonadmitted "Multimedia Conferencing" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Multimedia Conferencing" traffic is a guaranteed service using in-data-path signaling to ensure expected or timely delivery. Capacity-admitted Multimedia Conferencing service SHOULD NOT result in packet loss. The service SHOULD be engineered so that Multimedia Conferencing marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery.

Normally, traffic in this service class does not respond dynamically to packet loss. As such, Active Queue Management [RFC2309] SHOULD NOT be applied to MC or MC-Admit marked packet flows.

4.4. Multimedia Streaming Service Class
The Multimedia Streaming service class is RECOMMENDED for applications that require near-real-time packet forwarding of variable rate elastic traffic sources that are not as delay sensitive as applications using the Broadcast service class. Such applications include streaming audio and video, some video (movies) on-demand applications, and non-interactive webcasts. In general, the Multimedia Streaming service class assumes that the traffic is buffered at the source/destination; therefore, it is less sensitive to delay and jitter.

The Multimedia Streaming service class MUST use the Assured Forwarding (AF3x) PHB, defined in [RFC2597]. This service class MUST be configured to provide a minimum bandwidth assurance for AF31, AF32, and AF33 marked packets to ensure that they get forwarded. The Multimedia Streaming service class SHOULD be configured to use Rate Queuing system for AF32 and AF33 traffic flows, such as that defined in Section 1.4.1.2 of this document. However, AF31 MUST be designated as the DSCP for use when capacity-admission signaling has been used, such as RSVP or NSIS, to guarantee delivery through the network. AF32 and AF33 will be used for non-admitted streaming flows, as well as overflows from AF31 sources that send more packets than they have negotiated bandwidth for that call.

The following applications SHOULD use the Multimedia Streaming service class:

- Buffered streaming audio (unicast).
- Buffered streaming video (unicast).
- Non-interactive Webcasts.
- IP VPN service that specifies two rates and is less sensitive to delay and jitter.

The following are traffic characteristics:

- Variable size packets.
- The higher the rate, the higher the density of large packets.
- Variable rate.
- Elastic flows.
- Some bursting at start of flow from some applications, as well as an expected stepping up and down on the rate of the flow based on changes in resolution due to network conditions.

Applications or IP end points SHOULD pre-mark their packets with DSCP values as shown below. If the end point is not capable of
setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475], and mark all packets as AF3x. Note: In this case, the two-rate, three-color marker will be configured to operate in Color-Blind mode.

RECOMMENDED DSCP marking:

- AF31 = up to specified rate "A".
- AF32 = all traffic marked AF31 in excess of specified rate "A", or new AF32 traffic but below specified rate "B".
- AF33 = in excess of specified rate "B".
- Where "A" < "B".

Note: One might expect "A" to approximate the peak rates of sum of all streaming flows, plus the sum of the mean rates and "B" to approximate the sum of the peak rates of those same two flows.

RECOMMENDED conditioning performed at DiffServ network edge:

- The two-rate, three-color marker SHOULD be configured to provide the behavior as defined in trTCM [RFC2698].
- If packets are marked by trusted sources or a previously trusted DiffServ domain and the color marking is to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Aware mode.
- If the packet marking is not trusted or the color marking is not to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Blind mode.

The fundamental service offered to nonadmitted "Multimedia Streaming" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Multimedia Streaming" traffic is a guaranteed service using in-data-path signaling to ensure expected delivery in a reasonable manner. The service SHOULD be engineered so that AF31 marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Since the AF3x traffic is elastic and responds dynamically to packet loss, Active Queue Management [RFC2309] SHOULD be used primarily to reduce forwarding rate to the minimum assured rate at congestion points, unless AF31 has had a capacity-admission signaling protocol applied to the flow, such as RSVP or NSIS.

If a capacity-admission signaling protocol applied to the AF31 flow, which SHOULD be the case always, the AF31 PHB MAY be configured as another EF PHB that uses a relaxed performance parameter and a rate...
scheduler, in the priority queue as defined in Section 1.4.1.1 of this document.

The probability of loss of AF31 traffic MUST NOT exceed the probability of loss of AF32 traffic, which in turn MUST NOT exceed the probability of loss of AF33.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth for each DSCP, and the max-threshold specifies the queue depth above which all traffic with such a DSCP is dropped or ECN marked. Thus, in this service class, the following inequality MUST hold in queue configurations:

- \( \text{min-threshold AF33} < \text{max-threshold AF33} \)
- \( \text{max-threshold AF33} \leq \text{min-threshold AF32} \)
- \( \text{min-threshold AF32} < \text{max-threshold AF32} \)
- \( \text{max-threshold AF32} \leq \text{min-threshold AF31} \)
- \( \text{min-threshold AF31} < \text{max-threshold AF31} \)
- \( \text{max-threshold AF31} \leq \text{memory assigned to the queue} \)

Note#1: this confirmation MUST be modified if AF31 has a capacity-admission signaling protocol applied to those flows, and the above will only apply to AF32 and AF33, while AF31 (theoretically) has no packet loss.

Note#2: This configuration tends to drop AF33 traffic before AF32 and AF32 before AF31. Note: Many other AQM algorithms exist and are used; they SHOULD be configured to achieve a similar result.

4.5. Broadcast Service Class

The Broadcast service class is RECOMMENDED for applications that require near-real-time packet forwarding with very low packet loss of constant rate and variable rate inelastic traffic sources that are more delay sensitive than applications using the Multimedia Streaming service class. Such applications include broadcast TV, streaming of live audio and video events, some video-on-demand applications, and video surveillance. In general, the Broadcast service class assumes that the destination end point has a dejitter buffer, for video application usually a 2 - 8 video-frame buffer (66 to several hundred of milliseconds), thus expecting far less buffering before play-out than Multimedia Streaming, which can buffer in the seconds to minutes (to hours).

The Broadcast service class will use two PHBs:
Nonadmitted Broadcast traffic - MUST use the CS3 DSCP [RFC2474], and is for traffic that has not had any capacity admission signaling performed for that flow or session.

Capacity-Admitted Broadcast traffic - MUST use the CS3-Admit DSCP [ID-DSCP], and is for traffic that has had any capacity admission signaling performed for that flow or session, e.g., RSVP [RFC2205] or NSIS [RFC4080].

The capacity-admitted Broadcast traffic operation is similar to an ATM CBR service, which has guaranteed bandwidth and which, if it stays within the negotiated rate, experiences nominal delay and no loss.

Either of the above service classes can be configured as EF based by using a relaxed performance parameter and a rate scheduler.

When a user/endpoint initiates a new Broadcast session (i.e., starts an Internet radio application, starts a live Internet A/V event or a camera comes online to do video-surveillance), the admission procedure should be verified within the application that triggers the flow. The newly admitted data rates will SHOULD be within the engineered capacity of the Broadcast service class within that network. The bandwidth in the core network and the number of simultaneous Broadcast sessions that can be supported SHOULD be engineered to control traffic load for this service.

This service class SHOULD be configured to provide high assurance for bandwidth for CS3 marked packets to ensure that they get forwarded. The Broadcast service class SHOULD be configured to use Rate Queuing system such as that defined in Section 1.4.1.2 of this document. Note that either Broadcast PHB MAY be configured as another EF PHB, specifically CS3-Admit, that uses a relaxed performance parameter and a rate scheduler, in the priority queue as defined in Section 1.4.1.1 of this document.

The following applications SHOULD use the Broadcast service class:

- Video surveillance and security (unicast).
- TV broadcast including HDTV (likely multicast, but can be unicast).
- Video on demand (unicast) with control (virtual DVD).
- Streaming of live audio events (both unicast and multicast).
- Streaming of live video events (both unicast and multicast).

The following are traffic characteristics:
Internet-Draft  Std Config. for DiffServ Service Classes     March 2012

- Variable size packets.
- The higher the rate, the higher the density of large packets.
- Mixture of variable rate and constant rate flows.
- Fixed packet emission time intervals.
- Inelastic flows.

RECOMMENDED DSCP marking:

- All non-admitted flows in this service class are marked with CS3 (Class Selector 3).
- All capacity-admitted flows in this service class are marked with CS3-Admit.
- In some cases, such as those for security and video surveillance applications, it is NOT RECOMMENDED, but allowed to use a different DSCP marking.

If so, then locally user definable (EXP/LU) codepoints in the range ‘011x11’ MAY be used to provide unique traffic identification. The locally administrator definable (EXP/LU, from pool 2 of RFC 2474) codepoint(s) MAY be associated with the PHB that is used for CS3 or CS3-Admit traffic. Furthermore, depending on the network scenario, additional network edge conditioning policy MAY be needed for the EXP/LU codepoint(s) used.

Applications or IP end points SHOULD pre-mark their packets with CS3 or CS3-Admit DSCP value. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

RECOMMENDED conditioning performed at DiffServ network edge:

- Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods defined in [RFC2475].
- Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the traffic stays within its negotiated or engineered bounds.
- Packet flows from trusted sources (application servers inside administered network) MAY not require policing.
Policing of packet flows across peering points MUST be performed to the Service Level Agreement (SLA) of those peering entities.

The fundamental service offered to "Broadcast" traffic is enhanced best-effort service with controlled rate and delay. The fundamental service offered to capacity-admitted "Broadcast" traffic is a guaranteed service using in-data-path signaling to ensure expected or timely delivery. Capacity-admitted Broadcast service SHOULD NOT result in packet loss. The service SHOULD be engineered so that CS3 and CS3-Admit marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Normally, traffic in this service class does not respond dynamically to packet loss. As such, Active Queue Management [RFC2309] SHOULD NOT be applied to CS3 marked packet flows.

4.6. Low-Latency Data Service Class

The Low-Latency Data service class is RECOMMENDED for elastic and responsive typically client-/server-based applications. Applications forwarded by this service class are those that require a relatively fast response and typically have asymmetrical bandwidth need, i.e., the client typically sends a short message to the server and the server responds with a much larger data flow back to the client. The most common example of this is when a user clicks a hyperlink (~ few dozen bytes) on a web page, resulting in a new web page to be loaded (Kbytes or MBs of data). This service class is configured to provide good response for TCP [RFC1633] short-lived flows that require real-time packet forwarding of variable rate traffic sources.

The Low-Latency Data service class SHOULD use the Assured Forwarding (AF) PHB, defined in [RFC2597]. This service class SHOULD be configured to provide a minimum bandwidth assurance for AF21, AF22, and AF23 marked packets to ensure that they get forwarded. The Low-Latency Data service class SHOULD be configured to use a Rate Queuing system such as that defined in Section 1.4.1.2 of this document.

The following applications SHOULD use the Low-Latency Data service class:

- Client/server applications.
- Systems Network Architecture (SNA) terminal to host transactions (SNA over IP using Data Link Switching (DLSw)).
- Web-based transactions (E-commerce).
- Credit card transactions.
- Financial wire transfers.
Enterprise Resource Planning (ERP) applications (e.g., SAP/Baan).

- VPN service that supports Committed Information Rate (CIR) with up to two burst sizes.

- Instant Messaging and Presence protocols (e.g., SIP, XMPP).

The following are traffic characteristics:

- Variable size packets.
- Variable packet emission rate.
- With packet bursts of TCP window size.
- Short traffic bursts.
- Source capable of reducing its transmission rate based on detection of packet loss at the receiver or through explicit congestion notification.

Applications or IP end points SHOULD pre-mark their packets with DSCP values as shown below. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475] and mark all packets as AF2x. Note: In this case, the single-rate, three-color marker will be configured to operate in Color-Blind mode.

**RECOMMENDED DSCP marking:**

- AF21 = flow stream with packet burst size up to "A" bytes.
- AF22 = flow stream with packet burst size in excess of "A" but below "B" bytes.
- AF23 = flow stream with packet burst size in excess of "B" bytes.  
  - Where "A" < "B".

**RECOMMENDED conditioning performed at DiffServ network edge:**

- The single-rate, three-color marker SHOULD be configured to provide the behavior as defined in srTCM [RFC2697].

- If packets are marked by trusted sources or a previously trusted DiffServ domain and the color marking is to be preserved, then the single-rate, three-color marker SHOULD be configured to operate in Color-Aware mode.

- If the packet marking is not trusted or the color marking is not to be preserved, then the single-rate, three-color marker
SHOULD be configured to operate in Color-Blind mode.

