Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015
D. York
Internet Society
D. Black, Ed.
EMC
C. Jennings
P. Jones
Cisco
July 4, 2014

Differentiated Services (DiffServ) and Real-time Communication
draft-york-dart-dscp-rtp-02

Abstract

This document describes the interaction between Differentiated Services (DiffServ) network quality of service (QoS) functionality and real-time network communication, including communication based on the Real-time Transport Protocol (RTP). DiffServ is based on network nodes applying different forwarding treatments to packets whose IP headers are marked with different DiffServ Code Points (DSCPs). As a result, use of different DSCPs within a single traffic stream may cause transport protocol interactions (e.g., reordering). In addition, DSCP markings may be changed or removed between the traffic source and destination. This document covers the implications of these DiffServ aspects for real-time network communication, including RTCWEB.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Requirements Language 3
2. Background 3
2.1. RTP Background 3
2.2. Differentiated Services (DiffServ) Background 5
2.3. Diffserv PHBs (Per-Hop Behaviors) 7
2.4. DiffServ, Reordering and Transport Protocols 8
2.5. DiffServ, Reordering and Real-Time Communication 9
2.6. Traffic Classifiers and DSCP Remarking 10
3. RTP Multiplexing Background 11
4. Recommendations 12
5. Examples 13
6. IANA Considerations 14
7. Security Considerations 14
8. Acknowledgements 15
9. References 15
9.1. Normative References 15
9.2. Informative References 16
Authors' Addresses 19

1. Introduction

This document describes the interactions between Differentiated Services (DiffServ) network quality of service (QoS) functionality [RFC2475] and real-time network communication, including communication based on the Real-time Transport Protocol (RTP) [RFC3550]. DiffServ is based on network nodes applying different forwarding treatments to packets whose IP headers are marked with different DiffServ Code Points (DSCPs)[RFC2474]. As a result use of different DSCPs within a single traffic stream may cause transport protocol interactions (e.g., reordering). In addition, DSCP markings may be changed or removed between the traffic's source and destination. This document covers the implications of these DiffServ aspects for real-time network communication, including RTCWEB traffic [I-D.ietf-rtcweb-overview].

1.1. Requirements Language

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

2. Background

Real-time communications enables communication in real-time over an IP network using voice, video, text, content sharing, etc. It is possible to use one or more of these modalities in parallel in order to provide a richer communication experience.

A simple example of real-time communications is a voice call placed over the Internet wherein an audio stream is transmitted in each direction between two users. A more complex example is an immersive videoconferencing system that has multiple video screens, multiple cameras, multiple microphones, and some means of sharing content. For such complex systems, there may be multiple media streams that may be transmitted via a single IP address and port or via multiple IP addresses and ports.

2.1. RTP Background

The most common protocol used for real time media is the Real-Time Transport Protocol (RTP) [RFC3550]. RTP defines a common encapsulation format and handling rules for real-time data transmitted over the Internet. Unfortunately, RTP terminology usage has been inconsistent. For example, this document on RTP grouping terminology [I-D.ietf-avtext-rtp-grouping-taxonomy] observes that:

> RFC 3550 [RFC3550] uses the terms media stream, audio stream, video stream and streams of (RTP) packets interchangeably.

Terminology in this document is based on that RTP grouping terminology document with the following terms being of particular importance (see that terminology document for full definitions):

Source Stream: A reference clock synchronized, time progressing, digital media stream.

RTP Stream: A stream of RTP packets containing media data, which may be source data or redundant data. The RTP Packet Stream is identified by an RTP synchronization source (SSRC) belonging to a particular RTP session.

Media encoding and packetization of a source stream results in a source RTP stream plus zero or more redundancy RTP streams that provide resilience against loss of packets from the source RTP stream [I-D.ietf-avtext-rtp-grouping-taxonomy]. Redundancy information may also be carried in the same RTP stream as the encoded source stream, e.g., see Section 7.2 of [RFC5109]. With most applications, a single media type (e.g., audio) is transmitted within a single RTP session. However, it is possible to transmit multiple, distinct source streams over the same RTP session as one or more individual RTP streams. This is referred to as RTP multiplexing.

