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

B. Briscoe
BT
J. Manner
Aalto University

November 07, 2013

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

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on May 11, 2014.

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

recommendations for the Internet community.

1.1. Terminology and Scoping

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2. Recommendations

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

2.1. Recommendation on Queue Measurement

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

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

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

What this advice means for RED as a specific example:

1. A RED implementation SHOULD use byte mode queue measurement for measuring the congestion of bit-congestible resources and packet mode queue measurement for packet-congestible resources.

2. An implementation SHOULD NOT make it possible to configure the way a queue measures itself, because whether a queue is bit-congestible or packet-congestible is an inherent property of the queue.

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

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

2.2. Recommendation on Encoding Congestion Notification

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

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

What this advice means for RED as a specific example:

1. The RED AQM algorithm SHOULD NOT use byte-mode drop, i.e. it ought to use packet-mode drop. Byte-mode drop is more complex, it creates the perverse incentive to fragment segments into tiny pieces and it is vulnerable to floods of small packets.

2. If a vendor has implemented byte-mode drop, and an operator has turned it on, it is RECOMMENDED to switch it to packet-mode drop, after establishing if there are any implications on the relative performance of applications using different packet sizes. The unlikely possibility of some application-specific legacy use of byte-mode drop is the only reason that all the above recommendations on encoding congestion notification are not phrased more strongly.

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

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

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

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

2.3. Recommendation on Responding to Congestion

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

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

To be clear, the above recommendation solely describes how a transport should interpret the meaning of a congestion indication, as a long term goal. It makes no recommendation on whether a transport should act differently based on this interpretation. It merely aids interoperablity 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 SYNs and pure ACKs.

What this advice means for the case of TCP:

1. If two TCP flows with different packet sizes are required to run at equal bit rates under the same path conditions, this SHOULD be done by altering TCP (Section 4.2.2), not network equipment (the latter affects other transports besides TCP).

2. If it is desired to improve TCP performance by reducing the chance that a SYN or a pure ACK will be dropped, this SHOULD be done by modifying TCP (Section 4.2.3), not network equipment.

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

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

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

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

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

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

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

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

3. Motivating Arguments

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

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

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

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

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

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

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

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

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

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

3.2. Small != Control

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

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

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

3.3. Transport-Independent Network

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

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

- 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

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

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

Table 2 aims to summarise the potential effects of all the advice from different sources. Each column shows a different possible AQM behaviour in different queues in the network, using the terminology of 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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

11.2. Informative References

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

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

Boudec, "Congestion Control for Flows with Variable Packet Size", ACM CCR 34(2) 137--151, 2004, <http://doi.acm.org/10.1145/997150.997162>.

[CHOKe_Var_Pkt] Psounis, K., Pan, R., and B. Prabhaker, "Approximate Fair Dropping for Variable Length Packets", IEEE Micro 21(1):48--56, January-February 2001, <http://www.stanford.edu/~balaji/papers/01approximatefair.pdf}>.

[DRQ] Shin, M., Chong, S., and I. Rhee, "Dual-Resource TCP/AQM for Processing-Constrained Networks", IEEE/ACM Transactions on Networking Vol 16, issue 2, April 2008, <http://dx.doi.org/10.1109/TNET.2007.900415>.

[DupTCP] Wischik, D., "Short messages", Philosphical Transactions of the Royal Society A 366(1872):1941-1953, June 2008, <http://rsta.royalsocietypublishing.org/content/366/1872/1941.full.pdf+html>.

[ECNFixedWireless] Siris, V., "Resource Control for Elastic Traffic in CDMA Networks", Proc. ACM MOBICOM'02 , September 2002, <http://www.ics.forth.gr/netlab/publications/resource_control_elastic_cdma.html>.

[Evol_cc] Gibbens, R. and F. Kelly, "Resource pricing and the evolution of congestion control", Automatica 35(12)1969--1985, December 1999, <http://www.statslab.cam.ac.uk/~frank/evol.html>.

[GentleAggro] Flach, T., Dukkipati, N., Terzis, A., Raghavan, B., Cardwell, N., Cheng, Y., Jain, A., Hao, S., Katz-Bassett, E., and R. Govindan, "Reducing Web Latency: the Virtue of Gentle Aggression", ACM SIGCOMM CCR 43(4)159--170, August 2013, <http://doi.acm.org/10.1145/2486001.2486014>.

[I-D.nichols-tsvwg-codel] Nichols, K. and V. Jacobson, "Controlled Delay Active Queue Management",

draft-nichols-tsvwg-codel-01 (work in progress), February 2013.

[I-D.pan-tsvwg-pie] Pan, R., Natarajan, P., Piglione, C., and M. Prabhu, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", draft-pan-tsvwg-pie-00 (work in progress), December 2012.

[IOSArch] Bollapragada, V., White, R., and C. Murphy, "Inside Cisco IOS Software Architecture", Cisco Press: CCIE Professional Development ISBN13: 978-1-57870-181-0, July 2000.

[PktSizeEquCC] Vasallo, P., "Variable Packet Size Equation-Based Congestion Control", ICSI Technical Report tr-00-008, 2000, <http://http.icsi.berkeley.edu/ftp/global/pub/techreports/2000/tr-00-008.pdf>.

[RED93] Floyd, S. and V. Jacobson, "Random Early Detection (RED) gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking 1(4) 397--413, August 1993, <http://www.icir.org/floyd/papers/red/red.html>.

[REDbias] Eddy, W. and M. Allman, "A Comparison of RED's Byte and Packet Modes", Computer Networks 42(3) 261--280, June 2003, <http://www.ir.bbn.com/documents/articles/redbias.ps>.

[REDbyte] De Cnodder, S., Elloumi, O., and K. Pauwels, "RED behavior with different packet sizes", Proc. 5th IEEE Symposium on Computers and Communications (ISCC) 793--799, July 2000, <http://www.icir.org/floyd/red/Elloumi99.pdf>.

[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet",

RFC 2309, April 1998.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC3426] Floyd, S., "General Architectural and Policy Considerations", RFC 3426, November 2002.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3714] Floyd, S. and J. Kempf, "IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet", RFC 3714, March 2004.

[RFC4828] Floyd, S. and E. Kohler, "TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant", RFC 4828, April 2007.

[RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, September 2008.

[RFC5562] Kuzmanovic, A., Mondal, A., Floyd, S., and K. Ramakrishnan, "Adding Explicit Congestion Notification (ECN) Capability to TCP's SYN/ACK Packets", RFC 5562, June 2009.

[RFC5670] Eardley, P., "Metering and Marking Behaviour of PCN-Nodes", RFC 5670, November 2009.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.

[RFC5690] Floyd, S., Arcia, A., Ros, D., and J. Iyengar, "Adding Acknowledgement Congestion Control to TCP", RFC 5690, February 2010.

[RFC6077] Papadimitriou, D., Welzl, M., Scharf, M., and B. Briscoe, "Open Research Issues in Internet Congestion Control", RFC 6077, February 2011.

[RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, August 2012.

[RFC6789] Briscoe, B., Woundy, R., and A. Cooper, "Congestion Exposure (ConEx) Concepts and Use Cases", RFC 6789, December 2012.

[Rate_fair_Dis] Briscoe, B., "Flow Rate Fairness: Dismantling a Religion", ACM CCR 37(2)63--74, April 2007, <http://portal.acm.org/citation.cfm?id=1232926>.

[gentle_RED] Floyd, S., "Recommendation on using the "gentle_" variant of RED", Web page , March 2000, <http://www.icir.org/floyd/red/gentle.html>.

[pBox] Floyd, S. and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE/ACM Transactions on Networking 7(4) 458--472, August 1999, <http://www.aciri.org/floyd/end2end-paper.html>.

[pktByteEmail] Floyd, S., "RED: Discussions of Byte and Packet Modes", email , March 1997, <http://www-nrg.ee.lbl.gov/floyd/REDaveraging.txt>.

Appendix A. Survey of RED Implementation Status

This Appendix is informative, not normative.

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

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

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

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

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

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

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

Appendix B. Sufficiency of Packet-Mode Drop

This Appendix is informative, not normative.

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

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

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

Table 4: Four Possible Congestion Scenarios

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

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

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

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

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

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

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

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

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

We will set aside the last column until later. The columns labelled "Flow 1" and "Flow 2" compare two flows consisting of 60B and 1500B packets respectively. The body of the table considers two separate cases, one where the flows have equal bit-rate and the other with equal packet-rates. In both cases, the two flows fill a 96Mbps link. Therefore, in the equal bit-rate case they each have half the bit-rate (48Mbps). Whereas, with equal packet-rates, flow 1 uses 25 times smaller packets so it gets 25 times less bit-rate--it only gets 1/(1+25) of the link capacity (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".

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

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

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

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

The explanation so far has assumed that each flow consists of packets of only one constant size. Nonetheless, it extends naturally to flows with mixed packet sizes. In the right-most column of Table 5 a flow of mixed size packets is created simply by considering flow 1 and flow 2 as a single aggregated flow. There is no need for a flow to maintain an average packet size. It is only necessary for the transport to scale its response to each congestion indication by the size of each individual lost (or marked) packet. Taking for example the "Equal packet-rate" case, in one second about 8 small packets and 8 large packets are lost (making closer to 15 than 16 losses per second due to rounding). If the transport multiplies each loss by its size, in one second it responds to 8*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(u1*s1+ u2*s2).

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

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

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

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

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

1. A packet-congestible resource trying to code congestion level p_p into a packet stream should mark the idealised 'packet congestion' field in each packet with probability p_p irrespective of the packet's size. The transport should then take a packet with the packet congestion field marked to mean just one mark, irrespective of the packet size.

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

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

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

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

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

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

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

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

Appendix C. Byte-mode Drop Complicates Policing Congestion Response

This section is informative, not normative.

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

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

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

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

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

Appendix D. Changes from Previous Versions

To be removed by the RFC Editor on publication.

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

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

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

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

- appended "Similarly applications, over non-TCP transports could make any packets that are effectively control packets more robust by using Diffserv, data duplication, FEC etc."

+ Updated Wischik ref and added "Reducing Web Latency: the Virtue of Gentle Aggression" ref.

* Expanded more abbreviations (CoDel, PIE, MTU).

* Section 1. Intro [Stephen Farrell]:

  + In the places where the doc desribes the dichotomy between 'long-term goal' and 'expediency' the words long term goal and expedient have been introduced, to more explicitly refer back to this introductory para (S.2.1 & S.2.3).

  + Added explanation of what scaling with packet size means.

* Conclusions [Benoit Claise]:

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

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

* Abstract: clarified that advice applies to all AQMs including newer ones

* Abstract & Intro: changed 'read' to 'detect', because you don't read losses, you detect them.

* S.1. Introduction: Disambiguated summary of advice on queue measurement.

* Clarified that the doc deprecates any preference based solely on packet size, it's not only against preferring smaller packets.

* S.4.1.2. Congestion Measurement without a Queue: Explained that a queue of TXOPs represents a queue into spectrum congested by too many bits.

* S.5.2: Bit- & Packet-congestible Network: Referred to explanation in S.4.1.2 to make the point that TXOPs are not a primary unit of workload like bits and packets are, even though you get queues of TXOPs.

* 6. Security: Disambiguated 'bias towards'.

* 8. Conclusions: Made consistent with recommendation to use time if possible for queue measurement.

From -09 to -10: Following IESG review:

* Updates 2309: Left header unchanged reflecting eventual IESG consensus [Sean Turner, Pete Resnick].

