

TSVWG Working Group
Internet-Draft
Intended status: Informational
Expires: April 17, 2020

L. Han
Y. Qu
L. Dong
R. Li
Futurewei Technologies
T. Nadeau
Lucid Vision
K. Smith
Vodafone
J. Tantsura
Apstra
October 15, 2019

Resource Reservation Protocol for IP Transport QoS
draft-han-tsvwg-ip-transport-qos-03

Abstract

IP is designed for use in Best Effort Networks, which are networks that provide no guarantee that data is delivered, or that delivery meets any specified quality of service parameters. However there are new applications requiring IP to provide deterministic services in terms of bandwidth and latency, such as network based AR/VR (Augmented Reality and Virtual Reality), industrial internet. This document proposes a solution in IPv6 that can be used by transport layer protocols to guarantee certain level of service quality. This new service is fined-grained and could apply to individual or aggregated TCP/UDP flow(s).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on April 17, 2020.

Internet-Draft

ResRev for IP QoS

October 2019

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	Overview	6
3.1.	Design Targets	6
3.2.	Scope and Assumptions	7
3.3.	Sub-layer in IP for Transport Control	7
3.4.	IP In-band signaling	8
3.5.	IPv6 Approach	10
4.	Key Messages and Parameters	11
4.1.	Setup and Setup State Report messages	11
4.2.	Forwarding State and Forwarding State Report messages	12
4.3.	Hop Number	13
4.4.	Flow Identifying Method and Service ID	13
4.5.	QoS State and life of Time	14
4.6.	Authentication	14
5.	Packet Forwarding	15
5.1.	Basic Hardware Capability	15
5.2.	Flow Identification in Packet Forwarding	16
5.3.	QoS Forwarding State Detection and Failure Handling . . .	16
6.	Details of Working with Transport Layer	17
6.1.	Working with TCP	17
6.2.	Working with UDP and other Protocols	20
7.	Additional Considerations	20
7.1.	User and Application driven	20
7.2.	Traffic Management in Host	21
7.3.	Heterogeneous Network	22

7.4.	Proxy Control	22
8.	IANA Considerations	22
9.	Security Considerations	24
10.	References	25
10.1.	Normative References	25

10.2.	Informative References	26
Appendix A.	Acknowledgements	28
Appendix B.	Message Objects	29
B.1.	Setup State Object	29
B.2.	Bandwidth Object	31
B.3.	Burst Msg	31
B.4.	Latency Object	32
B.5.	Authentication Object	32
B.6.	OAM Object	33
B.7.	Forwarding State Object	34
B.8.	Setup State Report Object	34
B.9.	Forward State Report Object	35
	Authors' Addresses	36

[1.](#) Introduction

Recently, more and more new applications for The Internet are emerging. These applications have a number of key requirements that are common to all such as that their required bandwidth is very high and/or latency is very low compared with traditional applications like most of web and video applications.

For example, network based Augmented Reality (AR) or Virtual Reality (VR) applications may need hundreds of Mbps bandwidth (throughput) and a low single digit millisecond latency. Moreover, the difference between mean bit rate and peak bit rate may be significant due to the choice of compression algorithm

[\[I-D.han-iccr-g-arvr-transport-problem\]](#). This may result in large bursts, and make traffic management more difficult.

Some future applications may expect networks to provide a bounded latency service. One such example is tactile network [\[Tactile\]](#).

With the technology development in 5G [\[HU5G\]](#)[\[QU2016\]](#) and beyond, the wireless access network is also increasing the demand for the Ultra-Reliable and Low-Latency Communications (URLLC). This also leads to

the question of whether IP can provide such service in an Evolved Packet Core (EPC) [[EPC](#)] network. IP is becoming more and more important in the EPC when the Multi-access Edge Computing (MEC) [[MEC](#)] for 5G requires the cloud and data service to move closer to eNodeB [[eNodeB](#)].

[I-D.ietf-detnet-use-cases] identifies some use cases from different industries which have a common need for "deterministic flows". Such flows require guaranteed bandwidth and bounded latency.