The number of source streams and RTP streams in an overall real-time interaction can be surprisingly large. In addition to a voice source stream and a video source stream, there could be separate source streams for each of the cameras or microphones on a videoconferencing system. As noted above, there might also be separate redundancy RTP streams that provide protection to a source RTP stream, using techniques such as Forward Error Correction. Another example is simulcast transmission, where a video source stream can be transmitted at high resolution and low resolution RTP streams at the same time. In this case, a media processing function might choose to send one or both RTP streams onward to a receiver based on bandwidth availability or who the active speaker is in a multipoint conference. Lastly, a transmitter might send a the same media content concurrently as two RTP streams using different encodings (e.g., VP8 in parallel with H.264) to allow a media processing function to select a media encoding that best matches the capabilities of the receiver.

Other transport protocols may also be used to transmit real-time data or near-real-time data. For example, SCTP can be utilized to carry application sharing or whiteboarding information as part of an overall interaction that includes real time media. These additional transport protocols can be multiplexed with an RTP session via UDP encapsulation, thereby using a single pair of UDP ports.

The RTCWEB protocol suite [I-D.ietf-rtcweb-transports] employs two layers of multiplexing:

1. Individual source streams are carried in one or more individual RTP streams that can be multiplexed into a single RTP session as described in [RFC3550]; and

2. An RTP session could be multiplexed with other protocols via UDP encapsulation over a common pair of UDP ports as described in

[RFC5764] and [I-D.petithuguenin-avtcore-rfc5764-mux-fixes]. The resulting unidirectional UDP packet flow is identified by a 5-tuple, i.e., a combination of two IP addresses (source and destination), two UDP ports (source and destination), and the use of the UDP protocol.

For RTCWEB,an individual source stream is a MediaStreamTrack, and a MediaStream contains one or more MediaStreamTracks [W3C.WD-mediacapture-streams-20130903]. A MediaStreamTrack is transmitted as a source RTP stream plus zero or more redundancy RTP streams, so a MediaStream that consists of one MediaStreamTrack is transmitted as a single source RTP stream plus zero or more redundancy RTP streams.

For more information on use of RTP in RTCWEB, see [I-D.ietf-rtcweb-rtp-usage].

[I-D.westerlund-avtcore-transport-multiplexing] proposes to allow multiple RTP sessions to be multiplexed over a single UDP 5-tuple; the future of that expired proposal is uncertain.

For IPv6, addition of the flow label [RFC6437] to 5-tuples results in 6-tuples, but in practice, use of a flow label is unlikely to result in a finer-grain traffic subset than the corresponding 5-tuple (e.g., the flow label is likely to represent the combination of two ports with use of the UDP protocol). For that reason, discussion in this draft focuses on UDP 5-tuples.

[Editor’s Note: Multiple RTP sessions cannot be multiplexed on the same UDP 5-tuple, but what about multiple DTLS sessions for RTP? RFC 5764 appears to allow multiple DTLS sessions.]

[Editor’s Note: Should RTCP multiplexing w/RTP be mentioned here, as described in RFC 5761?]

2.2. Differentiated Services (DiffServ) Background

The DiffServ architecture is intended to enable scalable service discrimination in the Internet without requiring each network node to store per-flow state and participate in per-flow signaling. The services may be end-to-end or within a network; they include both those that can satisfy quantitative performance requirements (e.g., peak bandwidth) and those based on relative performance (e.g., "class" differentiation). Services can be constructed by a combination of well-defined building blocks deployed in network nodes that:

o classify traffic and set bits in an IP header field at network boundaries or hosts,

o use those bits to determine how packets are forwarded by the nodes inside the network, and

o condition the marked packets (e.g., meter, mark, shape, police) at network boundaries in accordance with the requirements or rules of each service.