The fundamental service offered to "Low-Latency Data" traffic is enhanced best-effort service with controlled rate and delay. The service SHOULD be engineered so that AF21 marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery. Since the AF2x traffic is elastic and responds dynamically to packet loss, Active Queue Management [RFC2309] SHOULD be used primarily to control TCP flow rates at congestion points by dropping packets from TCP flows that have large burst size. The probability of loss of AF21 traffic MUST NOT exceed the probability of loss of AF22 traffic, which in turn MUST NOT exceed the probability of loss of AF23. Explicit Congestion Notification (ECN) [RFC3168] MAY also be used with Active Queue Management.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth for each DSCP, and the max-threshold specifies the queue depth above which all traffic with such a DSCP is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- o min-threshold AF23 < max-threshold AF23
- o max-threshold AF23 <= min-threshold AF22
- o min-threshold AF22 < max-threshold AF22
- o max-threshold AF22 <= min-threshold AF21
- o min-threshold AF21 < max-threshold AF21
- o max-threshold AF21 <= memory assigned to the queue

Note: This configuration tends to drop AF23 traffic before AF22 and AF22 before AF21. Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

4.7. Conversational Signaling Service Class

The Signaling service class is MUST be limited to delay-sensitive signaling traffic only, and then only applying to signaling that involves the Conversational service group. Audio signaling includes signaling between IP phone and soft-switch, soft-client and soft-switch, and media gateway and soft-switch as well as peer-to-peer using various protocols. Video and Hi-Res signaling includes video endpoint to video endpoint, as well as to Media transfer Point (MTP), to call control server(S), etc. This service class is intended to be used for control of voice and video sessions and applications. Protocols using this service class require a relatively fast response, as there are typically several messages of different sizes sent for control of the session. This service class is configured to provide good response for short-lived, intermittent
flows that require real-time packet forwarding. This is not the
service class for Instant Messaging (IM), that’s within the bounds
of the Low-Latency Data service class. The Conversational Signaling
service class MUST be configured so that the probability of packet
drop or significant queuing delay under peak load is very low in IP
network segments that provide this interface.

The Conversational Signaling service class MUST use the new A/V-Sig
PHB, defined in [ID-DSCP]. This service class MUST be configured to
provide a minimum bandwidth assurance for A/V-Sig marked packets to
ensure that they get forwarded. In other words, this service class
MUST NOT be starved from transmission within a reasonable timeframe,
given that the entire Conversational service group depends on these
signaling messages successful delivery. Network engineering SHOULD
be done to ensure there is roughly 1-4% available per node interface
that audio and video traverse. Local conditions MUST be considered
when determining exactly how much bandwidth is given to this service
class. The Conversational Signaling service class SHOULD be
configured to use a Rate Queuing system such as that defined in
Section 1.4.1.2 of this document.

The following applications SHOULD use the Conversational Signaling
service class:

- Peer-to-peer conversational signaling (e.g., SIP, H.323, XMPP).
- Peer-to-peer signaling for multimedia applications (e.g., SIP, H.323, XMPP).
- Peer-to-peer real-time control function.
- Client-server conversational signaling using H.248, MEGACO, MGCP,
  IP encapsulated ISDN, or other proprietary protocols.
- Signaling to control IPTV applications using protocols such as
  IGMP.
- Signaling flows between high-capacity voice call servers or
  soft switches using protocol such as SIP-T. Such high-capacity
devices may control thousands of voice (VoIP) calls.
- Signaling for one-way video flows, such as RTSP [RFC2326].
- IGMP, when used for multicast session control such as channel
  changing in IPTV systems.
- OPTIONALLY, this service class can be used for on-path
  reservation signaling for the traffic flows that will use the
  "admitted" DSCPs. The alternative is to have the on-path
  signaling (for reservations) use the DSCP within that service
class. This provides a similar treatment of the signaling to the
data flow, which might be desired.
The following are traffic characteristics:

- Variable size packets, normally one packet at a time.
- Intermittent traffic flows.
- Traffic may burst at times.
- Delay-sensitive control messages sent between two end points.

**RECOMMENDED DSCP marking:**

- All flows in this service class are marked with A/V-Sig.

Applications or IP end points SHOULD pre-mark their packets with A/V-Sig DSCP value. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475].

**RECOMMENDED conditioning performed at DiffServ network edge:**

- Packet flow marking (DSCP setting) from untrusted sources (end user devices) SHOULD be verified at ingress to DiffServ network using Multifield (MF) Classification methods defined in [RFC2475].

- Packet flows from untrusted sources (end user devices) SHOULD be policed at ingress to DiffServ network, e.g., using single rate with burst size token bucket policer to ensure that the traffic stays within its negotiated or engineered bounds.

- Packet flows from trusted sources (application servers inside administered network) MAY not require policing.

- Policing of packet flows across peering points in which each peer is participating in the call set-up MUST be performed to the Service Level Agreement (SLA).

The fundamental service offered to "Conversational Signaling" traffic is enhanced best-effort service with controlled rate and delay. The service SHOULD be engineered so that A/V-Sig marked packet flows have sufficient bandwidth in the network to provide high assurance of delivery and low delay. Normally, traffic in this service class does not respond dynamically to packet loss. As such, Active Queue Management [RFC2309] SHOULD NOT be applied to A/V-Sig marked packet flows.

4.8. High-Throughput Data Service Class
The High-Throughput Data service class is RECOMMENDED for elastic applications that require timely packet forwarding of variable rate traffic sources and, more specifically, is configured to provide good throughput for TCP longer-lived flows. TCP [RFC1633] or a transport with a consistent Congestion Avoidance Procedure [RFC2581] [RFC3782] normally will drive as high a data rate as it can obtain over a long period of time. The FTP protocol is a common example, although one cannot definitively say that all FTP transfers are moving data in bulk.

The High-Throughput Data service class SHOULD use the Assured Forwarding (AF) PHB, defined in [RFC2597]. This service class SHOULD be configured to provide a minimum bandwidth assurance for AF11, AF12, and AF13 marked packets to ensure that they are forwarded in a timely manner. The High-Throughput Data service class SHOULD be configured to use a Rate Queuing system such as that defined in Section 1.4.1.2 of this document.

The following applications SHOULD use the High-Throughput Data service class:

- Store and forward applications.
- File transfer applications (e.g., FTP, HTTP, etc).
- Email.
- VPN service that supports two rates (committed information rate and excess or peak information rate).

The following are traffic characteristics:

- Variable size packets.
- Variable packet emission rate.
- Variable rate.
- With packet bursts of TCP window size.
- Source capable of reducing its transmission rate based on detection of packet loss at the receiver or through explicit congestion notification.

Applications or IP end points SHOULD pre-mark their packets with DSCP values as shown below. If the end point is not capable of setting the DSCP value, then the router topologically closest to the end point SHOULD perform Multifield (MF) Classification, as defined in [RFC2475], and mark all packets as AF1x. Note: In this case, the two-rate, three-color marker will be configured to operate in Color-Blind mode.
RECOMMENDED DSCP marking:

- AF11 = up to specified rate "A".
- AF12 = in excess of specified rate "A" but below specified rate "B".
- AF13 = in excess of specified rate "B".
- Where "A" < "B".

RECOMMENDED conditioning performed at DiffServ network edge:

- The two-rate, three-color marker SHOULD be configured to provide the behavior as defined in trTCM [RFC2698].
- If packets are marked by trusted sources or a previously trusted DiffServ domain and the color marking is to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Aware mode.
- If the packet marking is not trusted or the color marking is not to be preserved, then the two-rate, three-color marker SHOULD be configured to operate in Color-Blind mode.

The fundamental service offered to "High-Throughput Data" traffic is enhanced best-effort service with a specified minimum rate. The service SHOULD be engineered so that AF11 marked packet flows have sufficient bandwidth in the network to provide assured delivery. It can be assumed that this class will consume any available bandwidth and that packets traversing congested links may experience higher queuing delays or packet loss. Since the AF1x traffic is elastic and responds dynamically to packet loss, Active Queue Management [RFC2309] SHOULD be used primarily to control TCP flow rates at congestion points by dropping packets from TCP flows that have higher rates first. The probability of loss of AF11 traffic MUST NOT exceed the probability of loss of AF12 traffic, which in turn MUST NOT exceed the probability of loss of AF13. In such a case, if one network customer is driving significant excess and another seeks to use the link, any losses will be experienced by the high-rate user, causing him to reduce his rate. Explicit Congestion Notification (ECN) [RFC3168] MAY also be used with Active Queue Management.

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth for each DSCP, and the max-threshold specifies the queue depth above which all traffic with such a DSCP is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- min-threshold AF13 < max-threshold AF13
4.9. Standard Service Class

The Standard service class is RECOMMENDED for traffic that has not been classified into one of the other supported forwarding service classes in the DiffServ network domain. This service class provides the Internet's "best-effort" forwarding behavior. This service class typically has minimum bandwidth guarantee.

The Standard service class MUST use the Default Forwarding (DF) PHB, defined in [RFC2474], and SHOULD be configured to receive at least a small percentage of forwarding resources as a guaranteed minimum. This service class SHOULD be configured to use a Rate Queuing system such as that defined in Section 1.4.1.2 of this document.

The following applications SHOULD use the Standard service class:

- Network services, DNS, DHCP, BootP.
- Any undifferentiated application/packet flow transported through the DiffServ enabled network.

The following is a traffic characteristic:

- Non-deterministic, mixture of everything.

The RECOMMENDED DSCP marking is DF (Default Forwarding) ‘000000’.

Network Edge Conditioning:

There is no requirement that conditioning of packet flows be performed for this service class.

The fundamental service offered to the Standard service class is best-effort service with active queue management to limit overall delay. Typical configurations SHOULD use random packet dropping to implement Active Queue Management [RFC2309] or Explicit Congestion Notification [RFC3168], and MAY impose a minimum or maximum rate on the queue.
If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth, and the max-threshold specifies the queue depth above which all traffic is dropped or ECN marked. Thus, in this service class, the following inequality should hold in queue configurations:

- min-threshold DF < max-threshold DF
- max-threshold DF ≤ memory assigned to the queue

Note: Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

4.10. Low-Priority Data

The Low-Priority Data service class serves applications that run over TCP [RFC0793] or a transport with consistent congestion avoidance procedures [RFC2581] [RFC3782] and that the user is willing to accept service without guarantees. This service class is specified in [RFC3662] and [QBSS].

The following applications MAY use the Low-Priority Data service class:

- Any TCP based-application/packet flow transported through the DiffServ enabled network that does not require any bandwidth assurances.

The following is a traffic characteristic:

- Non-real-time and elastic.

Network Edge Conditioning:

- There is no requirement that conditioning of packet flows be performed for this service class.

The RECOMMENDED DSCP marking is CS1 (Class Selector 1).

The fundamental service offered to the Low-Priority Data service class is best-effort service with zero bandwidth assurance. By placing it into a separate queue or class, it may be treated in a manner consistent with a specific Service Level Agreement.

Typical configurations SHOULD use Explicit Congestion Notification [RFC3168] or random loss to implement Active Queue Management [RFC2309].

If RED [RFC2309] is used as an AQM algorithm, the min-threshold specifies a target queue depth, and the max-threshold specifies the queue depth above which all traffic is dropped or ECN marked. Thus,
in this service class, the following inequality should hold in queue configurations:

- min-threshold CS1 < max-threshold CS1
- max-threshold CS1 <= memory assigned to the queue

Note: Many other AQM algorithms exist and are used; they should be configured to achieve a similar result.

5. Additional Information on Service Class Usage

In this section, we provide additional information on how some specific applications should be configured to use the defined service classes.

5.1. Mapping for NTP

From tests that were performed, indications are that precise time distribution requires a very low packet delay variation (jitter) transport. Therefore, we suggest that the following guidelines for Network Time Protocol (NTP) be used:

- When NTP is used for providing high-accuracy timing within an administrator's (carrier's) network or to end users/clients, the audio service class SHOULD be used, and NTP packets should be marked with EF DSCP value.

- For applications that require "wall clock" timing accuracy, the Standard service class should be used, and packets should be marked with DF DSCP.

5.2. VPN Service Mapping

"Differentiated Services and Tunnels" [RFC2983] considers the interaction of DiffServ architecture with IP tunnels of various forms. Further to guidelines provided in RFC 2983, below are additional guidelines for mapping service classes that are supported in one part of the network into a VPN connection. This discussion is limited to VPNs that use DiffServ technology for traffic differentiation.

- The DSCP value(s) that is/are used to represent a PHB or a PHB group SHOULD be the same for the networks at both ends of the VPN tunnel, unless remarking of DSCP is done as ingress/egress processing function of the tunnel. DSCP marking needs to be preserved along the tunnel, end to end.

- The VPN MAY be configured to support one or more service classes. It is left up to the administrators of the two networks to agree on the level of traffic differentiation that will be provided in the network that supports VPN service. Service classes are then
mapped into the supported VPN traffic forwarding behaviors that meet the traffic characteristics and performance requirements of the encapsulated service classes.

- The traffic treatment in the network that is providing the VPN service needs to be such that the encapsulated service class or classes receive comparable behavior and performance in terms of delay, jitter, and packet loss and that they are within the limits of the service specified.

- The DSCP value in the external header of the packet forwarded through the network providing the VPN service can be different from the DSCP value that is used end to end for service differentiation in the end network.

- The guidelines for aggregation of two or more service classes into a single traffic forwarding treatment in the network that is providing the VPN service is for further study.