Traditionally, an IP network provides an unreliable or best-effort datagram service over a collection of underlying networks (i.e.:

ethernet, ATM, etc...). Integrated services(IntServ) [[RFC3175](#)] specifies a fine-grained QoS system, which requires all routers along the traffic path to support it and maintain the states for resource reserved IP flow(s), so it is difficult to scale up to keep track of all the reservations. Differentiated services (DiffServ) [[RFC2475](#)] specifies a simple and scalable mechanism to classify traffic and provide more coarse QoS, however because it can only specify per-hop behaviors (PHBs), and how individual routers deal with the DS [[RFC2474](#)] field is configuration specific. It is difficult to provide consistent resource reservation for specified class of traffic, thus hard to support the end-to-end bandwidth or latency guarantee.

The transport layer (TCP/UDP) on top of IP is based on the best-effort-only service, which has influenced the transport layer evolution for quite long time, and results in some widely accepted assumptions and solutions, such as:

1. The IP layer can only provide basic P2P (point to point) or P2MP (point to multi-point) end-to-end connectivity in the Internet, but the connectivity is not reliable and does not guarantee any quality of service to end-user or application, such as bandwidth, packet loss, latency etc. Due to this assumption, the transport layer or application must have its own control mechanism in congestion and flow to obtain the reliable and satisfactory service to cooperate with the under layer network quality.
2. The transport layer assumes that the IP layer can only process all IP flows equally in the hardware since the best effort

service is actually an un-differentiated service. The process includes scheduling, queuing and forwarding. Thus, the transport layer must behave nicely and friendly to make sure all flows will only obtain its own faired share of resource, and no one could consume more and no one could be starved.

This document proposes a new IP transport service that guarantees bandwidth and latency for new applications. The scope and criteria for the new technology will also be discussed. This new IP transport service is designed to be supplementary to regular IP transport services, only meant to be used for special applications that are bandwidth and/or latency sensitive.

2. Terminology

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

Han, et al.

Expires April 17, 2020

[Page 4]

Internet-Draft

ResRev for IP QoS

October 2019

14 [RFC 2119](#) [[RFC2119](#)] [RFC 8174](#) [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

Abbreviations used in this documents:

E2E

End-to-end.

EH

IPv6 Extension Header or Extension Option.

QoS

Quality of Service.

OAM

Operation and Management.

In-band Signaling

In telecommunications, in-band signaling is sending control information within the same band or channel used for voice or video.

Out-of-band Signaling

out-of-band signaling is that the control information sent over a different channel, or even over a separate network.

IP flow

For non-IPSec, an IP flow is identified by the source, destination IP address, the protocol number, the source and destination port number.

IP path

An IP path is the route that IP flow will traverse. It could be the shortest path determined by routing protocols (IGP or BGP), or the explicit path such as segment routing [[I-D.ietf-spring-segment-routing](#)].

QoS channel

A forwarding channel that is QoS guaranteed. It provides additional QoS service to IP forwarding. A QoS channel can be used for one or multiple IP flows depending on the granularity of in-band signaling.

CIR

Committed Information Rate.

PIR

Peak Information Rate.

HbH-EH

IPv6 Hop-by-Hop Extension Header.

Dst-EH

IPv6 Destination Extension Header.

HbH-EH-aware node

Network nodes that are configured to process the IPv6 Hop-by-Hop Extension Header.

[3.](#) Overview

Semiconductor chip technology has advanced significantly in the last decade, and as such the widely used network processing and forwarding process can now not only forward packets at line speed, but also

easily support other feature processing such as QoS for DiffServ/MPLS, Access Control List (ACL), fire wall, and Deep Packet Inspection (DPI).

This advancement enables network processors to do the general process to handle simple control messages for traffic management, such as signaling for hardware programming, congestion state report, OAM, etc. So now it's possible to treat some TCP/IP flows differently from others and give them specified resource are feasible now by using network processor.

This document proposes a deterministic IP transport service, which can provide guaranteed bandwidth and latency. The solution is based on the QoS implemented in network processor through in-band signaling.

[3.1.](#) Design Targets

The proposed transport service is expected to satisfy the following criteria:

- o End user or application may directly use the new service.
- o The new service can coexist with the current transport service and is backward compatible.
- o Service providers can manage the new service.
- o Performance and scalability targets of this new service are practical for vendors to achieve.
- o The new service is transport agnostic. TCP, UDP and other transport protocols on top of IP can use it.