A network node that supports DiffServ includes a classifier that selects packets based on the value of the DS field in IP headers, along with buffer management and packet scheduling mechanisms capable of delivering the specific packet forwarding treatment indicated by the DS field value. Setting of the DS field and fine-grain conditioning of marked packets need only be performed at network boundaries; internal network nodes operate on traffic aggregates that share a DS field value, or in some cases, a small set of related values.

The DiffServ architecture[RFC2475] maintains distinctions among:

o the QoS service provided to a traffic aggregate,

o the conditioning functions and per-hop behaviors (PHBs) used to realize services,

o the DS field value (DS codepoint, or DSCP) in the IP header used to mark packets to select a per-hop behavior, and

o the particular implementation mechanisms that realize a per-hop behavior.

This document focuses on PHBs and the usage of DSCPs to obtain those behaviors. In a network node’s forwarding path, the DSCP is used to map a packet to a particular forwarding treatment, or per-hop behavior (PHB) that specifies the forwarding treatment.

A per-hop behavior (PHB) is a description of the externally observable forwarding behavior of a network node for network traffic marked with a DSCP that selects that PHB. In this context, "forwarding behavior" is a general concept - for example, if only one DSCP is used for all traffic on a link, the observable forwarding behavior (e.g., loss, delay, jitter) will often depend only on the relative loading of the link. To obtain useful behavioral differentiation,multiple traffic subsets are marked with different DSCPs for different PHBs for which node resources such as buffer space and bandwidth are allocated. PHBs provide the framework for a

DiffServ network node to allocate resources to traffic subsets, with network-scope differentiated services constructed on top of this basic hop-by-hop (per-node) resource allocation mechanism.

The codepoints (DSCPs) may be chosen from a small set of fixed values (the class selector codepoints), or from a set of recommended values defined in PHB specifications, or from values that have purely local meanings to a specific network that supports DiffServ; in general, packets may be forwarded across multiple such networks between source and destination.

The mandatory DSCPs are the class selector code points as specified in [RFC2474]. The class selector codepoints (CS0-CS7) extend the deprecated concept of IP Precedence in the IPv4 header; three bits are added, so that the class selector DSCPs are of the form 'xxx000'. The all-zero DSCP ('000000' or CS0) designates a Default PHB that provides best-effort forwarding behavior and the remaining class selector code points were originally specified to provide relatively better per-hop-forwarding behavior in increasing numerical order, but:

o There is no requirement that any two adjacent class selector codepoints select different PHBs; adjacent class selector codepoints may use the same pool of resources on each network node in some networks.

o CS1 ('001000') was subsequently recommended for a Lower Effort (LE) PHB and service when such a service is offered by a network [RFC3662]. An LE service forwards traffic with "lower" priority than best effort and can be "starved" by best effort and other "higher" priority traffic. Not all networks offer an LE service. See [RFC3662] for further discussion of the LE PHB and service.

Applications and traffic sources cannot rely upon different class selector codepoints providing differentiated services or upon the presence of an LE service that is selected by the CS1 DSCP. There is no effective way for a network endpoint to determine whether the CS1 DSCP selects an LE service on a specific network, let alone end-to-end. Packets marked with the CS1 DSCP may be forwarded with best effort service or another "higher" priority service, see [RFC2474].

2.3. Diffserv PHBs (Per-Hop Behaviors)

Although Differentiated Services is a general architecture that may be used to implement a variety of services, three fundamental forwarding behaviors (PHBs) have been defined and characterized for general use. These are:

1. Default Forwarding (DF) for elastic traffic [RFC2474]. The Default PHB is always selected by the all-zero DSCP.

2. Assured Forwarding (AF) [RFC2597] to provide differentiated service to elastic traffic. Each instance of the AF behavior consists of three PHBs that differ only in drop precedence, e.g., AF11, AF12 and AF13; such a set of three AF PHBs is referred to as an AF class, e.g., AF1x. There are four defined AF classes, AF1x through AF4x, with higher numbered classes expected to receive better forwarding treatment than lower numbered classes.