6. Security Considerations

This document discusses policy and describes a common policy configuration, for the use of a Differentiated Services Code Point by transports and applications. If implemented as described, it should require that the network do nothing that the network has not already allowed. If that is the case, no new security issues should arise from the use of such a policy.

It is possible for the policy to be applied incorrectly, or for a wrong policy to be applied in the network for the defined service class. In that case, a policy issue exists that the network SHOULD detect, assess, and deal with. This is a known security issue in any network dependent on policy-directed behavior.

A well-known flaw appears when bandwidth is reserved or enabled for a service (for example, voice and/or video transport) and another service or an attacking traffic stream uses it. This possibility is inherent in DiffServ technology, which depends on appropriate packet markings. When bandwidth reservation or a priority queuing system is used in a vulnerable network, the use of authentication and flow admission is recommended. To the author's knowledge, there is no known technical way to respond to an unauthenticated data stream using service that it is not intended to use, and such is the nature of the Internet.

The use of a service class by a user is not an issue when the SLA between the user and the network permits him to use it, or to use it up to a stated rate. In such cases, simple policing is used in the Differentiated Services Architecture. Some service classes, such as Network Control, are not permitted to be used by users at all; such traffic should be dropped or remarked by ingress filters. Where service classes are available under the SLA only to an authenticated
user rather than to the entire population of users, authentication and authorization services are required, such as those surveyed in [AUTHMECH].

7. Contributing Authors

This section specifically calls out the authors of RFC 4594, from which this document is based on.

Jozef Babiarz
Nortel Networks

Kwok Ho Chan
Nortel Networks

Fred Baker
Cisco Systems
EMail: fred@cisco.com

Of note, two of the three mentioned authors above worked for Nortel Networks at the time of writing RFC 4594, a company that no longer exists. This author has not seen or heard from those two in many, many years or IETF meetings - as a result of not knowing their new email addresses (or phone numbers).

While much of this document has been rewritten with either edited or brand new material, there are many short paragraphs that remain as they were from RFC 4594, as well as many sentences that were also left unchanged. Additionally, there were no new graphs, charts, diagrams, or tables introduced, meaning the first 4 tables within this document existed in RFC 4594, created by those authors. Presently, each of those tables contain modified and new information. The last 3 tables, specifically tables 5, 6, & 7 were removed because the examples section was removed.

This author believes there must be proper credit given for all the contributions, including the framework this document retains from that RFC. Periodically, throughout this document, what was written remains the best way of conveying a thought, rule, or otherwise stated behavior or mechanism. Because RFC 4594 was rather large, there is no realistic way of identifying each part that was left untouched. Further, properly quoting that RFC and leaving those sentences embedded in this document would render this document highly unreadable. Another application could be used to show the changes, deletions and additions - but not one that the IETF accepts presently.

This author has created this "Contributing Authors" section as a way of properly identifying those 3 individuals that provided text within this document. We will let the community judge if this is 'good enough' (i.e., rough consensus), or if another way is better.
8. Acknowledgements

To this document, you can have your name here if you comment or contribute to this effort to make it better. We thank you in advance.

The author would like to thank Paul Jones, Glen Lavers, Mo Zanaty, David Benham, and Michael Ramalho for their comments and questions about this effort that ultimately helped shape this document.

Below are the folks that were acknowledged in RFC 4594, and this author does not want to lose their recognition of contributions to the original effort.

"The authors thank the TSVWG reviewers, David Black, Brian E. Carpenter, and Alan O’Neill for their review and input to this document.

The authors acknowledge a great many inputs, most notably from Bruce Davie, Dave Oran, Ralph Santitoro, Gary Kenward, Francois Audet, Morgan Littlewood, Robert Milne, John Shuler, Nalin Mistry, Al Morton, Mike Pierce, Ed Koehler Jr., Tim Rahrer, Fil Dickinson, Mike Fidler, and Shane Amante. Kimberly King, Joe Zebarth, and Alistair Munroe each did a thorough proofreading, and the document is better for their contributions."

9. References

9.1. Normative References


[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G.,
9.2. Informative References


Appendix A - Changes

Here is a list of all the changes that were captured during the editing process. This will not be a complete list, and others are free to point out what the authors missed, and we’ll include that in the next release.
A.1 Since RFC 4594 to Individual -00

- rewrote Intro to emphasize current topics
- Created a Conversational Service group, comprising the audio, video and Hi-Res service classes - because they have similar characteristics.
- Incorporated the 6 new DSCPs from [ID-DSCP].
- moved the example section, en mass, to an appendix that might not be kept for this version. We're not sure it accomplishes what it needs to, and might not provide any real usefulness.
- Moved 'Realtime-Interactive’ service class to CS5, from CS4
- Changed ‘Broadcast Video’ service class to ‘Broadcast’ service class
- Changed AF4X to ‘Video’ service class, replacing ‘Multimedia Conferencing’ service class
- Moved ‘Multimedia Conferencing’ service class to different DSCPs
- Added the ‘Hi-Res’ service class
- Removed section 5.1 on signaling choices. It has been included in the main body of the text.
- Changed document title
- removed the last 3 tables, as they were part of the examples section that is no longer within this document.
This Internet-Draft, draft-polk-tsvwg-rsvp-app-id-vv-profiles-02.txt, has expired, and has been deleted from the Internet-Drafts directory. An Internet-Draft expires 185 days from the date that it is posted unless it is replaced by an updated version, or the Secretariat has been notified that the document is under official review by the IESG or has been passed to the RFC Editor for review and/or publication as an RFC. This Internet-Draft was not published as an RFC.

Internet-Drafts are not archival documents, and copies of Internet-Drafts that have been deleted from the directory are not available. The Secretariat does not have any information regarding the future plans of the authors or working group, if applicable, with respect to this deleted Internet-Draft. For more information, or to request a copy of the document, please contact the authors directly.

Draft Authors:
James Polk<jmpolk@cisco.com>
Subha Dhesikan<sdhesika@cisco.com>
Tunneling Compressed Multiplexed Traffic Flows (TCMTF)
draft-saldana-tsvwg-tcmtf-00

Abstract

This document describes a method to improve the bandwidth utilization of network paths that carry multiple streams in parallel between two endpoints, as in voice trunking. The method combines standard protocols that provide compression, multiplexing, and tunneling over a network path for the purpose of reducing the bandwidth used when multiple streams are carried over that path.

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 August 19, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction .................................................. 3
   1.1. Requirements Language ................................. 3
   1.2. Bandwidth efficiency of real-time flows .............. 3
   1.3. Real-time applications not using RTP ................. 4
   1.4. Scenarios of application ............................... 5
   1.5. Objective of this standard ............................. 5
   1.6. Overview of Protocols .................................. 6
2. Protocol Operation ............................................. 8
   2.1. Models of implementation ............................... 8
   2.2. Choice of the compressing protocol .................... 9
       2.2.1. Context Synchronization in ECRTP .................. 10
       2.2.2. Context Synchronization in ROHC .................. 10
   2.3. Multiplexing ........................................... 10
       2.3.1. Tunneling Inefficiencies ......................... 11
   2.4. Tunneling .............................................. 11
   2.5. Encapsulation Formats .................................. 11
3. Contributing Authors ......................................... 13
4. Acknowledgements ............................................. 14
5. IANA Considerations .......................................... 14
6. Security Considerations ..................................... 14
7. References .................................................... 14
   7.1. Normative References ................................. 14
   7.2. Informative References ............................... 16
Author’s Address ............................................... 16
1. Introduction

This document describes a way to combine existing protocols for compression, multiplexing, and tunneling to save bandwidth for some applications that generate small packets, such as real-time ones.

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 RFC 2119 [RFC2119].

1.2. Bandwidth efficiency of real-time flows

In the last years we are witnessing the rise of new real-time services that use the Internet for the delivery of interactive multimedia applications. The most common of these services is VoIP, but many others have been developed, and are experiencing a significant growth: videoconferencing, telemedicine, video vigilance, online gaming, etc.

The first design of the Internet did not include any mechanism capable of guaranteeing an upper bound for delivery delay, taking into account that the first deployed services were e-mail, file transfer, etc., in which delay is not critical. RTP [RTP] was first defined in 1996 in order to permit the delivery of real-time contents. Nowadays, although there are a variety of protocols used for signaling real-time flows (SIP [SIP], H.323, etc.), RTP has become the standard par excellence for the delivery of real-time content.

RTP was designed to work over UDP datagrams. This implies that an IPv4 packet carrying real-time information has to include 40 bytes of headers: 20 for IPv4 header, 8 for UDP, and 12 for RTP. This overhead is significant, taking into account that many real-time services send very small payloads. It becomes even more significant with IPv6 packets, as the basic IPv6 header is twice the size of the IPv4 header (Table 1).
In order to mitigate this bad network efficiency, the multiplexing of a number of payloads into a single packet can be considered as a solution. If we have only one flow, the number of samples included in a packet can be increased, but at the cost of adding new packetization delays. However, if a number of flows share the same path between an origin and a destination, a multiplexer can build a bigger packet in which a number of payloads share a common header. A demultiplexer is necessary at the end of the common path, so as to rebuild the packets as they were originally sent, making multiplexing a transparent process for the extremes of the flow.

The headers of the original packets can be compressed to save more bandwidth, taking into account that there exist some header compressing standards ([cRTP], [ECRTP], [IPHC], [ROHC]). When different headers are compressed together, tunneling can be used to relieve intermediate routers from the decompression and compression processing.

1.3. Real-time applications not using RTP

But there are many real-time applications that do not use RTP. Some of them send UDP packets, e.g. First Person Shooter (FPS) online games, for which latency is very critical. There is also another fact which has to be taken into account: TCP is getting used for media delivery. For many reasons, such as avoiding firewalls, the standard RTP/UDP/IP protocol stack is substituted in many cases by FLV/HTTP/TCP/IP (FLash Video [FLV]).

There is also another kind of applications which have been reported as real-time using TCP: MMORPGs (Massively Multiplayer Online Role Playing Games), which in some cases have millions of players, thousands of them sharing the same virtual world. They use TCP packets to send the player commands to the server, and also to send to the player’s application the characteristics and situation of other gamers’ avatars. These games do not have the same interactivity of FPSs, but the quickness and the movements of the...
player are important, and can decide if they win or lose a fight.

1.4. Scenarios of application

Different scenarios of application can be considered for the tunneling, compressing and multiplexing solution: for example, voice trunking between gateways of different offices of an enterprise. Also, the traffic of the users of an application in a town or a district, which can be multiplexed and sent to the central server. Also Internet cafes are suitable of having many users of the same application (e.g. a game) sharing the same access link.

Another interesting scenario are satellite communication links that often manage the bandwidth by limiting the transmission rate, measured in packets per second (pps), to and from the satellite. Applications like VoIP that generate a large number of small packets can easily fill the limited number of pps slots, limiting the throughput across such links. As an example, a G.729a voice call generates 50 pps at 20 ms packetization time. If the satellite transmission allows 1,500 pps, the number of simultaneous voice calls is limited to 30. This results in poor utilization of the satellite link’s bandwidth as well as places a low bound on the number of voice calls that can utilize the link simultaneously. Multiplexing small packets into one packet for transmission would improve the efficiency. Satellite links would also find it useful to multiplex small TCP packets into one packet. This could be especially interesting for compressing TCP ACKs.

There is still another interesting use case: desktop or application sharing where the traffic from the server to the client typically consists of the delta of screen updates. Also, the standard for remote desktop sharing emerging for WebRTC in the RTCWEB Working Group is: {something}/SCTP/UDP (Stream Control Transmission Protocol [SCTP]). In this scenario, SCTP/UDP could be used in other cases: chatting, file sharing and applications related to WebRTC peers. There could be hundreds of clients at a site talking to a server located at a datacenter over a WAN. Compressing, multiplexing and tunneling this traffic could save WAN bandwidth and potentially improve latency.

1.5. Objective of this standard

In conclusion, a standard that multiplexes, compresses and sends packets using a tunnel can be interesting for many enterprises: developers of VoIP systems can include this option in their solutions; or game providers, who can achieve bandwidth savings in their supporting infrastructures. Other fact that has to be taken into account is that the technique not only saves bandwidth but also
reduces the number of packets per second, which sometimes can be a bottleneck for a satellite link or even for a network router.

If only one stream is tunneled and compressed, then little bandwidth savings will be obtained. In contrast, multiplexing is helpful to amortize the overhead of the tunnel header over many payloads.