[3.2.](#) Scope and Assumptions

The initial aim is to propose a solution for IPv6. To limit the scope of the document and simplify the design and solution, the following constraints are given:

1. The new service with QoS is aimed to be supplementary to regular IP service. It is targeted for the applications that are

bandwidth and/or latency sensitive. It is not intended to replace the TCP/IP variants that have been proved to be efficient and successful for current applications.

2. The new service is limited within one administrative domain, even it does not exclude the possibilities of extending the mechanism for inter-domain scenarios. Currently only inter-domain security is considered, and the inter-domain SLA, accounting and other issues are not discussed.
3. Due to high bandwidth requirement of new service for individual flow, the total number of the flows with the new service cannot be high for a port, or a system. From another point of view, the new service is targeted for applications that really need it, the number of supported applications/users should be controlled and cannot be unlimited. Hence the scalability requirement for the new service is limited.
4. The new service must be able to coexist with the regular transport service in the same hardware, and be backward compatible. Also, a transport flow can switch between regular transport and new service without service interruption.

[3.3.](#) Sub-layer in IP for Transport Control

In order to provide some new features for the layer above IP, it is very useful to introduce an additional sub-layer, Transport Control, between layer 3 (IP) and layer 4 (TCP/UDP). The new layer belongs to IP, and is present only when the system needs to provide extra control for the upper layer, in addition to the normal IP forwarding. Fig 1. illustrates a new stack with the sub-layer.

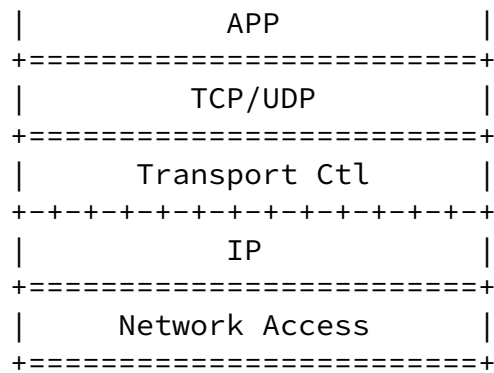


Figure 1: The new stack with a sub-layer in Layer 3

The new sub-layer is always bound with IP layer and can provide support of the features for upper layer, such as:

In-band Signaling

The IP header with the new sub-layer can carry the signaling information for the devices on the IP path. The information may include all QoS related parameters used for hardware programming.

Congestion control

The congestion state in each device on the path can be detected and notified to the source of flows by the sub-layer; The dynamic congestion control instruction can also be carried by the sub-layer and examined by network devices on the IP path.

IP Path OAM

The OAM instruction can be carried in the sub-layer, and the OAM state can be notified to the source of flows by the sub-layer. The OAM includes the path and device property detection, QoS forwarding diagnosis and report.

IPv4 could use the IP option for the purpose of the sub-layer. But due to the limit size of the IP option, the functionalities, scalability of the layer is restricted.

IPv6 can realize the sub-layer easily using IPv6 extension header [[RFC8200](#)]. The document will focus on the solution for IPv6 using different IPv6 extension headers.

[3.4.](#) IP In-band signaling

In-band signaling messages are carried along with the payload. It is guaranteed that the signaling follows the same path as the data flow, and this can bring up some advantages that other methods can hardly provide:

Diagnosis

The in-band signaling message takes the same path, same hops, same processing at each hop as the data packet, this will make the diagnosis for both signaling and data path easier.

Simplicity

The in-band signaling message is forwarded with the normal data packet, it does not need to run a separate protocol. This will dramatically reduce the complexity of the control.

Performance and scalability

Due to the simplicity of in-band signaling for control, it is easier to provide a better performance and scalability for a new future.

There have been similar works done or proposed in the industry for quite some time. The in-band QoS signaling for IPv6 was discussed by Lawrence Roberts in 2005 [[I-D.roberts-inband-qos-ipv6](#)]. The requirements of IP in-band signaling was proposed by Jon Harper in 2007 [[I-D.harper-inband-signalling-requirements](#)]. Telecommunications Industry Association (TIA) published a standard for "QoS Signaling for IP QoS Support and Sender Authentication" in 2006 [[TIA](#)].

This document proposes an optimized solution for QoS service using in-band signaling, and it also tries to address issues raised by previous proposals, such as security, scalability and performance.