3. Expedited Forwarding (EF) [RFC3246] intended for inelastic traffic. Beyond the basic EF PHB, the VOICE-ADMIT PHB [RFC5865] is an admission controlled variant of the EF PHB.

2.4. DiffServ, Reordering and Transport Protocols

[Editor’s note: This section and the recommendations in Section 4 are centered on TCP, UDP, and SCTP. They could use generalization to include other transport protocols - DCCP is a likely one to include, although it is not necessary to include every known transport protocol.]

Transport protocols provide data communication behaviors beyond those possible at the IP layer. An important example is that TCP provides reliable in-order delivery of data with congestion control. SCTP provides additional properties such as preservation of message boundaries, and the ability to avoid head-of-line blocking that may occur with TCP. In contrast, UDP is a basic unreliable datagram protocol that provides port-based multiplexing and demultiplexing on top of IP.

Transport protocols that provide reliable delivery (e.g., TCP, SCTP) are sensitive to network reordering of traffic. When a protocol that provides reliable delivery receives a packet other than the next expected packet for an ordered connection or stream, it usually assumes that the expected packet has been lost and respond with a retransmission request for that packet. In addition, congestion control functionality in transport protocols usually infers congestion when packets are lost, creating an additional sensitivity to significant reordering - such reordering may be (mis-)interpreted as indicating congestion-caused packet loss, causing a reduction in transmission rate. This remains true even when ECN [RFC3168] is in use, as ECN receivers are required to treat missing packets as potential indications of congestion. This requirement is based on two factors:

- o Severe congestion may cause ECN-capable network nodes to drop packets, and

- o ECN traffic may be forwarded by network nodes that do not support ECN and hence use packet drops to indicate congestion.

Congestion control is an important aspect of the Internet architecture, see [RFC2914] for further discussion.

In general, marking packets with different DSCPs results in different PHBs being applied at network nodes, making reordering possible due to use of different pools of forwarding resources for each PHB. The primary exception is that reordering is prohibited within each AF class (e.g., AF1x), as the three PHBs in an AF class differ solely in drop precedence. Reordering within a PHB or AF class may occur for other transient reasons (e.g., route flap or ECMP rebalancing).

UDP is the primary transport protocol that is not sensitive to reordering in the network, because it does not provide reliable delivery or congestion control. On the other hand, when UDP is used to encapsulate other protocols (e.g., as is the case for RTCWEB, see Section 2.1), the reordering considerations for the encapsulated protocols apply. For RTCWEB example in particular, every encapsulated protocol (i.e., RTP, SCTP and TCP) is sensitive to reordering as further discussed in this document.

2.5. DiffServ, Reordering and Real-Time Communication

Real-time communications are also sensitive to network reordering of packets. Such reordering may lead to spurious NACK generation and unneeded retransmission, as is the case for reliable delivery protocols (see Section Section 2.4). The degree of sensitivity depends on protocol or stream timers, in contrast to reliable delivery protocols that usually react to all reordering.

Receiver jitter buffers have important roles in the effect of reordering on real time communications:

- o Minor packet reordering that is contained within a jitter buffer usually has no effect on rendering of the received RTP stream.

- o Packet reordering that exceeds the capacity of a jitter buffer can cause user-perceptible quality problems (e.g., glitches, noise) for delay sensitive communication, such as interactive conversations. Interactive real-time communication implementations often choose to discard data that is sufficiently late to prevent it from being rendered in source stream order, making retransmission counterproductive. For this reason, implementations of interactive real-time communication often do not use retransmission.

- In contrast, replay of recorded media can typically uses significantly larger jitter buffers than can be tolerated for interactive conversations, with the result that replay is more tolerant to reordering than interactive conversations. The size of the jitter buffer imposes an upper bound on replay tolerance to reordering, but does enable retransmission to be used when the jitter buffer is significantly larger than the amount of data that will arrive during the round-trip latency for retransmission.

Network packet reordering caused by use of different DSCPs has no effective upper bound, and can exceed the size of any reasonable jitter buffer - in practice, the size of jitter buffers for replay is limited by external factors such as the amount of time that a human is willing to wait for replay to start.