1.6. Overview of Protocols

The current standard [TCRTP] defines a way to combine different standard protocols. Three layers are considered, as shown in the figure:

```
+-----------------------------+                  +-----------------------------+
| RTP/UDP                     |                  | ECRTP                       |
|                            +-------------------+-----------------------------+ |
|                            |       compressing layer          | multiplexing layer          |
|                            +-------------------+-----------------------------+ |
|                            |                   +-----------------------------+ |
|                            |                   | tunneling layer              |
|                            +-----------------------------+ |
|                            |                   +-----------------------------+ |
|                            |                   | IP                           |
```

Figure 1

In contrast, the new proposal includes other protocols to be compressed in addition to RTP/UDP, since real-time services can also be provided using UDP or TCP.
Each of the three layers is considered as independent of the other two, i.e. different combinations of protocols can be implemented according to the new standard:

- Regarding compression, a number of options can be considered: as the standards are able to compress different headers, the one to be used could be selected depending on the traffic to compress. It also exists the possibility of having a null header compression, in the case of wanting to avoid traffic compression, taking into account the need of storing a context for every flow and the problems of context desynchronization in certain scenarios. For this, different header compression protocols have been defined ([cRTP], [ECRTP], [IPHC], [ROHC]) by the IETF.

- Multiplexing is accomplished using PPP Multiplexing [PPP-MUX]. Nevertheless, other multiplexing protocols can also be considered.

- Tunneling is accomplished by using L2TP (Layer 2 Tunneling Protocol [L2TPv3]), GRE (Generic Routing Encapsulation [GRE]) or other schemes.

Finally, another option has been considered: A payload compression layer. When the payload is G.711 this layer can runs G.711.0, a
lossless and stateless compression/decompression of the payload [I.711]. This operation can be deployed by network elements like routers/switches, without the endpoints having to signal it using RTSP/SDP/SIP, since G.711 has a fixed RTP payload number.

2. Protocol Operation

This section describes how to combine three protocols: compressing, multiplexing, and tunneling, to save bandwidth for real-time applications.

2.1. Models of implementation

TCMTF can be implemented in different ways. The most straightforward is to implement it in the devices terminating the real-time streams (these devices can be e.g. voice gateways, or proxies grouping a number of flows):

[end device]---[end device]

\[
\begin{array}{c}
\text{TCMTF over IP}
\end{array}
\]

Figure 3

Another way TCMTF can be implemented is with an external concentration device. This device could be placed at strategic places in the network and could dynamically create and destroy TCMTF sessions without the participation of the endpoints that generate real-time flows.

[end device]\/[end device]
[end device]---[concentrator]---[concentrator]---[end device]
[end device]

\[
\begin{array}{c}
\text{Native IP} \quad \text{TCMTF over IP} \quad \text{Native IP}
\end{array}
\]

Figure 4

Such a design also allows classical compressing protocols to be used on links with only a few active flows per link.
2.2. Choice of the compressing protocol

There are different protocols that can be used for compressing real-time flows:

- IPHC (IP Header Compression [IPHC]) permits the compression of TCP/IP, UDP/IP and ESP/IP headers (Encapsulating Security Payload [ESP]). It has a low implementation complexity. On the other hand, the resynchronization of the context can be slow over long RTT links. It should be used in scenarios presenting very low packet loss percentage.

- cRTP (compressed RTP [cRTP]) works the same way as IPHC, but is also able to compress RTP headers. The link layer transport is not specified, but typically PPP is used. For cRTP to compress headers, it must be implemented on each PPP link. A lot of context is required to successfully run cRTP, and memory and processing requirements are high, especially if multiple hops must implement cRTP to save bandwidth on each of the hops. At higher line rates, cRTP’s processor consumption becomes prohibitively expensive. cRTP is not suitable over long-delay WAN links commonly used when tunneling, as proposed by this document. To avoid the per-hop expense of cRTP, a simplistic solution is to use cRTP with L2TP to achieve end-to-end cRTP. However, cRTP is only suitable for links with low delay and low loss. However, once multiple router hops are involved, cRTP’s expectation of low delay and low loss can no longer be met. Further, packets can arrive out of order.

- ECRTP (Enhanced Compressed RTP [ECRTP]) is an extension of cRTP [cRTP] that provides tolerance to packet loss and packet reordering between compressor and decompressor. Thus, ECRTP should be used instead of cRTP.

- ROHC (RObust Header Compression [ROHC]) is able to compress TCP/IP, UDP/IP, ESP/IP and RTP/UDP/IP headers. It is a robust scheme developed for header compression over links with high bit error rate, such as wireless ones. It incorporates mechanisms for
quick resynchronization of the context. It includes an improved encoding scheme for compressing the header fields that change dynamically. Its main drawback is that it requires significantly more processing and memory resources than the ones necessary for IPHC or ECRTP.

This standard does not determine which of the existing protocols has to be used for the compressing layer. The decision will depend on the scenario, and will mainly be determined by the packet loss probability, RTT, and the availability of memory and processing resources. The standard is also suitable to include other compressing schemes that may be further developed.

2.2.1. Context Synchronization in ECRTP

When the compressor receives an RTP packet that has an unpredicted change in the RTP header, the compressor should send a COMPRESSED_UDP packet (described in [ECRTP]) to synchronize the ECRTP decompressor state. The COMPRESSED_UDP packet updates the RTP context in the decompressor.

To ensure delivery of updates of context variables, COMPRESSED_UDP packets should be delivered using the robust operation described in [ECRTP].

Because the "twice" algorithm described in [ECRTP] relies on UDP checksums, the IP stack on the RTP transmitter should transmit UDP checksums. If UDP checksums are not used, the ECRTP compressor should use the cRTP Headers checksum described in [ECRTP].

2.2.2. Context Synchronization in ROHC

ROHC [ROHC] includes a more complex mechanism in order to maintain context synchronization. It has different operation modes and defines compressor states which change depending on link behavior.

2.3. Multiplexing

Header compressing algorithms require a layer two protocol that allows identifying different protocols. PPP [PPP] is suited for this, although other multiplexing protocols can also be used for this layer of TCMTF.

When header compression is used inside of a tunnel, it will reduce the size of the IP, UDP, and IP headers of the IP packet carried in the tunnel. However, the tunnel itself has overhead due to its IP header and the tunnel header (the information necessary to identify the tunneled payload). One way to reduce the overhead of the IP
header and tunnel header is to multiplex multiple real-time payloads in a single tunneled packet.

2.3.1. Tunneling Inefficiencies

To get reasonable bandwidth efficiency using multiplexing within an L2TP tunnel, multiple real-time streams should be active between the source and destination of an L2TP tunnel. The packet size of the real-time streams has to be small in order to permit a good bandwidth saving.

If the source and destination of the L2TP tunnel are the same as the source and destination of the compressing protocol sessions, then the source and destination must have multiple active real-time streams to get any benefit from multiplexing.

Because of this limitation, TCMTF is mostly useful for applications where many real-time sessions run between a pair of endpoints. The number of simultaneous sessions required to reduce the header overhead to the desired level depends on the size of the L2TP header. A smaller L2TP header will result in fewer simultaneous sessions being required to produce adequate bandwidth efficiencies.

2.4. Tunneling

L2TP tunnels should be used to tunnel the ECRTP payloads end to end. L2TP includes methods for tunneling messages used in PPP session establishment, such as NCP (Network Control Protocol). This allows [IPCP-HC] to negotiate ECRTP compression/decompression parameters.

Other tunneling schemes, such as GRE [GRE] may also be used to implement the tunneling layer of TCMTF.

2.5. Encapsulation Formats

The packet format for a packet compressed is:
The packet format of a multiplexed PPP packet as defined by [PPP-MUX] is:

```
+-------+---+------+-------+-----+   +---+------+-------+-----+
| Mux   |P L|      |       |     |   |P L|      |       |     |
| PPP   |F X|Len1  |  PPP  |     |   |F X|LenN  |  PPP  |     |
| Prot. |F T|      | Prot. |Info1| ˜ |F T|      | Prot. |InfoN|
| Field |          | Field1|     |   |          |FieldN |     |
| (1)   |1-2 octets| (0-2) |     |   |1-2 octets| (0-2) |     |
```

Figure 7

The combined format used for TCMTF with a single payload is all of the above packets concatenated. Here is an example with one payload:

```
+------+------+-------+----------+-------+--------+----+
| IP   |Tunnel| Mux   |P L|      |       |        |    |
| header|header| PPP   |F X|Len1  |  PPP  | Compr  |    |
| (20) |      | Proto |F T|      | Proto | header |Data|
|      |      | Field |          | Field1|        |    |
|      |      | (1)   |1-2 octets| (0-2) |        |    |
```

If the tunnel contains multiplexed traffic, multiple "PPPMux payload"s are transmitted in one IP packet.
3. Contributing Authors

Dan Wing
Cisco Systems
771 Alder Drive
San Jose, CA  95035
US
Phone: +44 7889 488 335
Email: dwing@cisco.com

Julian Fernandez-Navajas
University of Zaragoza
Dpt. IEC Ada Byron Building
Zaragoza, 50018
Spain
Phone: +34 976 761 963
Email: navajas@unizar.es

Jose Ruiz-Mas
University of Zaragoza
Dpt. IEC Ada Byron Building
Zaragoza, 50018
Spain
Phone: +34 976762158
Email: jruiz@unizar.es

Muthu Arul Mozhi Perumal
Cisco Systems, Inc.
Cessna Business Park
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka  560103
India
Phone:  +91 9449288768
Email: mperumal@cisco.com
4. Acknowledgements

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

All drafts are required to have a security considerations section. See RFC 3552 [RFC3552] for a guide.

7. References

7.1. Normative References


[GRE] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Saldana

Expires August 19, 2012


7.2. Informative References


Author's Address

Jose Saldana (editor)
University of Zaragoza
Dpt. IEC Ada Byron Building
Zaragoza, 50018
Spain

Phone: +34 976 762 698
Email: jsaldana@unizar.es
ECN for Stream Control Transmission Protocol (SCTP)
draft-stewart-tsvwg-sctpecn-02.txt

Abstract

This document describes the addition of the ECN to the Stream Control Transmission Protocol (SCTP).

Status of this Memo

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

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

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

This Internet-Draft will expire on August 20, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

1. Introduction ............................................ 3
2. Conventions ........................................... 3
3. Terminology ............................................ 3
4. Chunk and Parameter Formats ........................... 4
   4.1. ECN Support Parameter (32768) ..................... 4
   4.2. ECN Echo (12) ..................................... 4
   4.3. CWR Chunk(13) ................................... 5
5. Procedures ............................................... 5
   5.1. SCTP Initialization ................................ 5
   5.2. The SCTP Sender .................................... 7
   5.3. The SCTP Receiver ................................. 9
   5.4. Congestion on the SACK path ...................... 10
   5.5. Retransmitted SCTP Packets ...................... 10
   5.6. SCTP Window Probes ................................ 10
6. Security Considerations ................................ 10
7. IANA Considerations .................................... 11
8. Acknowledgements ....................................... 11
9. References .............................................. 11
   9.1. Normative references .............................. 11
   9.2. Informational References ......................... 11
Authors’ Addresses ........................................ 11
1. Introduction

At the time SCTP was initially defined in [RFC2960] ECN - [RFC2481] was still an experimental document. This left the authors of SCTP in a position where they could not directly refer to ECN without creating a normative reference in a standards track document to an experimental RFC. To work around this problem the authors of SCTP decided to add two reserved chunk types for ECN (CWR and ECNE) but did not fully specify how they were to be used except in a vague way within an appendix of the document. This worked around the document reference problem, but left ECN and its implementation for SCTP unspecified. This document is intended to fill in the details of ECN processing in SCTP in a standards track document.

This document assumes that the reader is familiar with ECN [RFC3168]. Readers unfamiliar with ECN are strongly encouraged to first read [RFC3168] since this document will not repeat any of the details on how the various IP level bits are set. This document will use the same terminology has [RFC3168]. For example the term ECT is used to indicate that the IP level packet is marked indicating the transport (SCTP) supports ECN.

2. Conventions

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

3. Terminology

All integer fields defined in this document included in an SCTP packet MUST be transmitted in network byte order, unless otherwise stated.

ECT  The term used to indicate that the IP level packet is marked indicating the transport is willing to support ECN for this packet.

not-ECT  The term used to indicate that the IP level packet is marked indicating the transport is NOT willing to support ECN for this packet.

CE  The term used to indicate that the IP level packet is marked indicating that a router in the network has marked the packet as having experienced congestion.
4. Chunk and Parameter Formats

4.1. ECN Support Parameter (32768)

This parameter is used to indicate the support for ECN. If this parameter is present, the sender of the chunk is indicating that it supports ECN and wishes to use ECN for the newly forming association.

Valid Chunk Appearance

The ECN Supported Parameter may appear in the INIT, or the INIT-ACK chunk type.

4.2. ECN Echo (12)

This parameter contains the lowest TSN number contained in the last packet received that was marked by the network with a CE indication.

Number CE Marked Packets: 32 bits (unsigned integer)
This parameter contains the total number of CE marked packets that has been seen since the first CE mark received while waiting for a CWR chunk.