* S.1 Intro: This memo adds to the congestion control principles enumerated in BCP 41 [Pete Resnick]

* Abstract, S.1, S.1.1, s.1.2 Intro, Scoping and Example: Made applicability to all AQMs clearer listing some more example AQMs and explained that we always use RED for examples, but this doesn't mean it's not applicable to other AQMs. [A number of reviewers have described the draft as "about RED"]

* S.1 & S.2.1 Queue measurement: Explained that the choice between measuring the queue in packets or bytes is only relevant if measuring it in time units is infeasible [So as not to imply that we haven't noticed the advances made by PDPC & CoDel]

* S.1.1. Terminology: Better explained why hybrid systems congested by both packets and bytes are often designed to be treated as bit-congestible [Richard Barnes].

* S.2.1. Queue measurement advice: Added examples. Added a counter-example to justify SHOULDs rather than MUSTs. Pointed to S.4.1 for a list of more complicated scenarios. [Benson

Schliesser, OpsDir]

* S2.2. Recommendation on Encoding Congestion Notification: Removed SHOULD treat packets equally, leaving only SHOULD NOT drop dependent on packet size, to avoid it sounding like we're saying QoS is not allowed. Pointed to possible app-specific legacy use of byte-mode as a counter-example that prevents us saying MUST NOT. [Pete Resnick]

* S.2.3. Recommendation on Responding to Congestion: capitalised the two SHOULDs in recommendations for TCP, and gave possible counter-examples. [noticed while dealing with Pete Resnick's point]

* S2.4. Splitting & Merging: RTCP -> RTP/RTCP [Pete McCann, Gen-ART]

* S.3.2 Small != Control: many control packets are small -> ...tend to be small [Stephen Farrell]

* S.3.1 Perverse incentives: Changed transport designers to app developers [Stephen Farrell]

* S.4.1.1. Fixed Size Packet Buffers: Nearly completely re-written to simplify and to reverse the advice when the underlying resource is bit-congestible, irrespective of whether the buffer consists of fixed-size packet buffers. [Richard Barnes & Benson Schliesser]

* S.4.2.1.2. Packet Size Bias Regardless of AQM: Largely re-written to reflect the earlier change in advice about fixed-size packet buffers, and to primarily focus on getting rid of tail-drop, not various nuances of tail-drop. [Richard Barnes & Benson Schliesser]

* Editorial corrections [Tim Bray, AppsDir, Pete McCann, Gen-ART and others]

* Updated refs (two I-Ds have become RFCs). [Pete McCann]

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

* S.2.1: Made RED-related queue measurement recommendations clearer

* S.2.3: Added to "Recommendation on Responding to Congestion" to make it clear that we are definitely not saying transports have to equalise bit-rates, just how to do it and not do it, if you

want to.

* S.3: Clarified motivation sections S.3.3 "Transport-Independent Network" and S.3.5 "Implementation Efficiency"

* S.3.4: Completely changed motivating argument from "Scaling Congestion Control with Packet Size" to "Partial Deployment of AQM".

From -07 to -08:

* Altered abstract to say it provides best current practice and highlight that it updates RFC2309

* Added null IANA section

* Updated refs

From -06 to -07:

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

From -05 to -06:

* Primarily editorial fixes.

From -04 to -05:

* Changed from Informational to BCP and highlighted non-normative sections and appendices

* Removed language about consensus

* Added "Example Comparing Packet-Mode Drop and Byte-Mode Drop"

* Arranged "Motivating Arguments" into a more logical order and completely rewrote "Transport-Independent Network" & "Scaling Congestion Control with Packet Size" arguments. Removed "Why Now?"

* Clarified applicability of certain recommendations

* Shifted vendor survey to an Appendix

* Cut down "Outstanding Issues and Next Steps"

* Re-drafted the start of the conclusions to highlight the three distinct areas of concern

* Completely re-wrote appendices

* Editorial corrections throughout.

From -03 to -04:

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

From -02 to -03 (this version)

* Structural changes:

  + Split off text at end of "Scaling Congestion Control with Packet Size" into new section "Transport-Independent Network"

  + Shifted "Recommendations" straight after "Motivating Arguments" and added "Conclusions" at end to reinforce Recommendations

  + Added more internal structure to Recommendations, so that recommendations specific to RED or to TCP are just corollaries of a more general recommendation, rather than being listed as a separate recommendation.

  + Renamed "State of the Art" as "Critical Survey of Existing Advice" and retitled a number of subsections with more descriptive titles.

  + Split end of "Congestion Coding: Summary of Status" into a new subsection called "RED Implementation Status".

  + Removed text that had been in the Appendix "Congestion Notification Definition: Further Justification".

* Reordered the intro text a little.

* Made it clearer when advice being reported is deprecated and when it is not.

* Described AQM as in network equipment, rather than saying "at the network layer" (to side-step controversy over whether functions like AQM are in the transport layer but in network

equipment).

* Minor improvements to clarity throughout

From -01 to -02:

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

From -00 to -01:

* Minor clarifications throughout and updated references

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

* Added note on relationship to existing RFCs

* Posed the question of whether packet-congestion could become common and deferred it to the IRTF ICCRG. Added ref to the dual-resource queue (DRQ) proposal.

* Changed PCN references from the PCN charter & architecture to the PCN marking behaviour draft most likely to imminently become the standards track WG item.

From -01 to -02:

* Abstract reorganised to align with clearer separation of issue in the memo.

* Introduction reorganised with motivating arguments removed to new Section 3.

* Clarified avoiding lock-out of large packets is not the main or only motivation for RED.

* Mentioned choice of drop or marking explicitly throughout, rather than trying to coin a word to mean either.

* Generalised the discussion throughout to any packet forwarding function on any network equipment, not just routers.

* Clarified the last point about why this is a good time to sort out this issue: because it will be hard / impossible to design new transports unless we decide whether the network or the transport is allowing for packet size.

* Added statement explaining the horizon of the memo is long term, but with short term expediency in mind.

* Added material on scaling congestion control with packet size (Section 3.4).

* Separated out issue of normalising TCP's bit rate from issue of preference to control packets (Section 3.2).

* Divided up Congestion Measurement section for clarity, including new material on fixed size packet buffers and buffer carving (Section 4.1.1 & Section 4.2.1) and on congestion measurement in wireless link technologies without queues (Section 4.1.2).

* Added section on 'Making Transports Robust against Control Packet Losses' (Section 4.2.3) with existing & new material included.

* Added tabulated results of vendor survey on byte-mode drop variant of RED (Table 3).

From -00 to -01:

* Clarified applicability to drop as well as ECN.

* Highlighted DoS vulnerability.

* Emphasised that drop-tail suffers from similar problems to byte-mode drop, so only byte-mode drop should be turned off, not RED itself.

* Clarified the original apparent motivations for recommending byte-mode drop included protecting SYNs and pure ACKs more than equalising the bit rates of TCPs with different segment sizes. Removed some conjectured motivations.

* Added support for updates to TCP in progress (ackcc & ecn-syn-ack).

* Updated survey results with newly arrived data.

* Pulled all recommendations together into the conclusions.

* Moved some detailed points into two additional appendices and a note.

* Considerable clarifications throughout.

* Updated references

Authors' Addresses

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

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

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

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

Transport Area Working Group
Internet-Draft
Updates: 2780, 2782, 3828, 4340,
4960, 5595 (if approved)
Intended status: BCP
Expires: August 16, 2011

M. Cotton
ICANN
L. Eggert
Nokia
J. Touch
USC/ISI
M. Westerlund
Ericsson
S. Cheshire
Apple

February 12, 2011

Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry
draft-ietf-tsvwg-iana-ports-10

Abstract

This document defines the procedures that the Internet Assigned Numbers Authority (IANA) uses when handling assignment and other requests related to the Service Name and Transport Protocol Port Number Registry. It also discusses the rationale and principles behind these procedures and how they facilitate the long-term sustainability of the registry.

This document updates IANA's procedures by obsoleting the previous UDP and TCP port assignment procedures defined in Sections 8 and 9.1 of the IANA allocation guidelines [RFC2780], and it updates the IANA Service Name and Port assignment procedures for UDP-Lite [RFC3828], DCCP [RFC4340] [RFC5595] and SCTP [RFC4960]. It also updates the DNS SRV specification [RFC2782] to clarify what a service name is and how it is registered.

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 16, 2011.

Table of Contents

1. Introduction

For many years, the assignment of new service names and port number values for use with the Transmission Control Protocol (TCP) [RFC0793] and the User Datagram Protocol (UDP) [RFC0768] have had less than clear guidelines. New transport protocols have been added - the Stream Control Transmission Protocol (SCTP) [RFC4960] and the Datagram Congestion Control Protocol (DCCP) [RFC4342] - and new mechanisms like DNS SRV records [RFC2782] have been developed, each with separate registries and separate guidelines. The community also recognized the need for additional procedures beyond just assignment; notably modification, revocation, and release.

A key element of the procedural streamlining specified in this document is to establish identical assignment procedures for all IETF transport protocols. This document brings the IANA procedures for TCP and UDP in line with those for SCTP and DCCP, resulting in a single process that requesters and IANA follow for all requests for all transport protocols, including future protocols not yet defined.

In addition to detailing the IANA procedures for the initial assignment of service names and port numbers, this document also specifies post-assignment procedures that until now have been handled in an ad hoc manner. These include procedures to de-assign a port number that is no longer in use, to take a port number assigned for one service that is no longer in use and reuse it for another service, and the procedure by which IANA can unilaterally revoke a prior port number assignment. Section 8 discusses the specifics of these procedures and processes that requesters and IANA follow for all requests for all current and future transport protocols.

IANA is the authority for assigning service names and port numbers. The registries that are created to store these assignments are maintained by IANA. For protocols developed by IETF working groups, IANA now also offers a method for the "early assignment" [RFC4020] of service names and port numbers, as described in Section 8.1.

This document updates IANA's procedures for UDP and TCP port numbers by obsoleting Sections 8 and 9.1 of the IANA assignment guidelines [RFC2780]. (Note that other sections of the IANA assignment guidelines, relating to the protocol field values in IPv4 headers, were also updated in February 2008 [RFC5237].) This document also updates the IANA assignment procedures for DCCP [RFC4340] [RFC5595]and SCTP [RFC4960].

The Lightweight User Datagram Protocol (UDP-Lite) shares the port space with UDP. The UDP-Lite specification [RFC3828] says: "UDP-Lite uses the same set of port number values assigned by the IANA for use

by UDP". An update of the UDP procedures therefore also results in a corresponding update of the UDP-Lite procedures.

This document also clarifies what a service name is and how it is assigned. This will impact the DNS SRV specification [RFC2782], because that specification merely makes a brief mention that the symbolic names of services are defined in "Assigned Numbers" [RFC1700], without stating to which section it refers within that 230-page document. The DNS SRV specification may have been referring to the list of Port Assignments (known as /etc/services on Unix), or to the "Protocol And Service Names" section, or to both, or to some other section. Furthermore, "Assigned Numbers" [RFC1700] has been obsoleted [RFC3232] and has been replaced by on-line registries [PORTREG][PROTSERVREG].

The development of new transport protocols is a major effort that the IETF does not undertake very often. If a new transport protocol is standardized in the future, it is expected to follow these guidelines and practices around using service names and port numbers as much as possible, for consistency.

2. Motivation

Information about the assignment procedures for the port registry has existed in three locations: the forms for requesting port number assignments on the IANA web site [SYSFORM][USRFORM], an introductory text section in the file listing the port number assignments themselves (known as the port numbers registry) [PORTREG], and two brief sections of the IANA Allocation Guidelines [RFC2780].

Similarly, the procedures surrounding service names have been historically unclear. Service names were originally created as mnemonic identifiers for port numbers without a well-defined syntax, apart from the 14-character limit mentioned on the IANA website [SYSFORM][USRFORM]. Even that length limit has not been consistently applied, and some assigned service names are 15 characters long. When service identification via DNS SRV Resource Records (RRs) was introduced [RFC2782], it became useful to start assigning service names alone, and because IANA had no procedure for assigning a service name without an associated port number, this lead to the creation of an informal temporary service name registry outside of the control of IANA, which now contains roughly 500 service names [SRVREG].