2.6. Traffic Classifiers and DSCP Remarking

DSCP markings are not end-to-end in general. Each network can make its own decisions about what PHBs to use and which DSCP maps to each PHB. While every PHB specification includes a recommended DSCP, and RFC 4594 [RFC4594] recommends their end-to-end usage, there is no requirement that every network support any PHBs or use any DSCPs, with the exception of the class selector codepoint requirements in RFC 2474 [RFC2474]. When DiffServ is used, the edge or boundary nodes of a network are responsible for ensuring that all traffic entering that network conforms to that network’s policies for DSCP and PHB usage, and such nodes remark traffic (change the DSCP marking as part of traffic conditioning) accordingly. As a result, DSCP remarking is possible at any network boundary, including the first network node that traffic sent by a host encounters. Remarking is also possible within a network, e.g., for traffic shaping.

DSCP remarking is part of traffic conditioning; the traffic conditioning functionality applied to packets at a network node is determined by a traffic classifier [RFC2475]. Edge nodes of a DiffServ network classify traffic based on selected packet header fields; typical implementations do not look beyond the traffic’s 5-tuple in the IP and transport protocol headers. As a result, when multiple DSCPs are used for traffic that shares a 5-tuple, remarking at a network boundary may result in all of the traffic being forwarded with a single DSCP, thereby removing any differentiation within the 5-tuple downstream of the remarking location. Network nodes within a DiffServ network generally classify traffic based solely on DSCPs, but may perform finer grain traffic conditioning similar to that performed by edge nodes.

So, for two arbitrary network endpoints, there can be no assurance that the DSCP set at the source endpoint will be preserved and presented at the destination endpoint. On the contrary, it is quite likely that the DSCP will be set to zero (e.g., at the boundary of a network operator that distrusts or does not use the DSCP field) or to a value deemed suitable by an ingress (MF) classifier for whatever 5-tuple it carries. DiffServ classifiers generally ignore embedded protocol headers (e.g., for SCTP or RTP embedded in UDP, classification will be only on the outer UDP header).

In addition, remarking may remove application-level distinctions in forwarding behavior - e.g., if multiple PHBs within an AF class are used to distinguish different types of frames within a video RTP stream, token-bucket-based remarkers operating in Color-Blind mode (see [RFC2697] and [RFC2698] for examples) may remark solely based on flow rate and burst behavior, removing the drop precedence distinctions specified by the source.

Backbone and other carrier networks may employ a small number of DSCPs (e.g., less than half a dozen) in order to manage a small number of traffic aggregates; hosts that use a larger number of DSCPs can expect to find that much of their intended differentiation is removed by such networks. Better results may be achieved when DSCPs are used to spread traffic among a smaller number of DiffServ-based traffic subsets or aggregates, see [I-D.geib-tsvwg-diffserv-intercon] for one proposal. This is of particular importance for MPLS-based networks due to the limited size of the Traffic Class (TC) field in an MPLS label [RFC5462] that is used to carry DiffServ information and the use of that TC field for other purposes, e.g., ECN [RFC5129]. For further discussion on use of DiffServ with MPLS, see [RFC3270] and [RFC5127].

3. RTP Multiplexing Background

Section 2 explains how source streams can be multiplexed over RTP sessions which can in turn be multiplexed over UDP with packets generated by other transport protocols. This section provides background on why this level of multiplexing is desirable. The rationale in this section applies both to multiplexing of source streams in RTP sessions and multiplexing of an RTP session with traffic from other transport protocols via UDP encapsulation.

Multiplexing reduces the number of ports utilized for real-time and related communication in an overall interaction. While a single endpoint might have plenty of ports available for communication, this traffic often traverses points in the network that are constrained on the number of available ports. A good example is a NAT/FW device sitting at the network edge. As the number of simultaneous protocol sessions increases, so does the burden placed on these devices in order to provide port mapping.