The major differences from the previous works are:

1. Focus on IPv6 only.
2. The proposed solution could be driven by end-user operating system's protocol stack such as TCP, UDP or other protocols, or by network device working as a proxy.
3. Simplified signaling process with minimal information carried, reduced QoS state maintenance at network devices.
4. Use different IPv6 options for signaling and signaling state report.
5. Support both bandwidth reservation and latency expectation at each hop.
6. Support dynamic resource reservation.

7. Support dynamic QoS forwarding state monitoring.

[3.5.](#) IPv6 Approach

IPv6 extension header is used for signaling. There are two types of extension header used for the purpose of transport QoS control, one is the hop-by-hop EH (HbH-EH) and another is the destination EH (Dst-EH).

The HbH-EH may be examined and processed by the nodes that are explicitly configured to do so [[RFC8200](#)], and these nodes are called HbH-EH-aware nodes. Note, not all nodes along a patch need to HbH-EH-aware. HbH-EH is used to carry the QoS requirement for dedicated flow(s) and then the information is intercepted by HbH-EH-aware nodes on the path to program hardware accordingly.

The destination EH will only be examined and processed by the destination device that is associated with the destination IPv6 address in the IPv6 header. This EH is used to send the QoS related report information directly to the source of the signaling at other end.

The following figure illustrates the path setup process:

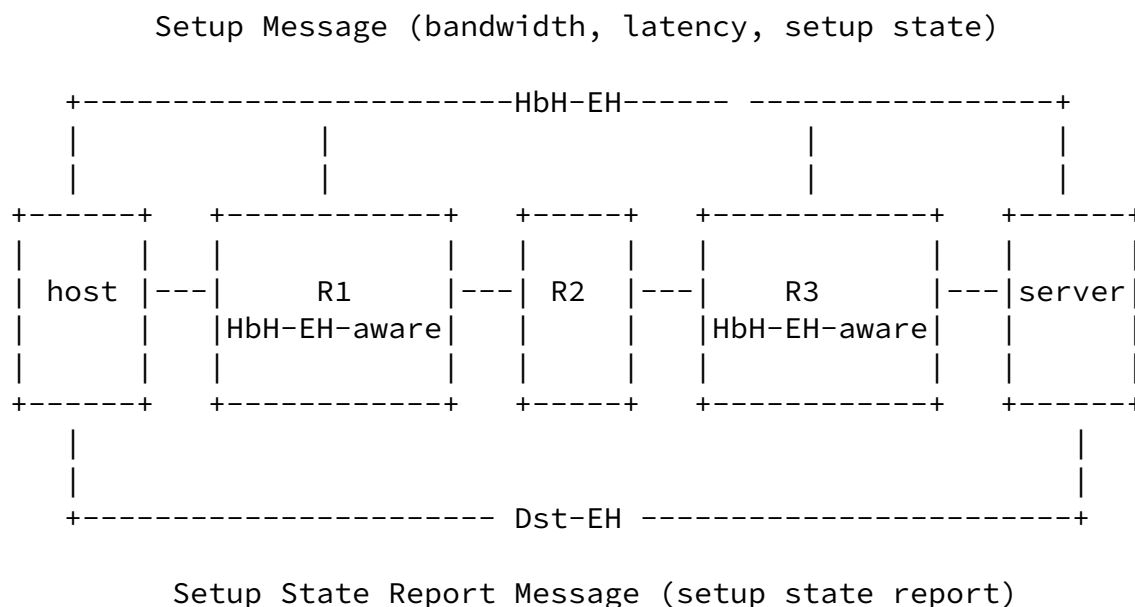


Figure 2: Path Setup

Using the figure.2 for illustration, to set up a path with resource reservation, a setup message including QoS requirements, such as max/min bandwidth, burst size, the latency, and the setup state is sent from the host to the server. After each HbH-EH-aware node along the path receives the message, it reads the QoS information and programs the hardware for resource reservation, queuing management etc. The setup state object is updated at each HbH-EH-aware node to include

the QoS programming and provisioning result and the necessary hardware reference information for IP forwarding with QoS. After the setup message reaches the server, the server will send a setup state report message encoded as Dst-EH to the host. The setup state report message carries the path setup results from the setup state object.