This document aggregates all this scattered information into a single reference that aligns and clearly defines the management procedures for both service names and port numbers. It gives more detailed

guidance to prospective requesters of service names and ports than the existing documentation, and it streamlines the IANA procedures for the management of the registry, so that requests can be completed in a timely manner.

This document defines rules for assignment of service names without associated port numbers, for such usages as DNS SRV records [RFC2782], which was not possible under the previous IANA procedures. The document also merges service name assignments from the non-IANA ad hoc registry [SRVREG] and from the IANA "Protocol and Service Names" registry [PROTSERVREG] into the IANA "Service Name and Transport Protocol Port Number" registry [PORTREG], which from here on is the single authoritative registry for service names and port numbers.

An additional purpose of this document is to describe the principles that guide the IETF and IANA in their role as the long-term joint stewards of the service name and port number registry. TCP and UDP have had remarkable success over the last decades. Thousands of applications and application-level protocols have service names and port numbers assigned for their use, and there is every reason to believe that this trend will continue into the future. It is hence extremely important that management of the registry follow principles that ensure its long-term usefulness as a shared resource. Section 7 discusses these principles in detail.

3. Background

The Transmission Control Protocol (TCP) [RFC0793] and the User Datagram Protocol (UDP) [RFC0768] have enjoyed a remarkable success over the decades as the two most widely used transport protocols on the Internet. They have relied on the concept of "ports" as logical entities for Internet communication. Ports serve two purposes: first, they provide a demultiplexing identifier to differentiate transport sessions between the same pair of endpoints, and second, they may also identify the application protocol and associated service to which processes connect. Newer transport protocols, such as the Stream Control Transmission Protocol (SCTP) [RFC4960] and the Datagram Congestion Control Protocol (DCCP) [RFC4342] have also adopted the concept of ports for their communication sessions and use 16-bit port numbers in the same way as TCP and UDP (and UDP-Lite [RFC3828], a variant of UDP).

Port numbers are the original and most widely used means for application and service identification on the Internet. Ports are 16-bit numbers, and the combination of source and destination port numbers together with the IP addresses of the communicating end

systems uniquely identifies a session of a given transport protocol. Port numbers are also known by their associated service names such as "telnet" for port number 23 and "http" (as well as "www" and "www-http") for port number 80.

Hosts running services, hosts accessing services on other hosts, and intermediate devices (such as firewalls and NATs) that restrict services need to agree on which service corresponds to a particular destination port. Although this is ultimately a local decision with meaning only between the endpoints of a connection, it is common for many services to have a default port upon which those servers usually listen, when possible, and these ports are recorded by the Internet Assigned Numbers Authority (IANA) through the service name and port number registry [PORTREG].

Over time, the assumption that a particular port number necessarily implies a particular service may become less true. For example, multiple instances of the same service on the same host cannot generally listen on the same port, and multiple hosts behind the same NAT gateway cannot all have a mapping for the same port on the external side of the NAT gateway, whether using static port mappings configured by hand by the user, or dynamic port mappings configured automatically using a port mapping protocol like NAT Port Mapping Protocol (NAT-PMP) [I-D.cheshire-nat-pmp] or Internet Gateway Device (IGD) [IGD].

Applications may use port numbers directly, look up port numbers based on service names via system calls such as getservbyname() on UNIX, look up port numbers by performing queries for DNS SRV records [RFC2782][I-D.cheshire-dnsext-dns-sd], or determine port numbers in a variety of other ways like the TCP Port Service Multiplexer (TCPMUX) [RFC1078].

Designers of applications and application-level protocols may apply to IANA for an assigned service name and port number for a specific application, and may - after assignment - assume that no other application will use that service name or port number for its communication sessions. Application designers also have the option of requesting only an assigned service name without a corresponding fixed port number if their application does not require one, such as applications that use DNS SRV records to look up port numbers dynamically at runtime. Because the port number space is finite (and therefore conservation is an important goal) the alternative of using service names instead of port numbers is RECOMMENDED whenever possible.

4. Conventions Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

This document uses the term "assignment" to refer to the procedure by which IANA provides service names and/or port numbers to requesting parties; other RFCs refer to this as "allocation" or "registration". This document assumes that all these terms have the same meaning, and will use terms other than "assignment" when quoting from or referring to text in these other documents.

5. Service Names

Service names are the unique key in the Service Name and Transport Protocol Port Number Registry. This unique symbolic name for a service may also be used for other purposes, such as in DNS SRV records [RFC2782]. Within the registry, this unique key ensures that different services can be unambiguously distinguished, thus preventing name collisions and avoiding confusion about who is the Assignee for a particular entry.

There may be more than one service name associated with a particular transport protocol and port. There are three ways that such port number overloading can occur:

- Overloading occurs when one service is an extension of another service, and an in-band mechanism exists for determining if the extension is present or not. One example is port 3478, which has the service name aliases "stun" and "turn". TURN [RFC5766] is an extension to the STUN [RFC5389] service. TURN-enabled clients wishing to locate TURN servers could attempt to discover "stun" services and then check in-band if the server also supports TURN, but this would be inefficient. Enabling them to directly query for "turn" servers by name is a better approach. (Note that TURN servers in this case should also be locatable via a "stun" discovery, because every TURN server is also a STUN server.)

- By historical accident, the service name "http" has two synonyms "www" and "www-http". When used in SRV records [RFC2782] and similar service discovery mechanisms, only the service name "http" should be used, not these additional names. If a server were to advertise "www", it would not be discovered by clients browsing for "http". Advertising or browsing for the aliases as well as the primary service name is inefficient, and achieves nothing that

is not already achieved by using the service name "http" exclusively.

- As indicated in this document in Section 10.1, overloading has been used to create replacement names that are consistent with the syntax this document prescribes for legacy names that do not conform to this syntax already. For such cases, only the new name should be used in SRV records, to avoid the same issues as with historical cases of multiple names, and also because the legacy names are incompatible with SRV record use.

Assignment requests for new names for existing registered services will be rejected, as a result. Implementers are requested to inform IANA if they discover other cases where a single service has multiple names, so that one name may be recorded as the primary name for service discovery purposes.

Service names are assigned on a "first come, first served" basis, as described in Section 8.1. Names should be brief and informative, avoiding words or abbreviations that are redundant in the context of the registry (e.g., "port", "service", "protocol", etc.) Names referring to discovery services, e.g., using multicast or broadcast to identify endpoints capable of a given service, SHOULD use an easily identifiable suffix (e.g., "-disc").

5.1. Service Name Syntax

Valid service names are hereby normatively defined as follows:

- MUST be at least 1 character and no more than 15 characters long
- MUST contain only US-ASCII [ANSI.X3-4.1986] letters 'A' - 'Z' and 'a' - 'z', digits '0' - '9', and hyphens ('-', ASCII 0x2D or decimal 45)
- MUST contain at least one letter ('A' - 'Z' or 'a' - 'z')
- MUST NOT begin or end with a hyphen
- hyphens MUST NOT be adjacent to other hyphens

The reason for requiring at least one letter is to avoid service names like "23" (could be confused with a numeric port) or "6000-6063" (could be confused with a numeric port range). Although service names may contain both upper-case and lower-case letters, case is ignored for comparison purposes, so both "http" and "HTTP" denote the same service.

Service names are purely opaque identifiers, and no semantics are implied by any superficial structure that a given service name may appear to have. For example, a company called "Example" may choose to register service names "Example-Foo" and "Example-Bar" for its "Foo" and "Bar" products, but the "Example" company cannot claim to "own" all service names beginning with "Example-"; they cannot prevent someone else from registering "Example-Baz" for a different service, and they cannot prevent other developers from using the "Example-Foo" and "Example-Bar" service types in order to interoperate with the "Foo" and "Bar" products. Technically speaking, in service discovery protocols, service names are merely a series of byte values on the wire; for the mnemonic convenience of human developers it can be convenient to interpret those byte values as human-readable ASCII characters, but software should treat them as purely opaque identifiers and not attempt to parse them for any additional embedded meaning.

In approximately 98% of cases, the new "service name" is exactly the same as the old historic "short name" from the IANA web forms [SYSFORM] [USRFORM]. In approximately 2% of cases, the new "service name" is derived from the old historic "short name" as described below in Section 10.1.

The rules for valid service names, excepting the limit of 15 characters maximum, are also expressed below (as a non-normative convenience) using ABNF [RFC5234].

```
SRVNAME = *(1*DIGIT [HYPHEN]) ALPHA *([HYPHEN] ALNUM)
ALNUM   = ALPHA / DIGIT     ; A-Z, a-z, 0-9
HYPHEN  = %x2D              ; "-"
ALPHA   = %x41-5A / %x61-7A ; A-Z / a-z [RFC5234]
DIGIT   = %x30-39           ; 0-9       [RFC5234]
```

5.2. Service Name Usage in DNS SRV Records

The DNS SRV specification [RFC2782] states that the Service Label part of the owner name of a DNS SRV record includes a "Service" element, described as "the symbolic name of the desired service", but as discussed above, it is not clear precisely what this means.

This document clarifies that the Service Label MUST be a service name as defined herein with an underscore prepended. The service name SHOULD be registered with IANA and recorded in the Service Name and Transport Protocol Port Number Registry [PORTREG].

The details of using Service Names in SRV Service Labels are

specified in the DNS SRV specification [RFC2782].

6. Port Number Ranges

TCP, UDP, UDP-Lite, SCTP and DCCP use 16-bit namespaces for their port number registries. The port registries for all of these transport protocols are subdivided into three ranges of numbers [RFC1340], and Section 8.1.2 describes the IANA procedures for each range in detail:

- the System Ports, also known as the Well Known Ports, from 0-1023 (assigned by IANA)
- the User Ports, also known as the Registered Ports, from 1024-49151 (assigned by IANA)
- the Dynamic Ports, also known as the Private or Ephemeral Ports, from 49152-65535 (never assigned)

Of the assignable port ranges (System Ports and User Ports, i.e., port numbers 0-49151), individual port numbers are in one of three states at any given time:

- Assigned: Assigned port numbers are currently assigned to the service indicated in the registry.
- Unassigned: Unassigned port numbers are currently available for assignment upon request, as per the procedures outlined in this document.
- Reserved: Reserved port numbers are not available for regular assignment; they are "assigned to IANA" for special purposes. Reserved port numbers include values at the edges of each range, e.g., 0, 1023, 1024, etc., which may be used to extend these ranges or the overall port number space in the future.

In order to keep the size of the registry manageable, IANA typically only records the Assigned and Reserved service names and port numbers in the registry. Unassigned values are typically not explicitly listed. (There are very many Unassigned service names and enumerating them all would not be practical.)

As a data point, when this document was written, approximately 76% of the TCP and UDP System Ports were assigned, and approximately 9% of the User Ports were assigned. (As noted, Dynamic Ports are never assigned.)

6.1. Service names and Port Numbers for Experimentation

Of the System Ports, two TCP and UDP port numbers (1021 and 1022), together with their respective service names ("exp1" and "exp2"), have been assigned for experimentation with new applications and application-layer protocols that require a port number in the assigned ports range [RFC4727].

Please refer to Sections 1 and 1.1 of "Assigning Experimental and Testing Numbers Considered Useful" [RFC3692] for how these experimental port numbers are to be used.

This document assigns the same two service names and port numbers for experimentation with new application-layer protocols over SCTP and DCCP in Section 10.2.

Unfortunately, it can be difficult to limit access to these ports. Users SHOULD take measures to ensure that experimental ports are connecting to the intended process. For example, users of these experimental ports might include a 64-bit nonce, once on each segment of a message-oriented channel (e.g., UDP), or once at the beginning of a byte-stream (e.g., TCP), which is used to confirm that the port is being used as intended. Such confirmation of intended use is especially important when these ports are associated with privileged (e.g., system or administrator) processes.