Another reason for multiplexing is to help reduce the time required to establish bi-directional communication. Since any two communicating users might be situated behind different NAT/FW devices, it is necessary to employ techniques like STUN/ICE/TURN in order to get traffic to flow between the two devices [I-D.ietf-rtcweb-transports]. Performing the tasks required of STUN/ICE/TURN take time and requiring an endpoint to perform these tasks for multiple protocol sessions can increase the time required. While tasks for different sessions can be performed in parallel, it is nonetheless necessary for applications to wait for all sessions to be opened before communication between to users can begin. Reducing the number of STUN/ICE/TURN steps reduces the probability of losing a packet and introducing delay in setting up a communication session. Further, reducing the number of STUN/ICE/TURN tasks means that there is a lower burden placed on the STUN and TURN servers.

Multiplexing may reduce the complexity and resulting load on an endpoint. A single instance of STUN/ICE/TURN is simpler to execute and manage than multiple instances STUN/ICE/TURN operations happening in parallel, as the latter require synchronization and create more complex failure situations that have to be cleaned up by additional code.

4. Recommendations

The only standardized use of multiple PHBs and DSCPs that avoids network reordering among packets marked with different DSCPs is use of PHBs within a single AF class. All other uses of multiple PHBs and/or the class selector DSCPs allow network reordering of packets that are marked with different DSCPs. Based on this and the foregoing discussion, the following requirements apply to use of DiffServ with real-time communications - applications and other traffic sources:

- SHOULD NOT use different PHBs and DSCPs that may cause reordering within a single RTP stream. If this is not done, significant network reordering may overwhelm implementation assumptions about limits on reordering, e.g., jitter buffer size, causing poor user experiences, see Section Section 2.5 above.

- SHOULD NOT use different PHBs and DSCPs that may cause reordering within an ordered session for a reliable transport protocol (e.g., TCP, SCTP). Receivers for such protocols interpret reordering as indicating loss of out-of-order packets causing undesired retransmission requests, and will infer congestion from significant reordering, causing throughput reduction. This requirement applies to both unencapsulated and encapsulated (e.g., via UDP) uses of reliable transport protocols.

- MAY use different PHBs and DSCPs that cause reordering within a single UDP 5-tuple, subject to the above constraints. The service differentiation provided by such usage is unreliable, as it may be removed at network boundaries for the reasons described in Section 2.6 above.

- MUST NOT rely on end-to-end preservation of DSCPs as network node remarking can change DSCPs and remove drop precedence distinctions see Section 2.6 above. For example, if a source uses drop precedence distinctions within an AF class to identify different types of video frames, using those DSCP values at the receiver to identify frame type is inherently unreliable.

- SHOULD use the CS1 codepoint only for traffic that is acceptable to forward as best effort traffic, as network support for use of CS1 to select a "less than best effort" PHB is inconsistent. Further, some networks may treat CS1 as providing "better than best effort" forwarding behavior.

There is no requirement in this document for network operators to differentiate traffic in any fashion. Networks may support all of the PHBs discussed herein, classify EF and AFxx traffic identically, or even remark all traffic to best effort at some ingress points. Nonetheless, it is useful for network endpoints to provide finer granularity DSCP marking on packets for the benefit of networks that offer QoS service differentiation. A specific example is that traffic originating from a browser may benefit from QoS service differentiation in within-building and residential access networks, even if the DSCP marking is subsequently removed or simplified. This is because such networks and the boundaries between them are likely traffic bottleneck locations (e.g., due to customer aggregation onto common links and/or speed differences among links used by the same traffic).

5. Examples

For real-time communications, one might want to mark the audio packets using EF and the video packets as AF41. However, in a video conference receiving the audio packets ahead of the video is not useful because lip sync is necessary between audio and video. It may still be desirable to send audio with a PHB that provides better service, because early arrival of audio helps assure smooth audio rendering, which is often more important than fully faithful video rendering. There are also limits, as some devices have difficulties in synchronizing voice and video when packets that need to be rendered together arrive at significantly different times. It makes more sense to use different PHBs when the audio and video source streams do not share a strict timing relationship. For example, video content may be shared within a video conference via playback, perhaps of an unedited video clip that is intended to become part of a television advertisement. Such content sharing video does not need precise synchronization with video conference audio, and could use a different PHB, as content sharing video is more tolerant to jitter, loss, and delay.