Note that the appendix of [RFC4960] did not have the field Number CE Marked Packets. Implementations SHOULD accept an 8 byte form of this chunk that does not include this field. In such a case the implementation SHOULD treat the missing field as indicating one CE marked packet for any purpose for which the implementation is using this field.

4.3. CWR Chunk(13)

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chunk Flags: 8 bits

The o Bit indicates if the CWR is a retransmission of an earlier CWR that may have been lost. If this bit is set, then the TSN number included is the latest TSN that a CWR has been responded to. If the o bit is clear, than the TSN indicated is the latest TSN for that destination.

Set to all zeros on transmit and ignored on receipt.

TSN Number: 32 bits (unsigned integer)

This parameter contains the TSN number to which the sender has reduced his congestion window to.

5. Procedures

5.1. SCTP Initialization

In the SCTP association setup phase, the source and destination SCTP endpoints exchange information about their willingness to use ECN. After the completion of this negotiation, an SCTP sender sets an ECT
codepoint in the IP header of data packets to indicate to the network that the transport is capable and willing to participate in ECN for this packet. This indicates to the routers that they may mark this packet with the CE codepoint.

If the SCTP association does not wish to use ECN notification for a particular packet, the sending SCTP sets the ECN codepoint to not-ECT, and the SCTP receiver ignores the CE codepoint in the received packet.

For this discussion we will call the endpoint initiating the SCTP association as EP-A and the listening SCTP endpoint as EP-Z.

Before an SCTP association can use ECN, EP-A sends an INIT chunk which includes the ECN Support parameter. By including the ECN Support parameter the sending endpoint (EP-A) will participate in ECN as both a sender and a receiver. Specifically, as a receiver, it will respond to incoming data packets that have the CE codepoint set in the IP header by sending an ECN Echo chunk bundled with the next outgoing SACK Chunk. As a sender, it will respond to incoming packets that include an ECN Echo chunk by reducing the congestion window and sending a CWR chunk when appropriate.

Including an ECN Support parameter in an INIT or INIT-ACK does not commit the SCTP sender to setting the ECT codepoint in any or all of the packets it may transmit. However, the commitment to respond appropriately to incoming packets with the CE codepoint set remains.

When EP-Z sends INIT-ACK chunk, it also includes an ECN Support parameter. Including the ECN Support parameter indicates that the SCTP transmitting the INIT-ACK chunk is ECN-Capable.

The following rules apply to the use of ECN for an SCTP association.

* If the SCTP Endpoint supports ECN a sender of either an INIT or INIT-ACK chunk MUST ALWAYS include the ECN Supported Parameter.

* After the exchange of the INIT and INIT-ACK if both endpoints have NOT indicated support of ECN by including an ECN Supported Parameter, then ECT MUST NOT be set on any IP packets sent by any endpoint which is ECN capable. Furthermore upon receiving IP packets with a CE codepoint set, the ECN capable endpoint SHOULD ignore the CE codepoint.

* If both endpoints have included an ECN Supported Parameter in the INIT and INIT-ACK exchange, then both endpoints MUST follow the ECN procedures defined in the rest of this document.
* A sending endpoint SHOULD set the ECT code points on IP packets that carry Data chunk. This includes IP packets that have other control chunks bundled with the Data.

5.2. The SCTP Sender

For an SCTP association using ECN, new data packets are transmitted with an ECT codepoint set in the IP header. When only one ECT codepoint is needed by a sender for all packets sent on an SCTP association ECT(0) SHOULD be used. If the sender receives an ECN-Echo chunk packet, then the sender knows that congestion was encountered in the network on the path from the sender to the receiver. The indication of congestion should be treated just as a congestion loss in non-ECN-Capable SCTP. That is, the SCTP source halves the congestion window "cwnd" for the destination address that the sender transmitted the data to and reduces the slow start threshold "ssthresh". A packet containing an ECN-Echo chunk shouldn’t trigger new data to be sent. SCTP follows the normal procedures for increasing the congestion window when it receives a packet with a SACK chunk without the ECN Echo chunk.

SCTP should not react to congestion indications more than once every round-trip time. That is, the SCTP sender’s congestion window should be reduced only once in response to a series of dropped and/or CE packets from a single window of data. In addition, the SCTP source should not decrease the slow-start threshold, ssthresh, if it has been decreased within the last round trip time.

One method to accomplish this is as following:

1) During association setup, create a new state variable ECN_ECHO_TSN and ECN_ECHO_LAST for each destination. The initial value of these variables are set to the initial TSN that will be assigned minus 1.

2) When an ECN Echo chunk arrives, use the TSN in the ECN Echo to establish which destination the packet was sent to. We will call this destination the selected destination. If the chunk cannot be found note that an override is occurring from the selected destination (if found) select its ECN Echo TSN.

3) Compare the ECN Echo TSN with the ECN_ECHO_TSN for the selected destination. If an override is not noted and the value of the ECN_ECHO_TSN is greater than the ECN Echo TSN proceed to step 4; else proceed to step 6b.
4) Reduce the cwnd and ssthresh for the selected destination the same as if a loss was detected during a fast retransmit. For details, see [RFC4960] Section 7.2.3 and Section 7.2.4.

5) Record in the ECN_ECHO_TSN value, the last TSN that was sent and recorded in ECN_ECHO_LAST the TSN number from the ECN Echo Chunk.

6a) If the implementation is tracking the number of marked packets, record the value found in the 'Number CE Marked Packets Seen since CWR' field and also add this number to the running loss count. If such a count is not being maintained, then proceed to step 7.

6b) If the implementation is tracking the number of marked packets, compare the number in the ECN Echo Chunk TSN to the ECN_ECHO_LAST. If it is greater than ECN_ECHO_LAST, update ECN_ECHO_LAST with this value. Take the difference between the stored 'Number CE Marked Packets' field and the value from the newly arriving 'Number CE Marked Packets' and add this difference to the total loss count. Then update the stored 'Number CE Marked Packets' with the ECN Echo Chunk TSN.

7) Create a CWR chunk with the value found in the ECN_ECHO_LAST for the selected destination. If an override was noted, set the 'O' bit within the CWR flags. Queue this chunk for transmission to the peer destination. Note if there is already such a chunk in queue to be sent, remove that chunk and replace it with the new chunk.

After the sending SCTP reduces its congestion window in response to a ECN Echo, incoming SACKs that continue to arrive can "clock out" outgoing packets as allowed by the reduced congestion window. Note that continued arrival of ECN Echo chunks should still be processed as described above, possibly reducing the cwnd, but always sending a CWR to the receiving SCTP. This assures that the ECN Echo and CWR are robust with regard to loss in either direction and that the implementation, if it desires, can maintain an accurate loss count per destination.

Note, originally in the appendix of [RFC4960] a definition was supplied for the ECN Echo chunk. This definition did NOT include the 'Number CE Marked Packets' field. An implementation SHOULD accept such a chunk, delineating it from the standards track version by the fact that the length field will be 8 bytes instead of 12. When processing this older style chunk, the 'Number CE Marked Packets' should be treated as if it contains the number 1. This may cause incorrect loss counts but will NOT cause any issues with SCTP’s ECN handling.
5.3. The SCTP Receiver

When an SCTP endpoint first receives a CE data packet at the destination end-system, the SCTP data receiver creates an ECN Echo chunk and records the lowest TSN number found in the data packet. It also sets the ‘Number CE Marked Packets’ to 1 and queues this chunk for transmission at the next opportunity. If there is any ACK withholding implemented, as in current "delayed-SACK" SCTP implementations where the SCTP receiver can send an SACK for two arriving data packets, then the ECN Echo chunk will not be sent until the SACK is sent. If the next arriving data packet also has the CE codepoint set, then the receiver updates the queued ECN Echo chunk to have a higher TSN value (the lowest one in the newly arriving data packet) and increments the ‘Number CE Marked Packets’ field in the queued chunk.

Multi-homing requires one added restriction upon the ECN Echo chunk, such a chunk MUST be bundled with a SACK, and the SACK MUST follow the ECN Echo Chunk. This ordering is necessary so that the receiver of the ECN Echo chunk will at least one time find the proper destination to which the chunk was originally sent. Without this restriction it is possible a SACK could arrive ahead of the ECN Echo Chunk, no matter what the sending order, causing the sender to free the DATA chunk and thus loose the association with what destination it was sent to. For the same reason we also require the ECN Echo Chunk be earlier in the packet ahead of the SACK so that the SACK is not processed before the ECN Echo Chunk.

After transmission of the ECN Echo chunk, usually bundled with the SACK, the receiver does NOT discard the ECN Echo chunk. Instead it keeps the chunk in its queue and continues to send this chunk bundled with at least a SACK chunk on each outgoing packet, updating it as described above if other CE codepoint data packets arrive. The ECN Echo chunk should only be discarded when a CWR Chunk arrives holding a TSN value that is greater than or equal to the value inside the ECN Echo Chunk.

This provides robustness against the possibility of a dropped SACK packet carrying an ECN Echo chunk. The SCTP receiver continues to transmit the ECN Echo chunk in subsequent SACK packets until the correct CWR is received.

After the receipt of the CWR chunk, acknowledgments for subsequent non-CE data packets will not have an ECN Echo chunk bundled with them. If another CE packet is received by the data receiver, the receiver would once again send SACK packets bundled with a newly created ECN Echo chunk. The receipt of a CWR packet guarantees that the data sender has received the ECN Echo chunk for the TSN.
specified, and reduced its congestion window at some point *after* it sent the data packet for which the CE codepoint was set.

When processing a CWR, it is important that the receiver of the CWR validate the source address from which the CWR came from. It SHOULD match the destination the ECN Echo was sent to unless the override bit is set in the CWR Chunk.

5.4. Congestion on the SACK path

For the current generation of SCTP congestion control algorithms, pure acknowledgement packets (e.g., packets that do not contain any accompanying data) MUST be sent with the not-ECT codepoint. Current SCTP receivers have no mechanisms for reducing traffic on the SACK-path in response to congestion notification. Mechanisms for responding to congestion on the SACK-path are areas for current and future research. For current SCTP implementations, a single dropped SACK generally has only a very small effect on SCTP’s sending rate.

5.5. Retransmitted SCTP Packets

This document specifies ECN-capable SCTP implementations MUST NOT set either ECT codepoint (ECT(0) or ECT(1)) in the IP header for retransmitted data packets, and that the SCTP data receiver SHOULD ignore the ECN field on arriving data packets that are outside of the receiver’s current window. The reasons for this can be found in [RFC3168] Section 6.1.5.

5.6. SCTP Window Probes

When the SCTP data receiver advertises a zero window, the SCTP data sender sends window probes to determine if the receiver’s window has increased. Window probe packets for SCTP do contain user data (one chunk). If a window probe packet is dropped in the network, this loss can be detected by the receiver. Therefore, the SCTP data sender MAY set an ECT codepoint on the initial send of the window probe, but the SCTP sender MUST NOT set the ECT codepoint on retransmissions of that TSN.

6. Security Considerations

[RFC3168] defines the security considerations for ECN. These same consideration that are described for TCP are applicable to SCTP.
7. IANA Considerations

TBD

8. Acknowledgements

9. References

9.1. Normative references


9.2. Informational References


Authors' Addresses

Randall R. Stewart
Adara Networks
Chapin, SC 29036
USA

Email: randall@lakerest.net
Michael Tuexen  
Muenster University of Applied Sciences  
Stegerwaldstr. 39  
48565 Steinfurt  
Germany  
Email: tuexen@fh-muenster.de

Xuesong Dong  
Huawei  
Pleasanton, CA 94566  
USA  
Email: stevedong@huawei.com
Abstract

The Stream Control Transmission Protocol (SCTP) is a transport protocol originally defined to run on top of the network protocols IPv4 or IPv6. This memo document specifies how SCTP can be used on top of the Datagram Transport Layer Security (DTLS) protocol. SCTP over DTLS is used by the RTCWeb protocol suite for transporting non-media data between browsers.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 5, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction .................................................. 3
2. Conventions .................................................. 3
3. Encapsulation and Decapsulation Procedure ..................... 3
4. DTLS Considerations ............................................ 4
5. SCTP Considerations ............................................ 4
6. IANA Considerations ............................................ 6
7. Security Considerations ........................................ 6
8. Acknowledgments ................................................ 6
9. References ..................................................... 6
   9.1. Normative References .................................... 6
   9.2. Informative References ................................... 7
Authors’ Addresses .................................................. 7
1. Introduction

1.1. Overview

The Stream Control Transmission Protocol (SCTP) as defined in [RFC4960] is a transport protocol running on top of the network protocols IPv4 or IPv6. This memo document specifies how SCTP can be used on top of the Datagram Transport Layer Security (DTLS) protocol. SCTP over DTLS is used by the RTCWeb protocol suite (see [I-D.ietf-rtcweb-overview] for an overview) for transporting non-media data between browsers. The architecture of this stack is described in [I-D.jesup-rtcweb-data].