7. Principles for Service Name and Transport Protocol Port Number Registry Management

Management procedures for the service name and transport protocol port number registry include assignment of service names and port numbers upon request, as well as management of information about existing assignments. The latter includes maintaining contact and description information about assignments, revoking abandoned assignments, and redefining assignments when needed. Of these procedures, careful port number assignment is most critical, in order to continue to conserve the remaining port numbers.

As noted earlier, only about 9% of the User Port space is currently assigned. The current rate of assignment is approximately 400 ports per year, and has remained steady for the past 8 years. At that rate, if similar conservation continues, this resource will sustain another 85 years of assignment - without the need to resort to reassignment of released values or revocation. The namespace available for service names is much larger, which allows for simpler management procedures.

7.1. Past Principles

The principles for service name and port number management are based on the recommendations of the IANA "Expert Review" team. Until recently, that team followed a set of informal guidelines developed based on the review experience from previous assignment requests. These original guidelines, although informal, had never been publicly documented. They are recorded here for historical purposes only; the current guidelines are described in Section 7.2. These guidelines previously were:

- TCP and UDP ports were simultaneously assigned when either was requested
- Port numbers were the primary assignment; service names were informative only, and did not have a well-defined syntax
- Port numbers were conserved informally, and sometimes inconsistently (e.g., some services were assigned ranges of many port numbers even where not strictly necessary)
- SCTP and DCCP service name and port number registries were managed separately from the TCP/UDP registries
- Service names could not be assigned in the old ports registry without assigning an associated port number at the same time

7.2. Updated Principles

This section summarizes the current principles by which IANA both handles the Service Name and Transport Protocol Port Number Registry and attempts to conserve the port number space. This description is intended to inform applicants requesting service names and port numbers. IANA has flexibility beyond these principles when handling assignment requests; other factors may come into play, and exceptions may be made to best serve the needs of the Internet. Applicants should be aware that IANA decisions are not required to be bound to these principles. These principles and general advice to users on port use are expected to change over time and are therefore documented separately, please see [I-D.touch-tsvwg-port-use].

IANA strives to assign service names that do not request an associated port number assignment under a simple "First Come, First Served" policy [RFC5226]. IANA MAY, at its discretion, refer service name requests to "Expert Review" in cases of mass assignment requests or other situations where IANA believes expert review is advisable [RFC5226]; use of the "Expert Review" helps advise IANA informally in cases where "IETF Review" or "IESG Review" is used, as with most IETF

protocols.

The basic principle of service name and port number registry management is to conserve use of the port space where possible. Extensions to support larger port number spaces would require changing many core protocols of the current Internet in a way that would not be backward compatible and interfere with both current and legacy applications.

Conservation of the port number space is required because this space is a limited resource, so applications are expected to participate in the traffic demultiplexing process where feasible. The port numbers are expected to encode as little information as possible that will still enable an application to perform further demultiplexing by itself. In particular, the principles form a goal that IANA strives to achieve for new applications (with exceptions as deemed appropriate, especially as for extensions to legacy services) as follows:

- IANA strives to assign only one assigned port number per service or application
- IANA strives to assign only one assigned port number for all variants of a service (e.g., for updated versions of a service)
- IANA strives to encourage the deployment of secure protocols
- IANA strives to assign only one assigned port number for all different types of device using or participating in the same service
- IANA strives to assign port numbers only for the transport protocol(s) explicitly named in an assignment request
- IANA may recover unused port numbers, via the new procedures of de-assignment, revocation, and transfer

Where possible, a given service is expected to demultiplex messages if necessary. For example, applications and protocols are expected to include in-band version information, so that future versions of the application or protocol can share the same assigned port. Applications and protocols are also expected to be able to efficiently use a single assigned port for multiple sessions, either by demultiplexing multiple streams within one port, or using the assigned port to coordinate using dynamic ports for subsequent exchanges (e.g., in the spirit of FTP [RFC0959].

These principles of port conservation are explained further in

[I-D.touch-tsvwg-port-use]. That document explains in further detail how ports are used in various ways, notably:

- as endpoint process identifiers
- as application protocol identifiers
- for firewall filtering purposes

Both the process identifier and the protocol identifier uses suggest that anything a single process can demultiplex, or that can be encoded into a single protocol, should be. The firewall filtering use suggests that some uses that could be multiplexed or encoded could instead be separated to allow for easier firewall management. Note that this latter use is much less sound, because port numbers have meaning only for the two endpoints involved in a connection, and drawing conclusions about the service that generated a given flow based on observed port numbers is not always reliable.

IANA will begin assigning port numbers for only those transport protocols explicitly included in an assignment request. This ends the long-standing practice of automatically assigning a port number to an application for both TCP and UDP, even if the request is for only one of these transport protocols. The new assignment procedure conserves resources by assigning a port number to an application for only those transport protocols (TCP, UDP, SCTP and/or DCCP) it actually uses. The port number will be marked as Reserved - instead of Assigned - in the port number registries of the other transport protocols. When applications start supporting the use of some of those additional transport protocols, the Assignee for the assignment MUST request IANA convert these reserved ports into assignments. An application MUST NOT assume that it can use a port number assigned to it for use with one transport protocol with another transport protocol without IANA converting the reservation into an assignment.

When the available pool of unassigned numbers has run out in a port range, it will be necessary for IANA to consider the Reserved ports for assignment. This is part of the motivation for not automatically assigning ports for transport protocols other than the requested one(s). This will allow more ports to be available for assignment at that point. To help conserve ports, application developers SHOULD request assignment of only those transport protocols that their application currently uses.

Conservation of port numbers is improved by procedures that allow previously allocated port numbers to become Unassigned, either through de-assignment or through revocation, and by a procedure that lets application designers transfer an assigned but unused port

number to a new application. Section 8 describes these procedures, which until now were undocumented. Port number conservation is also improved by recommending that applications that do not require an assigned port should register only a service name without an associated port number.

8. IANA Procedures for Managing the Service Name and Transport Protocol Port Number Registry

This section describes the process for handling requests associated with IANA's management of the Service Name and Transport Protocol Port Number Registry. Such requests include initial assignment, de-assignment, reuse, changes to the service name, and updates to the contact information or description associated with an assignment. Revocation is as additional process, initiated by IANA.

8.1. Service Name and Port Number Assignment

Assignment refers to the process of providing service names or port numbers to applicants. All such assignments are made from service names or port numbers that are Unassigned or Reserved at the time of the assignment.

- Unassigned names and numbers are allocated according to the rules described in Section 8.1.2 below.

- Reserved numbers and names are generally only assigned by a Standards Action or an IESG Approval, and MUST be accompanied by a statement explaining the reason a Reserved number or name is appropriate for this action. The only exception to this rule is that the current Assignee of a port number MAY request the assignment of the corresponding Reserved port number for other transport protocols when needed. IANA will initiate an "Expert Review" [RFC5226] for such requests.

When an assignment for one or more transport protocols is approved, the port number for any non-requested transport protocol(s) will be marked as Reserved. IANA SHOULD NOT assign that port number to any other application or service until no other port numbers remain Unassigned in the requested range.

8.1.1. General Assignment Procedure

A service name or port number assignment request contains the following information. The service name is the unique identifier of a given service:

```
Service Name (REQUIRED)
Transport Protocol(s) (REQUIRED)
Assignee (REQUIRED)
Contact (REQUIRED)
Description (REQUIRED)
Reference (REQUIRED)
Port Number (OPTIONAL)
Service Code (REQUIRED for DCCP only)
Known Unauthorized Uses (OPTIONAL)
Assignment Notes (OPTIONAL)
```

- Service Name: A desired unique service name for the service associated with the assignment request MUST be provided. This name may be used with various service selection and discovery mechanisms (including, but not limited to, DNS SRV records [RFC2782]). The name MUST be compliant with the syntax defined in Section 5.1. In order to be unique, they MUST NOT be identical to any currently assigned service name in the IANA registry [PORTREG]. Service names are case-insensitive; they may be provided and entered into the registry with mixed case for clarity, but for the comparison purposes the case is ignored.

- Transport Protocol(s): The transport protocol(s) for which an assignment is requested MUST be provided. This field is currently limited to one or more of TCP, UDP, SCTP, and DCCP. Requests without any port assignment and only a service name are still required to indicate which protocol the service uses.

- Assignee: Name and email address of the party to whom the assignment is made. This is REQUIRED. The Assignee is the organization, company or individual person responsible for the initial assignment. For assignments done through RFCs published via the "IETF Document Stream" [RFC4844], the Assignee will be the IESG <iesg@ietf.org>.

- Contact: Name and email address of the Contact person for the assignment. This is REQUIRED. The Contact person is the responsible person for the Internet community to send questions to. This person is also authorized to submit changes on behalf of the Assignee; in cases of conflict between the Assignee and the Contact, the Assignee decisions take precedence. Additional address information MAY be provided. For assignments done through RFCs published via the "IETF Document Stream" [RFC4844], the Contact will be the IETF Chair <chair@ietf.org>.

- Description: A short description of the service associated with the assignment request is REQUIRED. It should avoid all but the most well-known acronyms.

- Reference: A description of (or a reference to a document describing) the protocol or application using this port. The description must state whether the protocol uses IP-layer broadcast, multicast, or anycast communication.

  For assignments requesting only a Service Name, or a Service Name and User Port, a statement that the protocol is proprietary and not publicly documented is also acceptable, provided that the required information regarding the use of IP broadcast, multicast, or anycast is given.

  For any assignment request that includes a User Port, the assignment request MUST explain why a port number in the Dynamic Ports range is unsuitable for the given application.

  For any assignment request that includes a System Port, the assignment request MUST explain why a port number in the User Ports or Dynamic Ports ranges is unsuitable, and a reference to a stable protocol specification document MUST be provided.

  IANA MAY accept early assignment [RFC4020] requests (known as "early allocation" therein) from IETF Working Groups that reference a sufficiently stable Internet Draft instead of a published Standards-Track RFC.

- Port Number: If assignment of a port number is desired, either the port number the requester suggests for assignment or indication of port range (user or system) MUST be provided. If only a service name is to be assigned, this field is left empty. If a specific port number is requested, IANA is encouraged to assign the requested number. If a range is specified, IANA will choose a suitable number from the User or System Ports ranges. Note that the applicant MUST NOT use the requested port prior to the completion of the assignment.

- Service Code: If the assignment request includes DCCP as a transport protocol then the request MUST include a desired unique DCCP service code [RFC5595], and MUST NOT include a requested DCCP service code otherwise. Section 19.8 of the DCCP specification [RFC4340] defines requirements and rules for assignment, updated by this document. Note that, as per [RFC5595], some service codes are not assigned; zero (absence of a meaningful service code) or 4294967295 (invalid service code) are permanently reserved, and the Private service codes 1056964608-1073741823 (i.e., 32-bit

values with the high-order byte equal to a value of 63, corresponding to the ASCII character '?') are not centrally assigned.

- Known Unauthorized Uses: A list of uses by applications or organizations who are not the Assignee. This list may be augmented by IANA after assignment when unauthorized uses are reported.

- Assignment Notes: Indications of owner/name change, or any other assignment process issue. This list may be updated by IANA after assignment to help track changes to an assignment, e.g., de-assignment, owner/name changes, etc.

If the assignment request is for the addition of a new transport protocol to an already-assigned service name and the requester is not the Assignee or Contact for the already-assigned service name, IANA needs to confirm with the Assignee for the existing assignment whether this addition is appropriate.

If the assignment request is for a new service name sharing the same port as an already-assigned service name (see port number overloading in Section 5), IANA needs to confirm with the Assignee for the existing service name and other appropriate experts whether the overloading is appropriate.