[4.](#) Key Messages and Parameters

[4.1.](#) Setup and Setup State Report messages

Setup message is intended to program the hardware for QoS channel on the IP path from the source to the destination expressed in IPv6 header. It is embedded as the HbH-EH in an IPv6 packet and will be processed at each HbH-EH-aware node. For the simplicity, performance and scalability purpose, not all routers along the path need do the processing or be HbH-EH-aware. For different QoS requirements and scenarios, different criteria can be used to configure HbH-EH-aware nodes.

A throttle router is the device that an interested TCP/UDP session cannot get the enough bandwidth to support its application, and it will also contribute more to latency than non-throttle routers. The regular throttle routers include the BRAS (broadband remote access server) in broadband access network, the PGW (PDN Gateway) in LTE network etc. In more general case, any routers which aggregated traffic may become as a throttle router. Throttle routers should be configured to process HbH-EH when:

- o Reserved bandwidth is required: The throttle router is the critical point to be configured to process the hop-by-hop EH for the bandwidth reservation. Moreover, the direction of congestion must be considered.

- o Bounded latency is required: In theory, each router and switch could contribute some delay to the end-to-end latency, but the throttle router will contribute more than non-throttle routers, and slow device will contribute more than fast device. We can use OAM to detect the latency contribution in a network, and configure those worst-case devices to process the HbH-EH.

Setup State Report message is the message sent from the destination host to the source host (from the point of view of the Setup message). The message is embedded into the Dst-EH in any data packet. The Setup State Report in the message is just a copy from the Setup message received at the destination host for a typical TCP session. The message is used at the source host to forward the packet later and to do the congestion control.

```

<Setup Message> ::= <Setup State Object> [ <Bandwidth
Object> ]
                        [ <Burst Object> ] [ <Latency
Object> ]
                        [ <OAM Object> ] [ <Authentication
Object> ]
<Setup State Report Message> ::= <Setup State Report
Object>
                                [ <OAM Object> ]

```

[4.2.](#) Forwarding State and Forwarding State Report messages

After the QoS is programmed by the in-band signaling, the specified IP flows can be processed and forwarded for the QoS requirement. There are two ways for host to use the QoS channel for associated TCP session:

1. Host directly send the IP packet without any changes to the packet, this is for the following cases:
 - * The hardware was programmed to use the tuples in IP header as identification for QoS process (SIS = 0), and
 - * The packet does not function to collect the QoS forwarding state on the path.

2. Host add the Forward State message into a data packet's IP header as HbH-EH and send the packet, this is for the cases:

- * The hardware was programmed to use the Service ID as identification for QoS process (SIS != 0).
- * The hardware was programmed to use the tuples in IP header as identification for QoS process (SIS = 0), and the data packet functions to collect the QoS forwarding state on the path. This is the situation that host wants to detect the QoS forwarding state for the purpose of failure handling (See [section 4.3](#)).

Forwarding State message format is shown in the [Section 6.7](#). It is used to notify the service ID and also update QoS forwarding state for the hops that are HbH-EH-aware nodes.

After Forwarding State message is reaching the destination host, the host is supposed to retrieve it and form a Forwarding State Report message, and carry it in any data packet as the Dst-EH, then send it to the host in the reverse direction.

```
<Forward State Message> ::= <Forward State Object> [
<Latency Object> ]
                                [ <OAM Object> ] [ <Authentication
                                Object> ]
<Forward State Report Message> ::= <Forward State Report
Object>
                                [ <OAM Object> ]
```

[4.3](#). Hop Number

This is the parameter for total number of HbH-EH-aware nodes on the path. It is the field "Hop_num" in Setup message, and is used to locate the bit position for "Setup State" and the "Service ID" in "Service ID List". The value of "Hop_num" must be decremented at each HbH-EH-aware node. At the receiving host of the in-band signaling, the Hop_num must be zero.

The source host must know the exact hop number, and setup the initial

value in the Setup message. The exact hop number can be detected using OAM message.

[4.4.](#) Flow Identifying Method and Service ID

A QoS channel might be enforced for a group of flows or a delicate flow, and flow identifying method means the way of identifying a flow or a group of flows that can use a HW programmed QoS channel. Different levels of flow granularities to support QoS are defined as below:

Flow level

The flow identification could be 5 tuples for non IPSec IPv6 packet: the source, destination IP address, protocol number, source and destination port number, and also could be 3 tuples for IPSec IPv6 packet: the source, destination IP address and the flow label.