1.2. Terminology

This document uses the following terms:

Association: An SCTP association.

Stream: A unidirectional stream of an SCTP association. It is uniquely identified by a stream identifier.

1.3. Abbreviations


MTU: Maximum Transmission Unit.

PPID: Payload Protocol Identifier.


2. Conventions

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

3. Encapsulation and Decapsulation Procedure

When an SCTP packet is sent down to the DTLS layer, the complete SCTP packet, consisting of the SCTP common header and a number of SCTP
chunks, MUST be handled as the payload of the application layer protocol of DTLS. When the DTLS layer has processed a DTLS record containing a message of the application layer protocol, the payload MUST be given up to the SCTP layer. The SCTP layer expects an SCTP common header followed by a number of SCTP chunks.

4. DTLS Considerations

The DTLS implementation MUST be based on [RFC6347].

If path MTU discovery is performed by the DTLS layer, the method described in [RFC4821] MUST be used. For probe packets, the extension defined in [RFC6520] MUST be used.

If path MTU discovery is performed by the SCTP layer and IPv4 is used as the network layer protocol, the DTLS implementation MUST allow the DTLS user to enforce that the corresponding IPv4 packet is sent with the DF bit set.

SCTP performs segmentation and reassembly based on the path MTU. Therefore the DTLS layer MUST NOT use any compression algorithm.

5. SCTP Considerations

5.1. Base Protocol

SCTP as specified in [RFC4960] is used. However, the following restrictions are necessary to reflect that the lower layer is the connection oriented protocol DTLS instead of the connection less protocol IPv4 and IPv6:

- A DTLS connection MUST be established before an SCTP association can be set up.
- All associations MUST be single-homed.
- 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.
- 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

The SCTP implementation MUST support the Supported Extensions Parameter defined in [RFC5061] to signal the support of the SCTP stream reset extension (see Section 5.6). The other functionality described in [RFC5061] MUST NOT be used.

5.4. SCTP Authentication Extension

The SCTP authentication extension defined in [RFC4895] is not required.

5.5. Partial Reliability Extension

The SCTP implementation MUST support the extension defined in [RFC3758].

The SCTP implementation SHOULD support the following PR-SCTP policies:

- A user message is abandoned after a user specified lifetime.
- A user message is abandoned if the number of retransmissions exceeds a user specified threshold.

5.6. Stream Reset Extension

The SCTP implementation MUST support the SCTP stream reset extension defined in [RFC6525]. It is used to reset streams and add streams during the lifetime of the SCTP association.

5.7. Large User Message Extension

SCTP as defined in [RFC4960] does not support the multiplexing of large user messages that need to be fragmented and reassembled by the SCTP layer. To overcome this limitation, the SCTP implementation SHOULD support an extension, which has to be defined.

5.8. Congestion Control

In addition to the TCP-like congestion control specified in [RFC4960], other congestion control algorithms MAY be provided. For
example, it might be helpful to use a congestion control which does not increase the queueing delay substantially (see [I-D.ietf-ledbat-congestion] for an example).

6. IANA Considerations

   This document requires no actions from IANA.

7. Security Considerations

   TBD.

8. Acknowledgments

   The authors wish to thank XXX for their invaluable comments.

9. References

9.1. Normative References


September 2007.


9.2. Informative References


Authors’ Addresses

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

Phone: +1-215-354-5166
Email: randell_ietf@jesup.org
Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
FI
Email: Salvatore.Loreto@ericsson.com

Randall R. Stewart
Adara Networks
Chapin, SC 29036
US
Email: randall@lakerest.net

Michael Tuexen
Muenster University of Applied Sciences
Stegerwaldstrasse 39
48565 Steinfurt
DE
Email: tuexen@fh-muenster.de
Load Sharing for the Stream Control Transmission Protocol (SCTP)
draft-tuexen-tsvwg-sctp-multipath-04.txt

Abstract

The Stream Control Transmission Protocol (SCTP) supports multi-homing for providing network fault tolerance. However, mainly one path is used for data transmission. Only timer-based retransmissions are carried over other paths as well.

This document describes how multiple paths can be used simultaneously for transmitting user messages.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at 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 11, 2012.
Table of Contents

1. Introduction ................................................. 4
2. Conventions .................................................. 4
3. Load Sharing .................................................. 4
   3.1. Split Fast Retransmissions ............................... 4
   3.2. Appropriate Congestion Window Growth .................. 5
   3.3. Appropriate Delayed Acknowledgements .................. 5
4. Non-Renegable SACK ............................................ 6
   4.1. Negotiation .............................................. 6
   4.2. The New Chunk Type: Non-Renegable SACK (NR-SACK) ...... 7
   4.3. An Illustrative Example .................................. 11
   4.4. Procedures ............................................... 15
      4.4.1. Sending an NR-SACK chunk ........................... 15
      4.4.2. Receiving an NR-SACK Chunk ......................... 17
5. Buffer Blocking Mitigation .................................... 18
   5.1. Sender Buffer Splitting .................................. 18
   5.2. Receiver Buffer Splitting ................................ 18
   5.3. Problems during Path Failure ............................ 18
      5.3.1. Problem Description .................................. 18
      5.3.2. Solution: Potentially-failed Destination State .... 19
   5.4. Non-Renegable SACK ....................................... 19
      5.4.1. Problem Description .................................. 19
      5.4.2. Solution: Non-Renegable SACKs ........................ 20
6. Handling of Shared Bottlenecks ............................... 20
   6.1. Introduction .............................................. 20
   6.2. Initial Values ........................................... 21
   6.3. Congestion Window Growth ................................ 21
   6.4. Congestion Window Decrease ............................... 21
7. Chunk Scheduling .............................................. 21
8. Socket API Considerations .................................... 21
9. IANA Considerations .......................................... 21
   9.1. A New Chunk Type ........................................ 21
10. Security Considerations ...................................... 22
11. Acknowledgments ............................................. 22
12. References ................................................ 22
   12.1. Normative References .................................... 22
   12.2. Informative References .................................. 23
Authors’ Addresses ............................................. 24
1. Introduction

One of the important features of the Stream Control Transmission Protocol (SCTP), which is currently specified in [RFC4960], is network fault tolerance. This feature is for example required for Reliable Server Pooling (RSerPool, [RFC5351]). Therefore, transmitting messages over multiple paths is supported, but only for redundancy. So [RFC4960] does not specify how to use multiple paths simultaneously.

This document overcomes this limitation by specifying how multiple paths can be used simultaneously. This has several benefits:

- Improved bandwidth usage.
- Better availability check with real user messages compared to HEARTBEAT-based information.

2. Conventions

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

3. Load Sharing

Basic requirement for applying SCTP load sharing is the Concurrent Multipath Transfer (CMT) extension of SCTP, which utilises multiple paths simultaneously. We denote CMT-enabled SCTP as CMT-SCTP throughout this document. CMT-SCTP is introduced in [IAS06] and in more detail in [I06], some illustrative examples of chunk handling are provided in [DBP10a]. CMT-SCTP provides three modifications to standard SCTP (split Fast Retransmissions, appropriate congestion window growth and delayed SACKs), which are described in the following subsections.

3.1. Split Fast Retransmissions

Paths with different latencies lead to overtaking of DATA chunks. This leads to gap reports, which are handled by Fast Retransmissions. However, due to the fact that multiple paths are used simultaneously, these Fast Retransmissions are usually useless and furthermore lead to a decreased congestion window size.

To avoid unnecessary Fast Retransmissions, the sender has to keep track of the path each DATA chunk has been sent on and consider...
transmission paths before performing Fast Retransmissions. That is, on reception of a SACK, the sender MUST identify the highest acknowledged TSN on each path. A chunk SHOULD only be considered as missing if its TSN is smaller than the highest acknowledged TSN on its path. Section 3.1 of [DBP10a] contains an illustrated example.

3.2. Appropriate Congestion Window Growth

The congestion window adaptation algorithm for SCTP [RFC4960] allows increasing the congestion window only when a new cumulative ack (CumAck) is received by a sender. When SACKs with unchanged CumAcks are generated (due to reordering) and later arrive at a sender, the sender does not modify its congestion window. Since a CMT-SCTP receiver naturally observes reordering, many SACKs are sent containing new gap reports but not new CumAcks. When these gaps are later acked by a new CumAck, congestion window growth occurs, but only for the data newly acked in the most recent SACK. Data previously acked through gap reports will not contribute to congestion window growth, in order to prevent sudden increases in the congestion window resulting in bursts of data being sent.

To overcome the problems described above, the congestion window growth has to be handled as follows [IAS06]:

o The sender SHOULD keep track of the earliest non-retransmitted outstanding TSN per path.

o The sender SHOULD keep track of the earliest retransmitted outstanding TSN per path.

o The in-order delivery per path SHOULD be deduced.

o The congestion window of a path SHOULD be increased when the earliest non-retransmitted outstanding TSN of this path is advanced ('Pseudo CumAck') OR when the earliest retransmitted outstanding TSN of this path is advanced ('RTX Pseudo CumAck').

Section 3.2 of [DBP10a] contains an illustrated example of appropriate congestion window handling for CMT-SCTP.

3.3. Appropriate Delayed Acknowledgements

Standard SCTP [RFC4960] sends a SACK as soon as an out-of-sequence TSN has been received. Delayed Acknowledgements are only allowed if the received TSNs are in sequence. However, due to the load balancing of CMT-SCTP, DATA chunks may overtake each other. This leads to a high number of out-of-sequence TSNs, which have to be acknowledged immediately. Clearly, this behaviour increases the
overhead traffic (usually nearly one SACK chunk for each received packet containing a DATA chunk).

Delayed Acknowledgements for CMT-SCTP are handled as follows:

- In addition to [RFC4960], delaying of SACKs SHOULD *also* be applied for out-of-sequence TSNs.
- A receiver MUST maintain a counter for the number of DATA chunks received before sending a SACK. The value of the counter is stored into each SACK chunk (FIXME: add details; needs reservation of flags bits by IANA). After transmitting a SACK, the counter MUST be reset to 0. Its initial value MUST be 0.
- The SACK handling procedure for a missing TSN M is extended as follows:
  - If all newly acknowledged TSNs have been transmitted over the same path:
    - If there are newly acknowledged TSNs L and H so that L ≤ M ≤ H, the missing count of TSN M SHOULD be incremented by one (like for standard SCTP according to [RFC4960]).
    - Else if all newly acknowledged TSNs N satisfy the condition M ≤ N, the missing count of TSN M SHOULD be incremented by the number of TSNs reported in the SACK chunk.
  - Otherwise (that is, there are newly acknowledged TSNs on different paths), the missing count of TSN M SHOULD be incremented by one (like for standard SCTP according to [RFC4960]).

Section 3.3 of [DBP10a] contains an illustrated example of Delayed Acknowledgements for CMT-SCTP.

4. Non-Renegable SACK

4.1. Negotiation

Before sending/receiving NR-SACKs, both peer endpoints MUST agree on using NR-SACKs. This agreement MUST be negotiated during association establishment. NR-SACK is an extension to the core SCTP, and SCTP extensions that an endpoint supports are reported to the peer endpoint in Supported Extensions Parameter during association establishment (see Section 4.2.7 of [RFC5061].) The Supported Extensions Parameter consists of a list of non-standard Chunk Types.
that are supported by the sender.

An endpoint supporting the NR-SACK extension MUST list the NR-SACK chunk in the Supported Extensions Parameter carried in the INIT or INIT-ACK chunk, depending on whether the endpoint initiates or responds to the initiation of the association. If the NR-SACK chunk type ID is listed in the Chunk Types List of the Supported Extensions Parameter, then the receiving endpoint MUST assume that the NR-SACK chunk is supported by the sending endpoint.

Both endpoints MUST support NR-SACKs for either endpoint to send an NR-SACK. If an endpoint establishes an association with a remote endpoint that does not list NR-SACK in the Supported Extensions Parameter carried in INIT chunk, then both endpoints of the association MUST NOT use NR-SACKs. After association establishment, an endpoint MUST NOT renegotiate the use of NR-SACKs.

Once both endpoints indicate during association establishment that they support the NR-SACK extension, each endpoint SHOULD acknowledge received DATA chunks with NR-SACK chunks, and not SACK chunks. That is, throughout an SCTP association, both endpoints SHOULD send either SACK chunks or NR-SACK chunks, never a mixture of the two.

4.2. The New Chunk Type: Non-Renegable SACK (NR-SACK)

Table 1 illustrates a new chunk type that will be used to transfer NR-SACK information.

<table>
<thead>
<tr>
<th>Chunk Type</th>
<th>Chunk Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x10</td>
<td>Non-Renegable Selective Acknowledgment (NR-SACK)</td>
</tr>
</tbody>
</table>

Table 1: NR-SACK Chunk

As the NR-SACK chunk replaces the SACK chunk, many SACK chunk fields are preserved in the NR-SACK chunk. These preserved fields have the same semantics with the corresponding SACK chunk fields, as defined in [RFC4960], Section 3.3.4. The Gap Ack fields from RFC4960 have been renamed as R Gap Ack to emphasize their renegable nature. Their semantics are unchanged. For completeness, we describe all fields of the NR-SACK chunk, including those that are identical in the SACK chunk.