When IANA receives an assignment request - containing the above information - that is requesting a port number, IANA SHALL initiate an "Expert Review" [RFC5226] in order to determine whether an assignment should be made. For requests that are not seeking a port number, IANA SHOULD assign the service name under a simple "First Come First Served" policy [RFC5226].

8.1.2. Variances for Specific Port Number Ranges

Section 6 describes the different port number ranges. It is important to note that IANA applies slightly different procedures when managing the different port ranges of the service name and port number registry:

- Ports in the Dynamic Ports range (49152-65535) have been specifically set aside for local and dynamic use and cannot be assigned through IANA. Application software may simply use any dynamic port that is available on the local host, without any sort of assignment. On the other hand, application software MUST NOT assume that a specific port number in the Dynamic Ports range will always be available for communication at all times, and a port number in that range hence MUST NOT be used as a service

identifier.

- Ports in the User Ports range (1024-49151) are available for assignment through IANA, and MAY be used as service identifiers upon successful assignment. Because assigning a port number for a specific application consumes a fraction of the shared resource that is the port number registry, IANA will require the requester to document the intended use of the port number. For most IETF protocols, ports in the User Ports range will be assigned under the "IETF Review" or "IESG Approval" procedures [RFC5226] and no further documentation is required. Where these procedures do not apply, then the requester must input the documentation to the "Expert Review" procedure [RFC5226], by which IANA will have a technical expert review the request to determine whether to grant the assignment. Regardless of the path ("IETF Review", "IESG Approval", or "Expert Review"), the submitted documentation is expected to be the same, as described in this section, and MUST explain why using a port number in the Dynamic Ports range is unsuitable for the given application. Further, IANA MAY utilize the Expert Review process informally to inform their position in participating in "IETF Review" and "IESG Review"

- Ports in the System Ports range (0-1023) are also available for assignment through IANA. Because the System Ports range is both the smallest and the most densely allocated, the requirements for new assignments are more strict than those for the User Ports range, and will only be granted under the "IETF Review" or "IESG Approval" procedures [RFC5226]. A request for a System Port number MUST document *both* why using a port number from the Dynamic Ports range is unsuitable *and* why using a port number from the User Ports range is unsuitable for that application.

8.2. Service Name and Port Number De-Assignment

The Assignee of a granted port number assignment can return the port number to IANA at any time if they no longer have a need for it. The port number will be de-assigned and will be marked as Reserved. IANA should not re-assign port numbers that have been de-assigned until all unassigned port numbers in the specific range have been assigned.

Before proceeding with a port number de-assignment, IANA needs to reasonably establish that the value is actually no longer in use.

Because there is much less danger of exhausting the service name space compared to the port number space, it is RECOMMENDED that a given service name remain assigned even after all associated port number assignments have become de-assigned. Under this policy, it will appear in the registry as if it had been created through a

service name assignment request that did not include any port numbers.

On rare occasions, it may still be useful to de-assign a service name. In such cases, IANA will mark the service name as Reserved. IANA will involve their IESG-appointed expert in such cases.

IANA will include a comment in the registry when de-assignment happens to indicate its historic usage.

8.3. Service Name and Port Number Reuse

If the Assignee of a granted port number assignment no longer has a need for the assigned number, but would like to reuse it for a different application, they can submit a request to IANA to do so.

Logically, port number reuse is to be thought of as a de-assignment (Section 8.2) followed by an immediate (re-)assignment (Section 8.1) of the same port number for a new application. Consequently, the information that needs to be provided about the proposed new use of the port number is identical to what would need to be provided for a new port number assignment for the specific ports range.

Because there is much less danger of exhausting the service name space compared to the port number space, it is RECOMMENDED that the original service name associated with the prior use of the port number remains assigned, and a new service name be created and associated with the port number. This is again consistent with viewing a reuse request as a de-assignment followed by an immediate (re-)assignment. Re-using an assigned service name for a different application is NOT RECOMMENDED.

IANA needs to carefully review such requests before approving them. In some instances, the Expert Reviewer will determine that the application the port number was assigned to has found usage beyond the original Assignee, or that there is a concern that it may have such users. This determination MUST be made quickly. A community call concerning revocation of a port number (see below) MAY be considered, if a broader use of the port number is suspected.

8.4. Service Name and Port Number Revocation

A port number revocation can be thought of as an IANA-initiated de-assignment (Section 8.2), and has exactly the same effect on the registry.

Sometimes, it will be clear that a specific port number is no longer in use and that IANA can revoke it and mark it as Reserved. At other

times, it may be unclear whether a given assigned port number is still in use somewhere in the Internet. In those cases, IANA must carefully consider the consequences of revoking the port number, and SHOULD only do so if there is an overwhelming need.

With the help of their IESG-appointed Expert Reviewer, IANA SHALL formulate a request to the IESG to issue a four-week community call concerning the pending port number revocation. The IESG and IANA, with the Expert Reviewer's support, SHALL determine promptly after the end of the community call whether revocation should proceed and then communicate their decision to the community. This procedure typically involves similar steps to de-assignment except that it is initiated by IANA.

Because there is much less danger of exhausting the service name space compared to the port number space, revoking service names is NOT RECOMMENDED.

8.5. Service Name and Port Number Transfers

The value of service names and port numbers is defined by their careful management as a shared Internet resource, whereas enabling transfer allows the potential for associated monetary exchanges. As a result, the IETF does not permit service name or port number assignments to be transferred between parties, even when they are mutually consenting.

The appropriate alternate procedure is a coordinated de-assignment and assignment: The new party requests the service name or port number via an assignment and the previous party releases its assignment via the de-assignment procedure outlined above.

With the help of their IESG-appointed Expert Reviewer, IANA SHALL carefully determine if there is a valid technical, operational or managerial reason to grant the requested new assignment.

8.6. Maintenance Issues

In addition to the formal procedures described above, updates to the Description and Contact information are coordinated by IANA in an informal manner, and may be initiated by either the Assignee or by IANA, e.g., by the latter requesting an update to current Contact information. (Note that the Assignee cannot be changed as a separate procedure; see instead Section 8.5 above.)

8.7. Disagreements

In the case of disagreements around any request there is the possibility of appeal following the normal appeals process for IANA assignments as defined by Section 7 of "Guidelines for Writing an IANA Considerations Section in RFCs" [RFC5226].

9. Security Considerations

The IANA guidelines described in this document do not change the security properties of UDP, TCP, SCTP, or DCCP.

Assignment of a service name or port number does not in any way imply an endorsement of an application or product, and the fact that network traffic is flowing to or from an assigned port number does not mean that it is "good" traffic, or even that it is used by the assigned service. Firewall and system administrators should choose how to configure their systems based on their knowledge of the traffic in question, not based on whether or not there is an assigned service name or port number.

Services are expected to include support for security, either as default or dynamically negotiated in-band. The use of separate service name or port number assignments for secure and insecure variants of the same service is to be avoided in order to discourage the deployment of insecure services.

10. IANA Considerations

This document obsoletes Sections 8 and 9.1 of the March 2000 IANA Allocation Guidelines [RFC2780].

Upon approval of this document, IANA is requested to contact Stuart Cheshire, maintainer of the independent service name registry [SRVREG], in order to merge the contents of that private registry into the official IANA registry. It is expected that the independent registry web page will be updated with pointers to the IANA registry and to this RFC.

IANA is instructed to create a new service name entry in the service name and port number registry [PORTREG] for any entry in the "Protocol and Service Names" registry [PROTSERVREG] that does not already have one assigned.

IANA is also instructed to indicate in the Assignment Notes for "www" and "www-http" that they are duplicate terms that refer to the "http"

service, and should not be used for discovery purposes. For this conceptual service (human-readable web pages served over HTTP) the correct service name to use for service discovery purposes is "http" (see Section 5).

10.1. Service Name Consistency

Section 8.1 defines which character strings are well-formed service names, which until now had not been clearly defined. The definition in Section 8.1 was chosen to allow maximum compatibility of service names with current and future service discovery mechanisms.

As of August 5, 2009 approximately 98% of the so-called "Short Names" from existing port number assignments [PORTREG] meet the rules for legal service names stated in Section 8.1, and hence for these services their service name will be exactly the same as their "Short Name".

The remaining approximately 2% of the exiting "Short Names" are not suitable to be used directly as well-formed service names because they contain illegal characters such as asterisks, dots, pluses, slashes, or underscores. All existing "Short Names" conform to the length requirement of 15 characters or fewer. For these unsuitable "Short Names", listed in the table below, the service name will be the Short Name with any illegal characters replaced by hyphens. IANA SHALL add an entry to the registry giving the new well-formed primary service name for the existing service, that otherwise duplicates the original assignment information. In the description field of this new entry giving the primary service name, IANA SHALL record that it assigns a well-formed service name for the previous service and reference the original assignment. In the Assignment Notes field of the original assignment, IANA SHALL add a note that this entry is an alias to the new well-formed service name, and that the old service name is historic, not usable for use with many common service discovery mechanisms.

Names containing illegal characters to be replaced by hyphens:

914c/g	acmaint_dbd	acmaint_transd
atex_elmd	avanti_cdp	badm_priv
badm_pub	bdir_priv	bdir_pub
bmc_ctd_ldap	bmc_patroldb	boks_clntd
boks_servc	boks_servm	broker_service
bues_service	canit_store	cedros_fds
cl/1	contamac_icm	corel_vncadmin
csc_proxy	cvc_hostd	dbcontrol_agent
dec_dlm	dl_agent	documentum_s
dsmeter_iatc	dsx_monitor	elpro_tunnel
elvin_client	elvin_server	encrypted_admin
erunbook_agent	erunbook_server	esri_sde
EtherNet/IP-1	EtherNet/IP-2	event_listener
flr_agent	gds_db	ibm_wrless_lan
iceedcp_rx	iceedcp_tx	iclcnet_svinfo
idig_mux	ife_icorp	instl_bootc
instl_boots	intel_rci	interhdl_elmd
lan900_remote	LiebDevMgmt_A	LiebDevMgmt_C
LiebDevMgmt_DM	mapper-ws_ethd	matrix_vnet
mdbs_daemon	menandmice_noh	msl_lmd
nburn_id	ncr_ccl	nds_sso
netmap_lm	nms_topo_serv	notify_srvr
novell-lu6.2	nuts_bootp	nuts_dem
ocs_amu	ocs_cmu	pipe_server
pra_elmd	printer_agent	redstorm_diag
redstorm_find	redstorm_info	redstorm_join
resource_mgr	rmonitor_secure	rsvp_tunnel
sai_sentlm	sge_execd	sge_qmaster
shiva_confsrvr	sql*net	srvc_registry
stm_pproc	subntbcst_tftp	udt_os
universe_suite	veritas_pbx	vision_elmd
vision_server	wrs_registry	z39.50

Following the example set by the "application/whoispp-query" MIME Content-Type [RFC2957], the service name for "whois++" will be "whoispp".

10.2. Port Numbers for SCTP and DCCP Experimentation

Two System UDP and TCP ports, 1021 and 1022, have been reserved for experimental use [RFC4727]. This document assigns the same port numbers for SCTP and DCCP, updates the TCP and UDP assignments, and also instructs IANA to automatically assign these two port numbers for any future transport protocol with a similar 16-bit port number

namespace.

Note that these port numbers are meant for temporary experimentation and development in controlled environments. Before using these port numbers, carefully consider the advice in Section 6.1 in this document, as well as in Sections 1 and 1.1 of "Assigning Experimental and Testing Numbers Considered Useful" [RFC3692]. Most importantly, application developers must request a permanent port number assignment from IANA as described in Section 8.1 before any kind of non-experimental deployment.