Within a layered video RTP stream, ordering of frame communication is preferred, but importance of frame types varies, making use of PHBs with different drop precedences appropriate. For example, I-frames that contain an entire image are usually more important than P-frames that contain only changes from the previous image because loss of a P-frame (or part thereof) can be recovered (at the latest) via the next I-frame, whereas loss of an I-frame (or part thereof) may cause rendering problems for all of the P-frames that depend on the I-frame. For this reason, it is appropriate to mark I-frame packets with a PHB that has lower drop precedence than the PHB used for P-frames, as long as the PHBs preserve ordering among frames (e.g., are in an AF class) - AF41 for I-frames and AF43 for P-frames is one possibility. Additional spatial and temporal layers beyond the base video layer could also be marked with higher drop precedence than the base video layer, as their loss reduces video quality, but does not disrupt video rendering.

Additional RTP streams in a real-time communication interaction could be marked with CS0 and carried as best effort traffic. One example is real-time text transmitted as specified in RFC 4103[RFC4103]; best effort forwarding suffices when redundancy encoding is used (as required by RFC 4103). Best effort forwarding suffices because such real-time text has loose timing requirements; RFC 4103 recommends sending text in chunks every 300ms. Such text is technically real-time, but does not need a PHB promising better service than best effort, in contrast to audio or video.

6. IANA Considerations

This document includes no request to IANA.

7. Security Considerations

The security considerations for all of the technologies discussed in this document apply; in particular see the security considerations for RTP in [RFC3550] and DiffServ in [RFC2474] and [RFC2475].

Multiplexing of multiple protocols onto a single UDP 5-tuple via encapsulation has implications for network functionality that is based on monitoring or inspection of individual protocol flows, e.g., firewalls and traffic monitoring systems. When implementations of such functionality lack visibility into encapsulated traffic (likely for many current implementations), it may be difficult or impossible to apply network security policy and controls at a finer grain than the overall UDP 5-tuple.

Use of multiple DSCPs to provide differentiated QoS service may reveal information about the encrypted traffic to which different service levels are provided. For example, DSCP-based identification of RTP streams combined with packet frequency and packet size could reveal the type or nature of the encrypted source streams. The IP header used for forwarding has to be unencrypted for obvious reasons, and the DSCP likewise has to be unencrypted in order to enable different IP forwarding behaviors to be applied to different packets. The nature of encrypted traffic components can be disguised via encrypted dummy data padding and encrypted dummy packets, e.g., see the discussion of traffic flow confidentiality in [RFC4303]. Encrypted dummy packets could even be added in a fashion that an observer of the overall encrypted traffic might mistake for another encrypted RTP stream.

8. Acknowledgements

This document is the result of many conversations that have occurred within multiple working groups in the RAI and Transport areas. Many thanks to Harald Alvestrand, Erin Bournival, Brian Carpenter, Ruediger Geib and James Polk for their reviews and input.

9. References

9.1. Normative References

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

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

[RFC2597] Heinanen, J., Baker, F., Weiss, W., and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.

[RFC3246] Davie, B., Charny, A., Bennet, J., Benson, K., Le Boudec, J., Courtney, W., Davari, S., Firoiu, V., and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002.

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

[RFC5865] Baker, F., Polk, J., and M. Dolly, "A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic", RFC 5865, May 2010.

9.2. Informative References

[I-D.geib-tsvwg-diffserv-intercon]
Geib, R., "DiffServ interconnection classes and practice", draft-geib-tsvwg-diffserv-intercon-05 (work in progress), February 2014.

[I-D.ietf-avtext-rtp-grouping-taxonomy]
Lennox, J., Gross, K., Nandakumar, S., and G. Salgueiro, "A Taxonomy of Grouping Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", draft-ietf-avtext-rtp-grouping-taxonomy-02 (work in progress), June 2014.