Similar to the SACK chunk, the NR-SACK chunk is sent to a peer endpoint to (1) acknowledge DATA chunks received in-order, (2) acknowledge DATA chunks received out-of-order, and (3) identify DATA chunks received more than once (i.e., duplicate.) In addition, the NR-SACK chunk (4) informs the peer endpoint of non-renegable out-of-
order DATA chunks.

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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = 0x10 |  Chunk Flags  |         Chunk Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Cumulative TSN Ack                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Advertised Receiver Window Credit (a_rwnd)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Number of R Gap Ack Blocks = N | Number of NR Gap Ack Blocks = M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Number of Duplicate TSNs = X |       Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| R Gap Ack Block #1 Start      |   R Gap Ack Block #1 End      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      R Gap Ack Block #N Start |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      NR Gap Ack Block #1 Start    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   NR Gap Ack Block #M Start   |  NR Gap Ack Block #M End      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Duplicate TSN 1                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Duplicate TSN X                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Type: 8 bits

This field holds the IANA defined chunk type for NR-SACK chunk. The suggested value of this field for IANA is 0x10.

Chunk Flags: 8 bits
Currently not used. It is recommended a sender set all bits to zero on transmit, and a receiver ignore this field.

Chunk Length: 16 bits (unsigned integer) [Same as SACK chunk]

This value represents the size of the chunk in bytes including the Chunk Type, Chunk Flags, Chunk Length, and Chunk Value fields.

Cumulative TSN Ack: 32 bits (unsigned integer) [Same as SACK chunk]

The value of the Cumulative TSN Ack is the last TSN received before a break in the sequence of received TSNs occurs. The next TSN value following the Cumulative TSN Ack has not yet been received at the endpoint sending the NR-SACK.

Advertised Receiver Window Credit (a_rwnd): 32 bits (unsigned integer) [Same as SACK chunk]

Indicates the updated receive buffer space in bytes of the sender of this NR-SACK, see Section 6.2.1 of [RFC4960] for details.

Number of (R)enegable Gap Ack Blocks (N): 16 bits (unsigned integer)

Indicates the number of Renegable Gap Ack Blocks included in this NR-SACK.

Number of (N)on(R)enegable Gap Ack Blocks (M): 16 bits (unsigned integer)

Indicates the number of Non-Renegable Gap Ack Blocks included in this NR-SACK.

Number of Duplicate TSNs (X): 16 bits [Same as SACK chunk]

Contains the number of duplicate TSNs the endpoint has received. Each duplicate TSN is listed following the NR Gap Ack Block list.

Reserved : 16 bits

Currently not used. It is recommended a sender set all bits to zero on transmit, and a receiver ignore this field.

(R)enegable Gap Ack Blocks:

The NR-SACK contains zero or more R Gap Ack Blocks. Each R Gap Ack Block acknowledges a subsequence of renegable out-of-order TSNs. By definition, all TSNs acknowledged by R Gap Ack Blocks are "greater than" the value of the Cumulative TSN Ack.
Because of TSN numbering wraparound, comparisons and all arithmetic operations discussed in this document are based on "Serial Number Arithmetic" as described in Section 1.6 of [RFC4960].

R Gap Ack Blocks are repeated for each R Gap Ack Block up to 'N' defined in the Number of R Gap Ack Blocks field. All DATA chunks with TSNs >= (Cumulative TSN Ack + R Gap Ack Block Start) and <= (Cumulative TSN Ack + R Gap Ack Block End) of each R Gap Ack Block are assumed to have been received correctly, and are renegable.

**R Gap Ack Block Start: 16 bits (unsigned integer)**

Indicates the Start offset TSN for this R Gap Ack Block. This number is set relative to the cumulative TSN number defined in Cumulative TSN Ack field. To calculate the actual start TSN number, the Cumulative TSN Ack is added to this offset number. The calculated TSN identifies the first TSN in this R Gap Ack Block that has been received.

**R Gap Ack Block End: 16 bits (unsigned integer)**

Indicates the End offset TSN for this R Gap Ack Block. This number is set relative to the cumulative TSN number defined in the Cumulative TSN Ack field. To calculate the actual TSN number, the Cumulative TSN Ack is added to this offset number. The calculated TSN identifies the TSN of the last DATA chunk received in this R Gap Ack Block.

**N(on)R(enegable) Gap Ack Blocks:**

The NR-SACK contains zero or more NR Gap Ack Blocks. Each NR Gap Ack Block acknowledges a continuous subsequence of non-renegable out-of-order DATA chunks. If a TSN is nr-gap-acked in any NR-SACK chunk, then all subsequently transmitted NR-SACKs with a smaller cum-ack value than that TSN SHOULD also nr-gap-ack that TSN.

NR Gap Ack Blocks are repeated for each NR Gap Ack Block up to 'M' defined in the Number of NR Gap Ack Blocks field. All DATA chunks with TSNs >= (Cumulative TSN Ack + NR Gap Ack Block Start) and <= (Cumulative TSN Ack + NR Gap Ack Block End) of each NR Gap Ack Block are assumed to be received correctly, and are Non-Renegable.

**NR Gap Ack Block Start: 16 bits (unsigned integer)**

Indicates the Start offset TSN for this NR Gap Ack Block. This number is set relative to the cumulative TSN number defined in Cumulative TSN Ack field. To calculate the actual TSN number, the Cumulative TSN Ack is added to this offset number. The calculated
TSN identifies the first TSN in this NR Gap Ack Block that has been received.

NR Gap Ack Block End: 16 bits (unsigned integer)

Indicates the End offset TSN for this NR Gap Ack Block. This number is set relative to the cumulative TSN number defined in Cumulative TSN Ack field. To calculate the actual TSN number, the Cumulative TSN Ack is added to this offset number. The calculated TSN identifies the TSN of the last DATA chunk received in this NR Gap Ack Block.

Note:

NR Gap Ack Blocks and R Gap Ack Blocks in an NR-SACK chunk SHOULD acknowledge disjoint sets of TSNs. That is, an out-of-order TSN SHOULD be listed in either an R Gap Ack Block or an NR Gap Ack Block, but not the both. R Gap Ack Blocks and NR Gap Ack Blocks together provide the information as do the Gap Ack Block of a SACK chunk, plus additional information about non-renegability.

If all out-of-order data acked by an NR-SACK are renegable, then the Number of NR Gap Ack Blocks MUST be set to 0. If all out-of-order data acked by an NR-SACK are non-renegable, then the Number of R Gap Ack Blocks SHOULD be set to 0. TSNs listed in R Gap Ack Block will be referred as r-gap-acked.

Duplicate TSN: 32 bits (unsigned integer) [Same as SACK chunk]

Indicates a duplicate TSN received since the last NR-SACK was sent. Exactly ‘X’ duplicate TSNs SHOULD be reported where ‘X’ was defined in Number of Duplicate TSNs field.

Each duplicate TSN is listed in this field as many times as the TSN was received since the previous NR-SACK was sent. For example, if a data receiver were to get the TSN 19 three times, the data receiver would list 19 twice in the outbound NR-SACK. After sending the NR-SACK if the receiver received one more TSN 19, the receiver would list 19 as a duplicate once in the next outgoing NR-SACK.

4.3. An Illustrative Example

Assume the following DATA chunks have arrived at the receiver.
The above figure shows the list of DATA chunks at the receiver. TSN denotes the transmission sequence number of the DATA chunk, SID denotes the stream id to which the DATA chunk belongs, SSN denotes the sequence number of the DATA chunk within its stream, and the U bit denotes whether the DATA chunk requires ordered(=0) or unordered(=1) delivery [RFC4960]. Note that TSNs 4, 9, 10, and 12 have not arrived.

This data can be viewed as three separate streams as follows (assume each stream begins with SSN=0). Note that in this example, the application uses stream 2 for unordered data transfer. By definition, SSN fields of unordered DATA chunks are ignored.

Stream-0:
The NR-SACK to acknowledge the above data SHOULD be constructed as follows for each of the three cases described below (the a_rwnd is arbitrarily set to 4000):

CASE-1: Minimal Data Receiver Responsibility - no out-of-order deliverable data yet delivered

None of the deliverable out-of-order DATA chunks have been delivered, and the receiver of the above data does not take responsibility for any of the received out-of-order DATA chunks. The receiver reserves the right to renege any or all of the out-of-order DATA chunks.

```
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+
| Type = 0x10  | 00000000   |      Chunk Length = 32      |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+
|                    Cumulative TSN Ack = 3 |
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+
|                                | a_rwnd = 4000              |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+
|                                | Num of R Gap Ack Blocks = 3 | Num of NR Gap Ack Blocks = 0 |
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+
|                                | Num of Duplicates = 0      |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |                                |
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+
|                                | R Gap Ack Block #1 Start = 2 | R Gap Ack Block #1 End = 5   |
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+
|                                | R Gap Ack Block #2 Start = 8 | R Gap Ack Block #2 End = 8   |
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+
|                                | R Gap Ack Block #3 Start = 10| R Gap Ack Block #3 End = 13  |
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+
```

CASE-2: Minimal Data Receiver Responsibility - all out-of-order
deliverable data delivered

In this case, the NR-SACK chunk is being sent after the data receiver has delivered all deliverable out-of-order DATA chunks to its receiving application (i.e., TSNs 5, 6, 7, 8, 13, and 16.) The receiver reserves the right to renege on all undelivered out-of-order DATA chunks (i.e., TSNs 11, 14, and 15.)

```
+------------------------------+------------------------------+  
| Type = 0x10  |     0x00      |       Chunk Length = 40      |
|------------------------------+------------------------------+  
| Cumulative TSN Ack = 3      |
|------------------------------+------------------------------+  
|                       a_rwnd = 4000 |
|------------------------------+------------------------------+  
| Num of R Gap Ack Blocks = 2 | Num of NR Gap Ack Blocks = 3 |
|------------------------------+------------------------------+  
| Num of Duplicates = 0       | 0x00                         |
|------------------------------+------------------------------+  
| R Gap Ack Block #1 Start = 8 | R Gap Ack Block #1 End = 8   |
|------------------------------+------------------------------+  
| R Gap Ack Block #2 Start = 11| R Gap Ack Block #2 End = 12  |
|------------------------------+------------------------------+  
| NR Gap Ack Block #1 Start = 2| NR Gap Ack Block #1 End = 5  |
|------------------------------+------------------------------+  
| NR Gap Ack Block #2 Start = 10| NR Gap Ack Block #2 End = 10 |
|------------------------------+------------------------------+  
| NR Gap Ack Block #3 Start = 13| NR Gap Ack Block #3 End = 13 |
```

CASE-3: Maximal Data Receiver Responsibility

In this special case, all out-of-order data blocks acknowledged are non-renegable. This case would occur when the data receiver is programmed never to renege, and takes responsibility to deliver all DATA chunks that arrive out-of-order. In this case Num of R Gap Ack Blocks is zero indicating all reported out-of-order TSNs are nr-gap-acked.
4.4. Procedures

The procedures regarding "when" to send an NR-SACK chunk are identical to the procedures regarding when to send a SACK chunk, as outlined in Section 6.2 of [RFC4960].

4.4.1. Sending an NR-SACK chunk

All of the NR-SACK chunk fields identical to the SACK chunk MUST be formed as described in Section 6.2 of [RFC4960].

It is up to the data receiver whether or not to take responsibility for delivery of each out-of-order DATA chunk. An out-of-order DATA chunk that has already been delivered, or that the receiver takes responsibility to deliver (i.e., guarantees not to renege) is Non Renegable (NR), and SHOULD be included in an NR Gap Ack Block field of the outgoing NR-SACK. All other out-of-order data is (R)enegable, and SHOULD be included in R Gap Ack Block field of the outgoing NR-SACK.

Consider three types of data receiver:

CASE-1: Data receiver takes no responsibility for delivery of any out-of-order DATA chunks

CASE-2: Data receiver takes responsibility for all out-of-order DATA chunks that are "deliverable" (i.e., DATA chunks in-sequence within the stream they belong to, or DATA chunks whose (U)nordered bit is 1)
CASE-3: Data receiver takes responsibility for delivery of all out-of-order DATA chunks, whether deliverable or not deliverable

The data receiver SHOULD follow the procedures outlined below for building the NR-SACK.

CASE-1:

1A) Identify the TSNs received out-of-order.

1B) For these out-of-order TSNs, identify the R Gap Ack Blocks. Fill the Number of R Gap Ack Blocks (N) field, R Gap Ack Block #i Start, and R Gap Ack Block #i End where i goes from 1 to N.