Service Name	exp1
Transport Protocol	DCCP, SCTP, TCP, UDP
Assignee	IESG <iesg@ietf.org>
Contact	IETF Chair <chair@ietf.org>
Description	RFC3692-style Experiment 1
Reference	[RFC4727] [RFCyyyy]
Port Number	1021

Service Name	exp2
Transport Protocol	DCCP, SCTP, TCP, UDP
Assignee	IESG <iesg@ietf.org>
Contact	IETF Chair <chair@ietf.org>
Description	RFC3692-style Experiment 2
Reference	[RFC4727] [RFCyyyy]
Port Number	1022

[RFC Editor Note: Please change "yyyy" to the RFC number allocated to this document before publication.]

10.3. Updates to DCCP Registries

This document updates the IANA assignment procedures for the DCCP Port Number and DCCP Service Codes Registries [RFC4340].

10.3.1. DCCP Service Code Registry

Service Codes are assigned first-come-first-served according to Section 19.8 of the DCCP specification [RFC4340]. This document updates that section by extending the guidelines given there in the following ways:

- IANA MAY assign new Service Codes without seeking Expert Review using their discretion, but SHOULD seek expert review if a request asks for more than five Service Codes.

- IANA should feel free to contact the DCCP Expert Reviewer with any questions related to requests for DCCP-related codepoints.

10.3.2. DCCP Port Numbers Registry

The DCCP ports registry is defined by Section 19.9 of the DCCP specification [RFC4340]. Assignments in this registry require prior assignment of a Service Code. Not all Service Codes require IANA-assigned ports. This document updates that section by extending the guidelines given there in the following way:

- IANA should normally assign a value in the range 1024-49151 to a DCCP server port. IANA requests to assign port numbers in the System Ports range (0 through 1023), require an "IETF Review" [RFC5226] prior to assignment by IANA [RFC4340].

- IANA MUST NOT assign more than one DCCP server port to a single service code value.

- The assignment of multiple service codes to the same DCCP port is allowed, but subject to expert review.

- The set of Service Code values associated with a DCCP server port should be recorded in the service name and port number registry.

- A request for additional Service Codes to be associated with an already-allocated Port Number requires Expert Review. These requests will normally be accepted when they originate from the contact associated with the port assignment. In other cases, these applications will be expected to use an unallocated port, when this is available.

The DCCP specification [RFC4340] notes that a short port name MUST be associated with each DCCP server port that has been assigned. This document clarifies that this short port name is the Service Name as defined here, and this name MUST be unique.

11. Contributors

Alfred Hoenes (ah@tr-sys.de) and Allison Mankin (mankin@psg.com) have contributed text and ideas to this document.

12. Acknowledgments

The text in Section 10.3 is based on a suggestion originally proposed as a part of the DCCP Service Codes document [RFC5595] by Gorry Fairhurst.

Lars Eggert is partly funded by the Trilogy Project [TRILOGY], a research project supported by the European Commission under its Seventh Framework Program.

13. References

13.1. Normative References

[ANSI.X3-4.1986]
American National Standards Institute, "Coded Character Set - 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.

[RFC4020] Kompella, K. and A. Zinin, "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 4020, February 2005.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, November 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5595] Fairhurst, G., "The Datagram Congestion Control Protocol (DCCP) Service Codes", RFC 5595, September 2009.

13.2. Informative References

[I-D.cheshire-dnsext-dns-sd]
Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", draft-cheshire-dnsext-dns-sd-08 (work in progress), January 2011.

[I-D.cheshire-nat-pmp]
Cheshire, S., "NAT Port Mapping Protocol (NAT-PMP)", draft-cheshire-nat-pmp-03 (work in progress), April 2008.

[I-D.touch-tsvwg-port-use]
Touch, J., "Recommendations for Transport Port Uses", draft-touch-tsvwg-port-use-00 (work in progress), December 2010.

[IGD] UPnP Forum, "Internet Gateway Device (IGD) V 1.0", November 2001.

[PORTREG] Internet Assigned Numbers Authority (IANA), "Service Name and Transport Protocol Port Number Registry", http://www.iana.org/assignments/port-numbers.

[PROTSERVREG]
Internet Assigned Numbers Authority (IANA), "Protocol and Service Names Registry", http://www.iana.org/assignments/service-names.

[RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

[RFC1078] Lottor, M., "TCP port service Multiplexer (TCPMUX)", RFC 1078, November 1988.

[RFC1340] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1340,

July 1992.

[RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1700, October 1994.

[RFC2957] Daigle, L. and P. Faltstrom, "The application/whoispp-query Content-Type", RFC 2957, October 2000.

[RFC3232] Reynolds, J., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, January 2002.

[RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, January 2004.

[RFC4342] Floyd, S., Kohler, E., and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)", RFC 4342, March 2006.

[RFC4844] Daigle, L. and Internet Architecture Board, "The RFC Series and RFC Editor", RFC 4844, July 2007.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC5237] Arkko, J. and S. Bradner, "IANA Allocation Guidelines for the Protocol Field", BCP 37, RFC 5237, February 2008.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

[SRVREG] "DNS SRV Service Types Registry", http://www.dns-sd.org/ServiceTypes.html.

[SYSFORM] Internet Assigned Numbers Authority (IANA), "Application for System (Well Known) Port Number", http://www.iana.org/.

[TRILOGY] "Trilogy Project", http://www.trilogy-project.org/.

[USRFORM] Internet Assigned Numbers Authority (IANA), "Application for User (Registered) Port Number", http://www.iana.org/.

Authors' Addresses

Michelle Cotton
Internet Corporation for Assigned Names and Numbers
4676 Admiralty Way, Suite 330
Marina del Rey, CA 90292
USA

Phone: +1 310 823 9358
Email: michelle.cotton@icann.org
URI: http://www.iana.org/

Lars Eggert
Nokia Research Center
P.O. Box 407
Nokia Group 00045
Finland

Phone: +358 50 48 24461
Email: lars.eggert@nokia.com
URI: http://research.nokia.com/people/lars_eggert/

Joe Touch
USC/ISI
4676 Admiralty Way
Marina del Rey, CA 90292
USA

Phone: +1 310 448 9151
Email: touch@isi.edu
URI: http://www.isi.edu/touch

Magnus Westerlund
Ericsson
Farogatan 6
Stockholm 164 80
Sweden

Phone: +46 8 719 0000
Email: magnus.westerlund@ericsson.com

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
USA

Phone: +1 408 974 3207
Email: cheshire@apple.com

TSVWG WG
Internet-Draft
Expires: September 14, 2011
Intended Status: Standards Track (PS)
Updates: RFC 2205, 2210, & 4495 (if published as an RFC)

James Polk
Subha Dhesikan
Cisco Systems

March 14, 2011

Integrated Services (IntServ) Extension to Allow Signaling of Multiple Traffic Specifications and Multiple Flow Specifications in RSVPv1
draft-polk-tsvwg-intserv-multiple-tspec-06

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 to IETF in full conformance with the provisions of BCP 78 and BCP 79.

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

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/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 September 14, 2011.

Table of Contents

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.

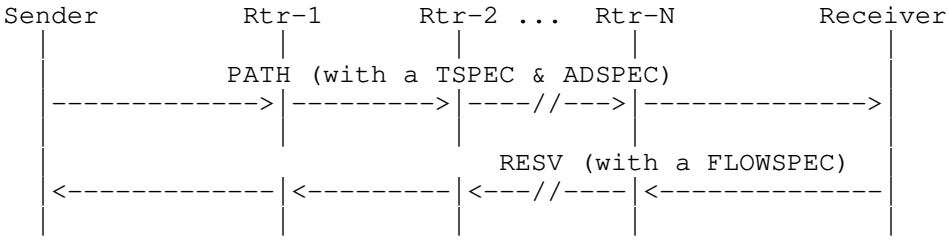


Figure 1. Concept of RSVP in a Single Direction

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.