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-10 (work in progress), June 2014.

[I-D.ietf-rtcweb-rtp-usage]
Perkins, C., Westerlund, M., and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP", draft-ietf-rtcweb-rtp-usage-15 (work in progress), May 2014.

[I-D.ietf-rtcweb-transports]
Alvestrand, H., "Transports for RTCWEB", draft-ietf-rtcweb-transports-05 (work in progress), June 2014.

[I-D.petithuguenin-avtcore-rfc5764-mux-fixes]
Petit-Huguenin, M. and G. Salgueiro, "Multiplexing Scheme Updates for Secure Real-time Transport Protocol (SRTP) Extension for Datagram Transport Layer Security (DTLS)", draft-petithuguenin-avtcore-rfc5764-mux-fixes-00 (work in progress), July 2014.

[I-D.westerlund-avtcore-transport-multiplexing]
Westerlund, M. and C. Perkins, "Multiplexing Multiple RTP Sessions onto a Single Lower-Layer Transport", draft-westerlund-avtcore-transport-multiplexing-07 (work in progress), October 2013.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[RFC2697] Heinanen, J. and R. Guerin, "A Single Rate Three Color Marker", RFC 2697, September 1999.

[RFC2698] Heinanen, J. and R. Guerin, "A Two Rate Three Color Marker", RFC 2698, September 1999.

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

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

[RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, May 2002.

[RFC3662] Bless, R., Nichols, K., and K. Wehrle, "A Lower Effort Per-Domain Behavior (PDB) for Differentiated Services", RFC 3662, December 2003.

[RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, June 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, August 2006.

[RFC5109] Li, A., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, December 2007.

[RFC5127] Chan, K., Babiarz, J., and F. Baker, "Aggregation of Diffserv Service Classes", RFC 5127, February 2008.

[RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, January 2008.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, February 2009.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, May 2010.

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

[W3C.WD-mediacapture-streams-20130903] Burnett, D., Bergkvist, A., Jennings, C., and A. Narayanan, "Media Capture and Streams", World Wide Web Consortium WD WD-mediacapture-streams-20130903, September 2013, <http://www.w3.org/TR/2013/WD-mediacapture-streams-20130903>.

Appendix A. Change History

[To be removed before RFC publication.]

Changes from draft-york-dart-dscp-rtp-00 to -01

o Added examples (Section 5)

o Reworked text on RTP session multiplexing, at most one RTP session can be used per UDP 5-tuple.

o Initial terminology alignment with RTP grouping taxonomy draft.

o Added Section 2.5 on real-time communication interaction w/ reordering based on text from Harald Alvestrand.

o Strengthened warnings on loss of differentiation, but indicate that differentiation may still be useful from source to point of loss.

o Added a few sentences on DiffServ and MPLS.

o Added discussion of UDP-encapsulated protocols that are reordering sensitive.

o Added initial security considerations.

- o Many editorial changes

Changes from draft-york-dart-dscp-rtp-01 to -02

- o More terminology alignment with RTP grouping taxonomy draft: "RTP packet stream" -> "RTP stream"

- o Aligned terminology for less-than-best-effort with RFC 3662 - LE (Lower Effort) PHB and service

- o Minor reference updates

Authors' Addresses

Dan York
Internet Society
Keene, N.H.
USA

Phone: +1-802-735-1624
Email: dyork@lodestar2.com

David Black (editor)
EMC
176 South Street
Hopkinton, MA 01748
USA

Phone: +1 508 293-7953
Email: david.black@emc.com

Cullen Jennings
Cisco
170 West Tasman Drive
MS: SJC-21/2
San Jose, CA 95134
USA

Phone: +1 408 421-9990
Email: fluffy@cisco.com

Paul Jones
Cisco
7025 Kit Creek Road
Research Triangle Park, MA 27502
USA

Phone: +1 919 476 2048
Email: paulej@packetizer.com