1C) Set the Number of NR Gap Ack Blocks (M) field to 0.

CASE-2:

2A) Identify the TSNs received out-of-order.

2B) For the received out-of-order TSNs, check the (U)nordered bit of each TSN. Tag unordered TSNs as NR.

2C) For each stream, also identify the TSNs received out-of-order but are in-sequence within that stream. Tag those in-sequence TSNs as NR.

2D) Tag all out-of-order data that is not NR as (R)enegable.

2E) For those TSNs tagged as (R)enegable, identify the (R)enegable Blocks. Fill the Number of R Gap Ack Blocks(N) field, R Gap Ack Block #i Start, and R Gap Ack Block #i End where i goes from 1 to N.

2F) For those TSNs tagged as NR, identify the NR Blocks. Fill the Number of NR Gap Ack Blocks(M) field, NR Gap Ack Block #i Start, and NR Gap Ack Block #i End where i goes from 1 to M.
CASE-3:

3A) Identify the TSNs received out-of-order. All of these TSNs SHOULD be nr-gap-acked.

3B) Set the Number of R Gap Ack Blocks (N) field to 0.

3C) For these out-of-order TSNs, identify the NR Gap Ack Blocks. Fill the Number of NR Gap Ack Blocks (M) field, NR Gap Ack Block #i Start, and NR Gap Ack Block #i End where i goes from 1 to M.

RFC4960 states that the SCTP endpoint MUST report as many Gap Ack Blocks as can fit in a single SACK chunk limited by the current path MTU. When using NR-SACKs, the SCTP endpoint SHOULD fill as many R Gap Ack Blocks and NR Gap Ack Blocks starting from the Cumulative TSN Ack value as can fit in a single NR-SACK chunk limited by the current path MTU. If space remains, the SCTP endpoint SHOULD fill as many Duplicate TSNs as possible starting from Cumulative TSN Ack value.

4.4.2. Receiving an NR-SACK Chunk

When an NR-SACK chunk is received, all of the NR-SACK fields identical to a SACK chunk SHOULD be processed and handled as in SACK chunk handling outlined in Section 6.2.1 of [RFC4960].

The NR Gap Ack Block Start(s) and NR Gap Ack Block End(s) are offsets relative to the cum-ack. To calculate the actual range of nr-gap-acked TSNs, the cum-ack MUST be added to the Start and End.

For example, assume an incoming NR-SACK chunk’s cum-ack is 12 and an NR Gap Ack Block defines the NR Gap Ack Block Start=5, and the NR Gap Ack Block End=7. This NR Gap Ack block nr-gap-acks TSNs 17 through 19 inclusive.

Upon reception of an NR-SACK chunk, all TSNs listed in either R Gap Ack Block(s) or NR Gap Ack Block(s) SHOULD be processed as would be TSNs included in Gap Ack Block(s) of a SACK chunk. All TSNs in all NR Gap Ack Blocks SHOULD be removed from the data sender’s retransmission queue as their delivery to the receiving application has either already occurred, or is guaranteed by the data receiver.

Although R Gap Ack Blocks and NR Gap Ack Blocks SHOULD be disjoint sets, NR-SACK processing SHOULD work if an NR-SACK chunk has a TSN listed in both an R Gap Ack Block and an NR Gap Ack Block. In this case, the TSN SHOULD be treated as Non-Renegable.
Implementation Note:

Most of NR-SACK processing at the data sender can be implemented by using the same routines as in SACK that process the cum ack and the gap ack(s), followed by removal of nr-gap-acked DATA chunks from the retransmission queue. However, with NR-SACKs, as out-of-order DATA is sometimes removed from the retransmission queue, the gap ack processing routine should recognize that the data sender’s retransmission queue has some transmitted data removed. For example, while calculating missing reports, the gap ack processing routine cannot assume that the highest TSN transmitted is always at the tail (right edge) of the retransmission queue.

5. Buffer Blocking Mitigation

TBD. See [ADB11], [DBR10].

5.1. Sender Buffer Splitting

TBD. See [ADB11], [DBR10].

5.2. Receiver Buffer Splitting

TBD. See [ADB11], [DBR10].

5.3. Problems during Path Failure

This section discusses CMT’s receive buffer related problems during path failure, and proposes a solution for the same.

5.3.1. Problem Description

Link failures arise when a router or a link connecting two routers fails due to link disconnection, hardware malfunction, or software error. Overloaded links caused by flash crowds and denial-of-service (DoS) attacks also degrade end-to-end communication between peer hosts. Ideally, the routing system detects link failures, and in response, reconfigures the routing tables and avoids routing traffic via the failed link. However, existing research highlights problems with Internet backbone routing that result in long route convergence times. The pervasiveness of path failures motivated us to study their impact on CMT, since CMT achieves better throughput via simultaneous data transmission over multiple end-to-end paths.

CMT is an extension to SCTP, and therefore retains SCTP’s failure detection process. A CMT sender uses a tunable failure detection threshold called Path.Max.Retrans (PMR). When a sender experiences
more than PMR consecutive timeouts while trying to reach an active
destination, the destination is marked as failed. With PMR=5, the
failure detection takes 6 consecutive timeouts or 63s. After every
timeout, the CMT sender continues to transmit new data on the failed
path increasing the chances of receive buffer (rbuf) blocking and
degrading CMT performance during permanent and short-term path
failures [NEA08].

5.3.2. Solution: Potentially-failed Destination State

To mitigate the rbuf blocking, we introduce a new destination state
called ’potentially-failed’ state in SCTP (and CMT’s) failure
detection process [I-D.nishida-tsvwg-sctp-failover]. This solution
is based on the rationale that loss detected by a timeout implies
either severe congestion or failure en route. After a single timeout
on a path, a sender is unsure, and marks the corresponding
destination as ’potentially-failed’ (PF). A PF destination is not
used for data transmission or retransmission. CMT’s retransmission
policies are augmented to include the PF state. Performance
evaluations prove that the PF state significantly reduces rbuf
blocking during failure detection [NEA08].

5.4. Non-Renegable SACK

This section discusses problems with SCTP’s SACK mechanism and how it
affects the send buffer and CMT performance.

5.4.1. Problem Description

Gap-acks acknowledge DATA chunks that arrive out-of-order to a
transport layer data receiver. A gap-ack in SCTP is advisory, in
that, while it notifies a data sender about the reception of
indicated DATA chunks, the data receiver is permitted to later
discard DATA chunks that it previously had gap-acked. Discarding a
previously gap-acked DATA chunk is known as ’reneging’. Because of
the possibility of reneging in SCTP, any gap-acked DATA chunk MUST
NOT be removed from the data sender’s retransmission queue until the
DATA chunk is later CumAcked.

Situations exist when a data receiver knows that reneging on a
particular out-of-order DATA chunk will never take place, such as
(but not limited to) after an out-of-order DATA chunk is delivered to
the receiving application. With current SACKs in SCTP, it is not
possible for a data receiver to inform a data sender if or when a
particular out-of-order ’deliverable’ DATA chunk has been ’delivered’
to the receiving application. Thus the data sender MUST keep a copy
of every gap-acked out-of-order DATA chunk(s) in the data sender’s
retransmission queue until the DATA chunk is CumAcked. This use of
the data sender’s retransmission queue is wasteful. The wasted buffer often degrades CMT performance; the degradation increases when a CMT flow traverses via paths with disparate end-to-end properties [NEY08].

5.4.2. Solution: Non-Renegable SACKs

Non-Renegable Selective Acknowledgments (NR-SACKs) Section 4 are a new kind of acknowledgements, extending SCTP’s SACK chunk functionalities. The NR-SACK chunk is an extension of the existing SACK chunk. Several fields are identical, including the Cumulative TSN Ack, the Advertised Receiver Window Credit (a_rwnd), and Duplicate TSNs. These fields have the same semantics as described in [RFC4960].

NR-SACKs also identify out-of-order DATA chunks that a receiver either: (1) has delivered to its receiving application, or (2) takes full responsibility to eventually deliver to its receiving application. These out-of-order DATA chunks are ‘non-renegable.’ Non-Renegable data are reported in the NR Gap Ack Block field of the NR-SACK chunk as described Section 4. We refer to non-renegable selective acknowledgements as ‘nr-gap-acks.’

When an out-of-order DATA chunk is nr-gap-acked, the data sender no longer needs to keep that particular DATA chunk in its retransmission queue, thus allowing the data sender to free up its buffer space sooner than if the DATA chunk were only gap-acked. NR-SACKs improve send buffer utilization and throughput for CMT flows [NEY08].

6. Handling of Shared Bottlenecks

6.1. Introduction

CMT-SCTP assumes all paths to be disjoint. Since each path independently uses a TCP-like congestion control, an SCTP association using N paths over the same bottleneck acquires N times the bandwidth of a concurrent TCP flow. This is clearly unfair. A reliable detection of shared bottlenecks is impossible in arbitrary networks like the Internet. Therefore, [DBA11], [DBP10b] apply the idea of Resource Pooling to CMT-SCTP. Resource Pooling (RP) denotes ‘making a collection of resources behave like a single pooled resource’ [WHB09]. The modifications of RP-enabled CMT-SCTP, further denoted as CMT/RP-SCTP, are described in the following subsections. A detailed description of CMT/RP-SCTP, including congestion control examples, can be found in [DBA11], [DBP10b].
6.2. Initial Values

TDB.

6.3. Congestion Window Growth

TDB. See [DBA11].

6.4. Congestion Window Decrease

TDB. See [DBA11].

7. Chunk Scheduling

TDB. See [DST10].

8. Socket API Considerations

See [I-D.dreibholz-tsvwg-sctpsocket-multipath] and [I-D.dreibholz-tsvwg-sctpsocket-sqinfo].

9. IANA Considerations

[NOTE to RFC-Editor:

"RFCXXXX" is to be replaced by the RFC number you assign this document.
]

[NOTE to RFC-Editor:

The suggested values for the chunk type and the chunk parameter types are tentative and to be confirmed by IANA.
]

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

9.1. A New Chunk Type

A chunk type has to be assigned by IANA. It is suggested to use the values given in Section 4. IANA should assign this value from the
pool of chunks with the upper two bits set to '00'.

This requires an additional line in the "Chunk Types" registry for SCTP:

<table>
<thead>
<tr>
<th>ID</th>
<th>Value</th>
<th>Chunk Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Non-Renegable SACK (NR-SACK)</td>
<td>[RFCXXXX]</td>
<td></td>
</tr>
</tbody>
</table>

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

10. Security Considerations

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

11. Acknowledgments

The authors wish to thank Phillip Conrad, Jonathan Leighton, and Ertugrul Yilmaz for their invaluable comments.

12. References

12.1. Normative References


January 2011.

[I-D.nishida-tsvwg-sctp-failover]

[I-D.dreibholz-tsvwg-sctpsocket-multipath]

[I-D.dreibholz-tsvwg-sctpsocket-sqinfo]
Dreibholz, T., Seggelmann, R., and M. Becke, "Sender Queue Info Option for the SCTP Socket API", draft-dreibholz-tsvwg-sctpsocket-sqinfo-03 (work in progress), March 2012.

12.2. Informative References


[DBP10a]   Dreibholz, T., Becke, M., Pulinthanath, J., and E.


Authors’ Addresses

Paul D. Amer
University of Delaware, Computer and Information Sciences Department
Newark, DE  19716
US
Phone: +1-302-831-1944
Email: amer@cis.udel.edu

Martin Becke
University of Duisburg-Essen, Institute for Experimental Mathematics
Ellernstrasse 29
45326 Essen, Nordrhein-Westfalen
DE
Phone: +49-201-183-7667
Fax: +49-201-183-7673
Email: martin.becke@uni-due.de

Thomas Dreibholz
University of Duisburg-Essen, Institute for Experimental Mathematics
Ellernstrasse 29
45326 Essen, Nordrhein-Westfalen
DE
Phone: +49-201-183-7637
Fax: +49-201-183-7673
Email: dreibh@iem.uni-due.de
URI:  http://www.iem.uni-due.de/~dreibh/

Nasif Ekiz
University of Delaware, Computer and Information Sciences Department
Newark, DE  19716
US
Email: nekiz@udel.edu
Janardhan Iyengar  
Franklin and Marshall College, Mathematics and Computer Science  
PO Box 3003  
Lancaster, Pennsylvania 17604-3003  
US  
Phone: +1-717-358-4774  
Email: jiyengar@fandm.edu  
URI: http://www.fandm.edu/jiyengar/

Preethi Natarajan  
Cisco Systems  
425 East Tasman Drive  
San Jose, California 95134  
US  
Email: prenatar@cisco.com

Randall R. Stewart  
Adara Networks  
Chapin, SC 29036  
US  
Email: randall@lakerest.net

Michael Tuexen  
Muenster University of Applied Sciences  
Stegerwaldstrasse 39  
48565 Steinfurt  
DE  
Email: tuexen@fh-muenster.de