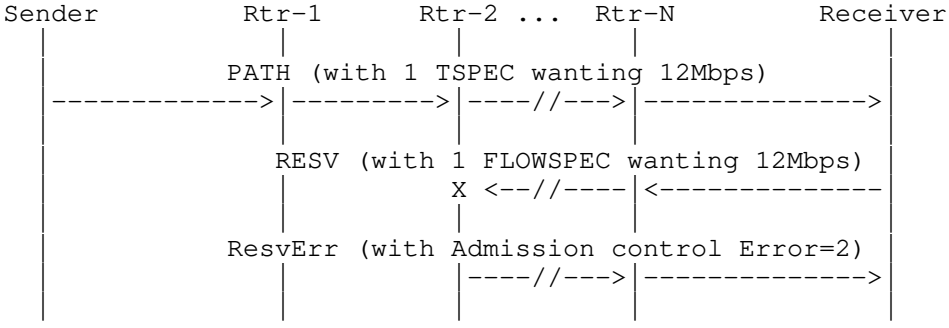


Figure 2. Concept of RSVP Rejection due to Limited Bandwidth

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]:

```
        <PATH Message> ::= <Common Header> [ <INTEGRITY> ]

                                   <SESSION> <RSVP_HOP>

                                   <TIME_VALUES>

                                  [ <POLICY_DATA> ... ]

                                  [ <sender descriptor> ]

        <sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
                                                   ^^^^^^^^^^^^

                                  [ <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:

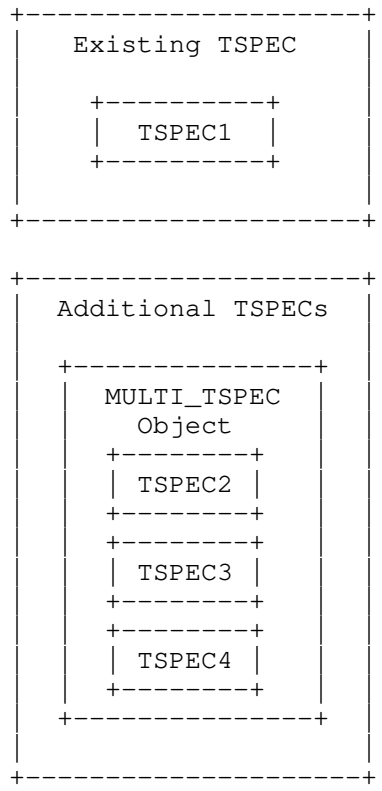


Figure 3. Encoding of Multiple Traffic Specifications in the TSPEC and MULTI_TSPEC objects

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 the RESV message.

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:

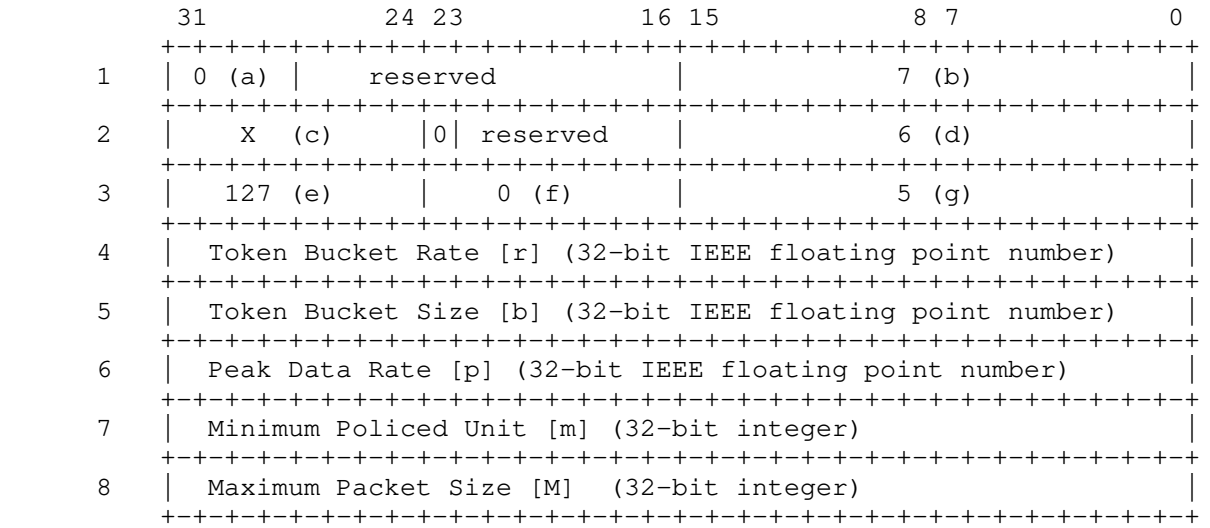
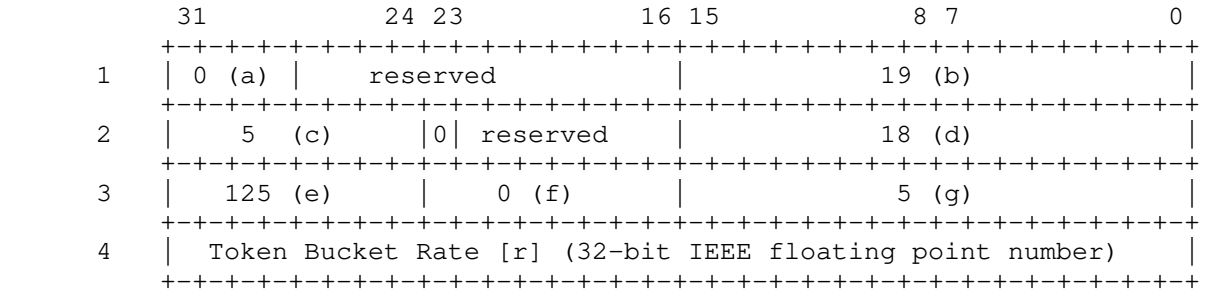


Figure 4. SENDER_TSPEC in PATH

(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.



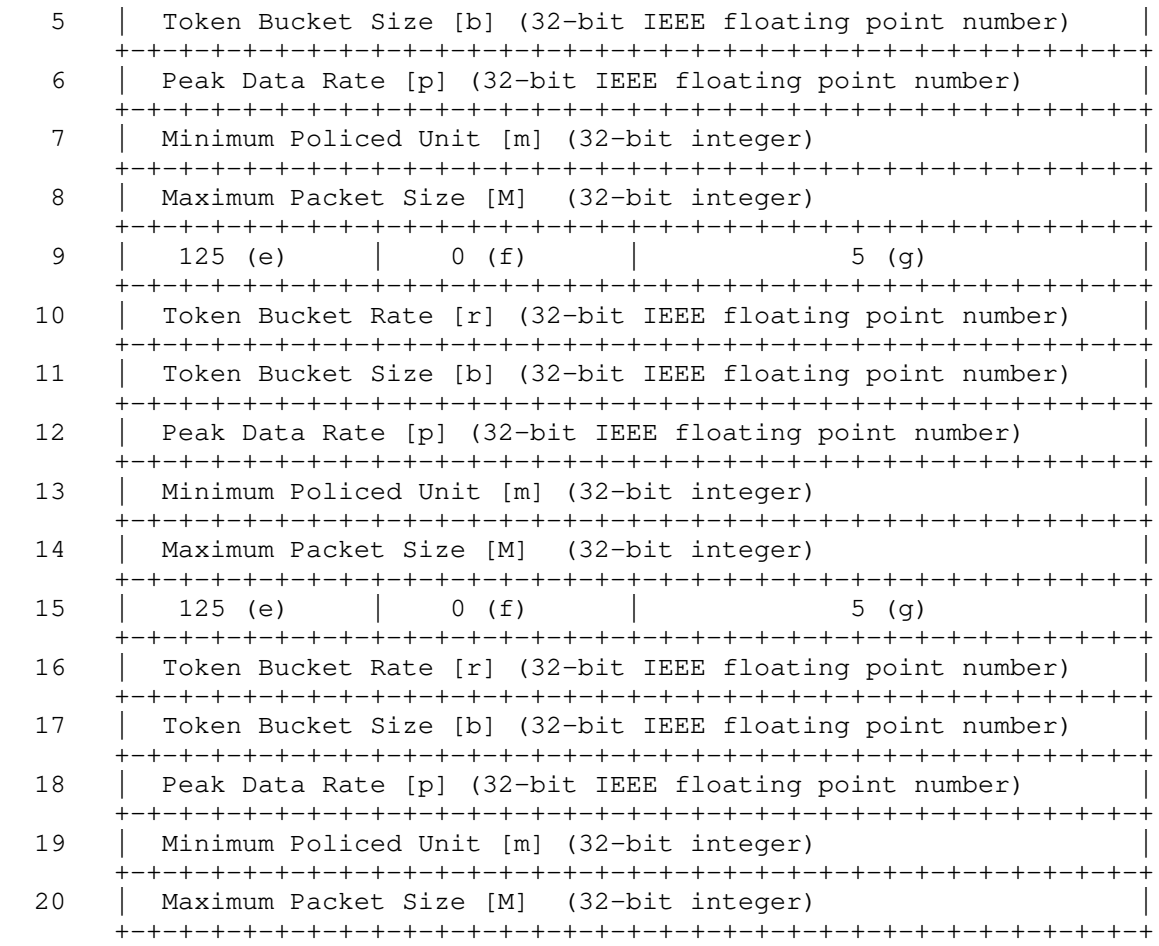


Figure 5. MULTI_TSPEC Object

(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.

The Service header, here service number 5 (Controlled-Load) MUST remain the same.

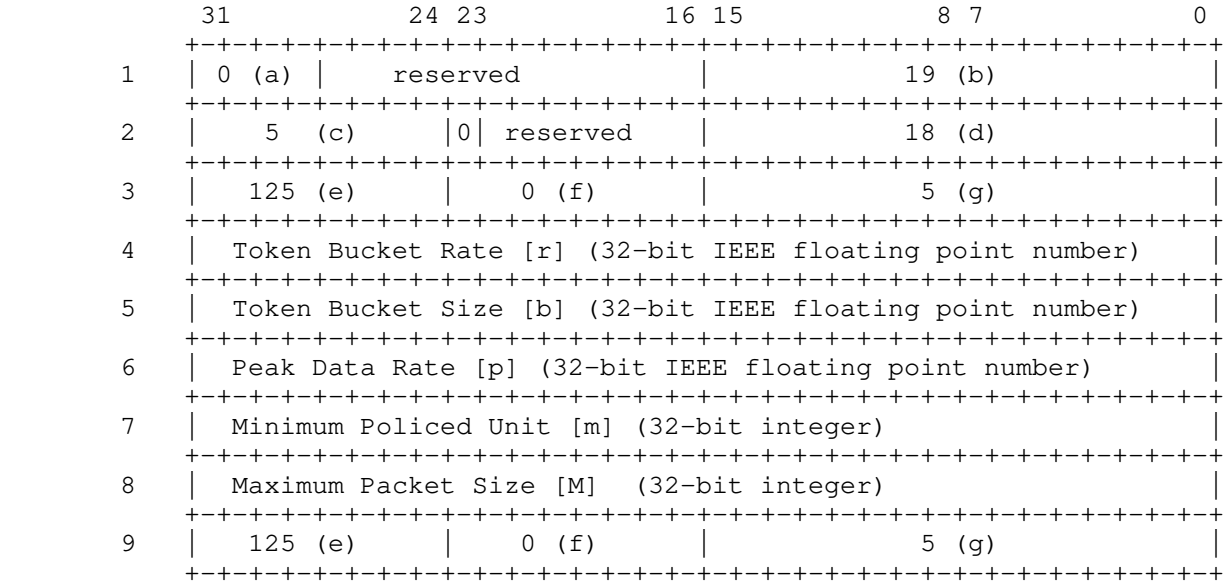
Each TSPEC is six 32-bit Words long (the per-service header plus the 5 values that are 1 Word each in length), therefore the length is in 6 Word increments for each additional TSPEC. Case in point, from the above Figure 5, Words 3-8 are the first TSPEC (2nd preferred), Words 9-14 are the next TSPEC (3rd 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:



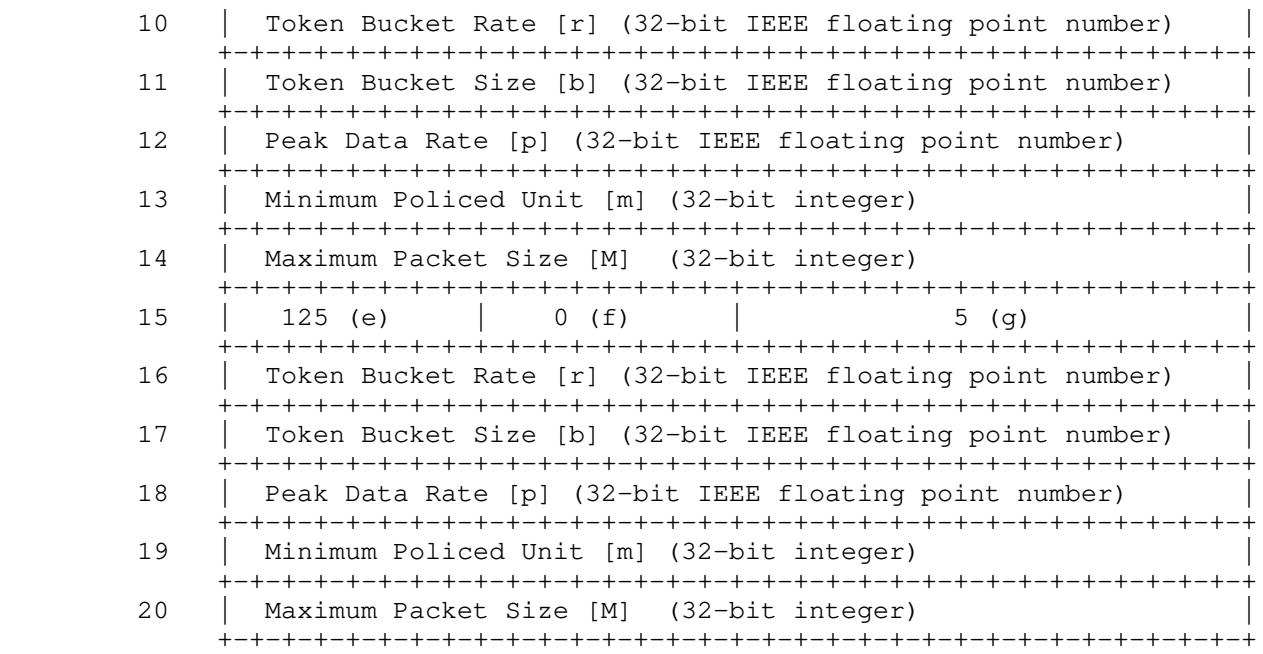


Figure 5. Multiple FLOWSPEC for Controlled-Load service

(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:

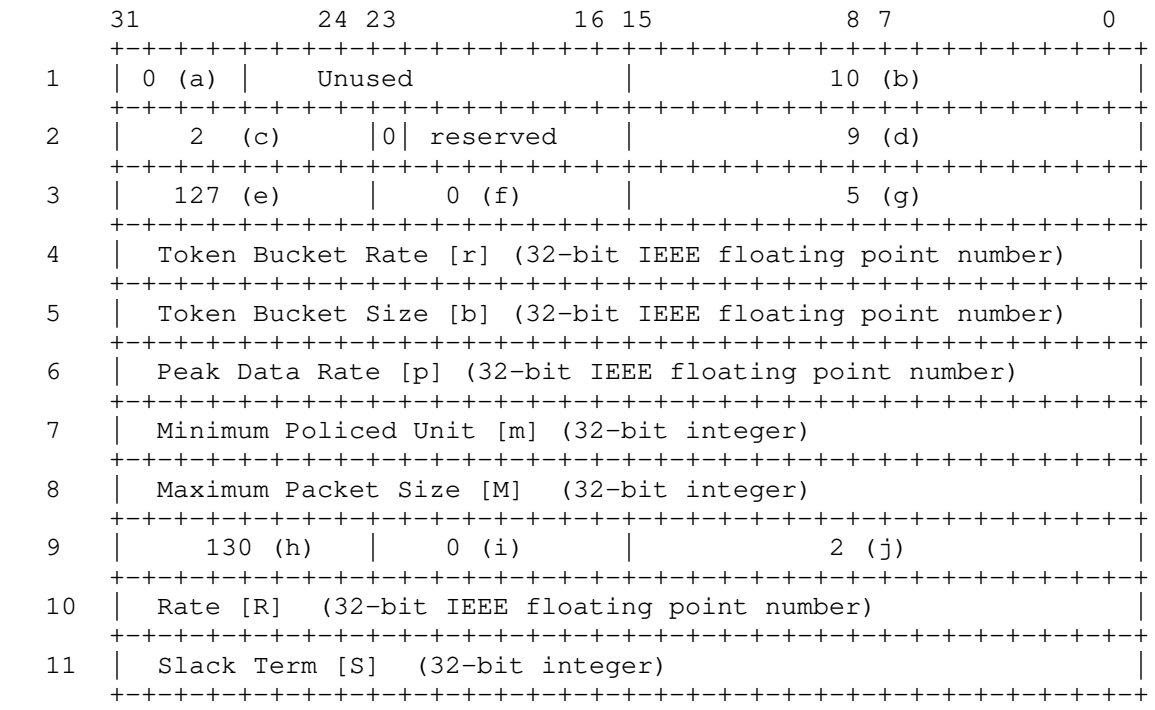
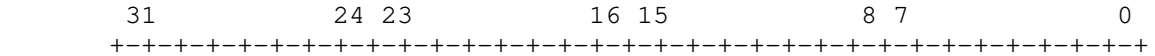


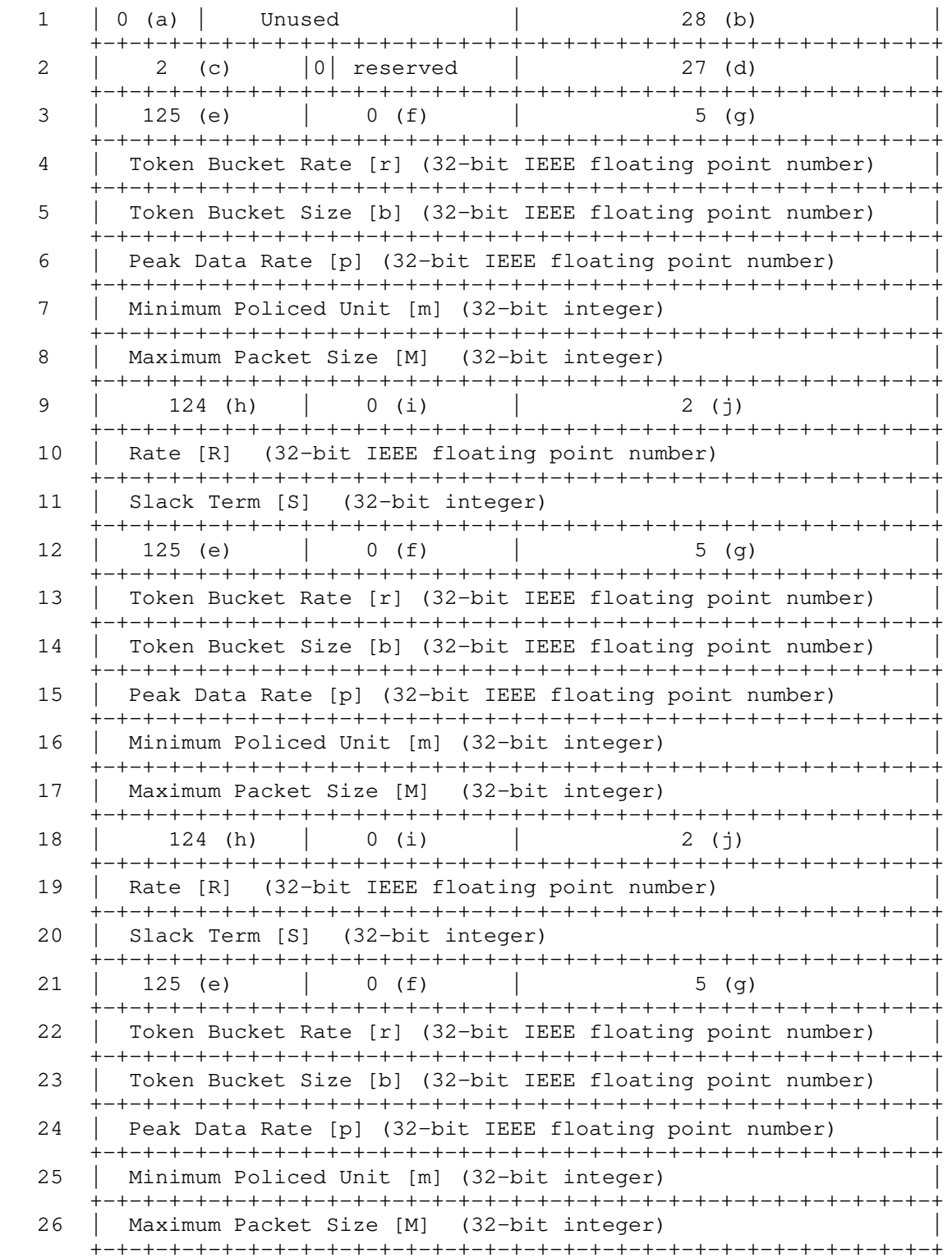
Figure 6. FLOWSPEC for Guaranteed service

(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.




0 (a)
Unused
28 (b)
2 (c)
0
reserved
27 (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)
124 (h)
0 (i)
2 (j)
Rate [R] (32-bit IEEE floating point number)
Slack Term [S] (32-bit integer)
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)
124 (h)
0 (i)
2 (j)
Rate [R] (32-bit IEEE floating point number)
Slack Term [S] (32-bit integer)
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)

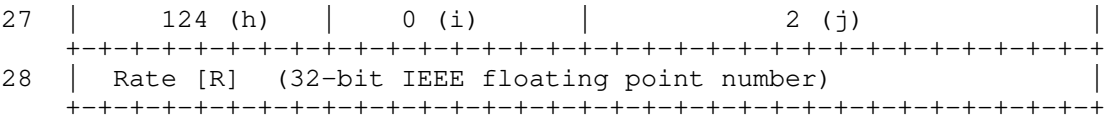


Figure 7. Multiple FLOWSPECs for Guaranteed service

(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 session.

The bandwidth reduction RFC [RFC4495] has the ability to partially

preempt existing reservations. However, it does not address the need that this draft 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 with 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 will extend the RFC 4495 defined error to work for previous hops while a reservation is being established.

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:

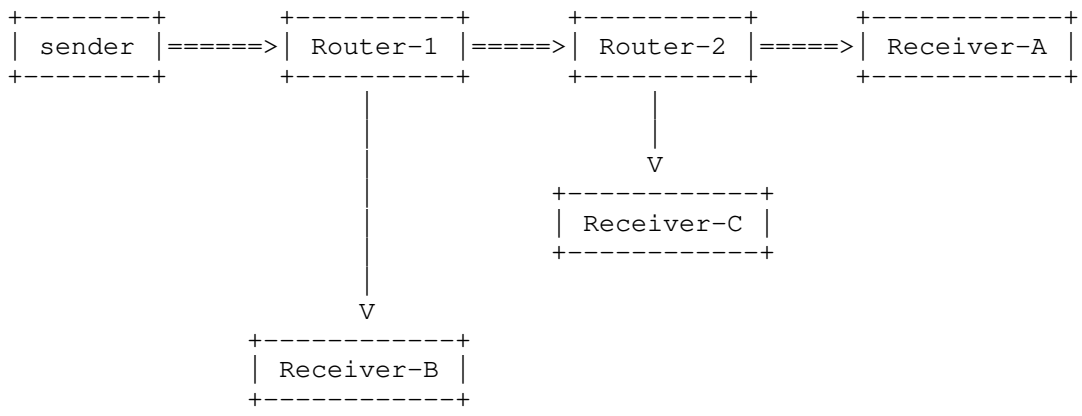


Figure 8. MULTI_TPSEC and Multicast

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:

o Both the idea of MULTI_RSPEC and MULTI_FLOWSPEC need to be fleshed out, and IANA registered.

o 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.

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 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

[RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997

[RFC2205] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997

[RFC2210] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997

[RFC2212] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997

[RFC2215] S. Shenker, J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", RFC 2212, September 1997

[RFC4495] J. Polk, S. Dhesikan, "A Resource Reservation Protocol (RSVP) Extension for the Reduction of Bandwidth of a Reservation Flow", RFC 4495, May 2006

8.2. Informative References

[IANA] http://www.iana.org/assignments/integ-serv

Authors' Addresses

James Polk
3913 Treemont Circle
Colleyville, Texas, USA
+1.817.271.3552

mailto: jmpolk@cisco.com

Subha Dhesikan
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134 USA

mailto: sdhesika@cisco.com

Appendix A: 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:

```
        <PATH Message> ::= <Common Header> [ <INTEGRITY> ]

                                    <SESSION> <RSVP_HOP>

                                    <TIME_VALUES>

                                   [ <POLICY_DATA> ... ]

                                   [ <sender descriptor> ]

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

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 may be added:

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:

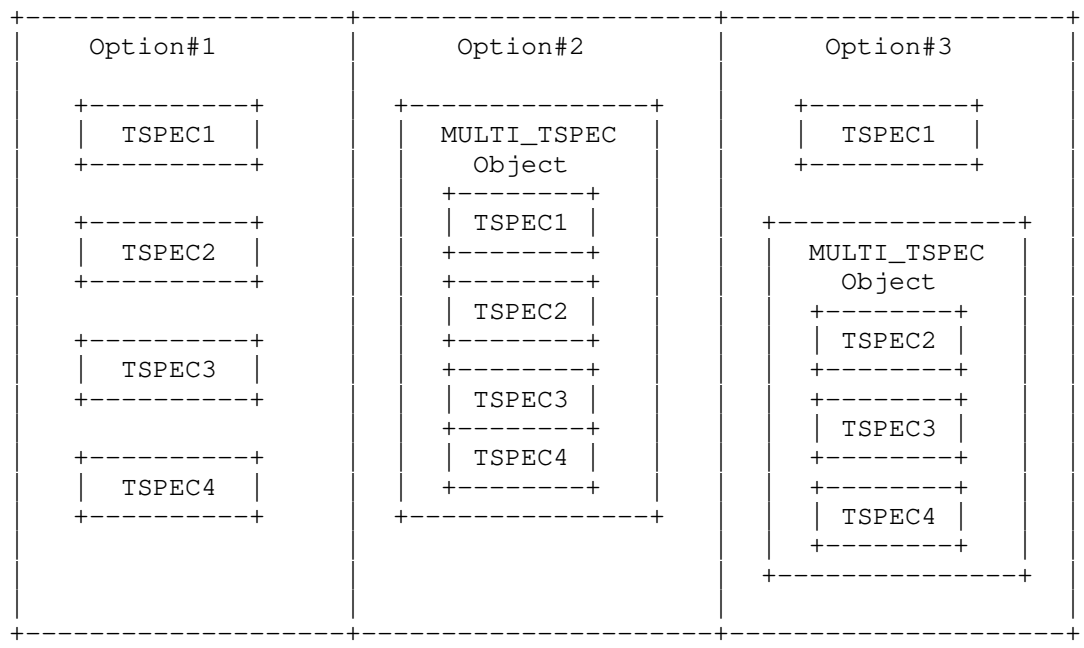


Figure 3. Concept of Option Choice

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.