Address level In-band Signaling

A flow of packets share the same source, destination IP address, but with different protocol number. This is the scenario that the signaling is for the aggregated flows which have the same source, destination address. i.e, All TCP/UDP flows between the same client and same server (only one address for client and one for server)

Transport level In-band Signaling

Packets share the same source, destination IP address, protocol number, but with different source or destination port number (non-IPSec) or different flow label (IPSec). This could be for the

aggregated TCP or UDP flows that started and terminated at the same IP addresses.

DiffServ level In-band Signaling

Packets share the same DSCP value. This means aggregated differentiated service flows that have the same DSCP value. The DSCP value is determined by the 6 most-significant bits in 8-bits DiffServ field for IPv4 or 8-bits Traffic Class field for IPv6.

There are two ways for flow identifying. One is by tuple or DSCP value in IP header, another is by a local significant number, called

service ID, generated and maintained in a router. When "Service ID Size" (SIS) is zero, it means the "Flow identification method" (FI) is used for both control plane and data plane. When "SIS" is not zero, it means "FI" is only used in signaling of setting up the QoS channel, and the data plane will only use the "Service ID". The use of local generated number to identify flow is to speed up the flow lookup and QoS process for data plane.

The "Service ID List" is a list of "Service ID" for all hops that are HbH-EH-aware nodes on the IP path. When a router receives a HbH-EH, it may generate a service ID for the flow(s) that is defined by the Flow Identifying Method in "FI". Then the router must attach the service ID value to the end of the Service ID List. After the packet reaches the destination host, the Service ID List will be that the 1st router's service ID as the list header, and the last router's service ID as the list tail.

[4.5.](#) QoS State and life of Time

After a router is programmed for a QoS, a QoS state is created. The QoS state life is determined by the "Time" in the Setup message. Whenever there is a packet processed by a QoS state, the associated timer for the QoS state is reset. If the timer of a QoS state is expired, the QoS state will be erased and the associated resource will be released.

In order to keep the QoS state active, a application at source host can send some zero size of data to refresh the QoS state.

When the Time is set to zero, it means the life of the QoS State will be kept until the de-programming message is received.

[4.6.](#) Authentication

The in-band signaling is designed to have a basic security mechanism to protect the integrity of a signaling message. The Authentication message is to attach to a signaling message, the source host

calculates the harsh value of a key and all invariable part of a signaling message (Setup message: ver, FI, R, SIS, P, Time; Bandwidth message, Latency message, Burst message). The key is only known to the hosts and all HbH-EH-aware nodes. The securely distribution of

the key is out the scope of the document.

5. Packet Forwarding

To achieve the required QoS, after the path setup with guaranteed bandwidth there are some requirements to be met during data forwarding. These include the hardware capability, the scheme for the data forwarding, QoS processing, state report, etc.

5.1. Basic Hardware Capability

[Section 4](#) explains how QoS guaranteed path can be set up and the corresponding messages used, however different implementations may vary in details. To achieve the satisfactory targets for performance and scalability, the protocol must be cooperated with capable hardware to provide the desired fine-grained QoS for different transport.

In our experiment to implement the feature for TCP, we used a network processor with traffic management feature. The traffic management can provide the fine-grained QoS for any configured flow(s).

The following capabilities are RECOMMENDED:

1. The in-banding signaling is processed in network processor without punting to controller CPU for help
2. The QoS forwarding state is kept and maintained in network processor without the involvement from controller CPU.
3. The QoS state has a life of a pre-configured time and will be automatically deleted if there is no data packet processed by that QoS state. The timer can be changed on the fly.
4. The data forwarding does not need to be done at the controller CPU, or so called slow path. It is at the same hardware as the normal IP forwarding. For any IP packet, the QoS forwarding is executed first. Normal forwarding will be executed if there is no QoS state associated with the identification of the flow.
5. The QoS forwarding and normal forwarding can be switched on the fly.

The details of data plane and hardware related implementations, such as traffic classification, shaping, queuing and scheduling, are out of scope of this document. The report of [\[NGP\]](#) has given some experiments and results by using commercial hardware.

[5.2.](#) Flow Identification in Packet Forwarding

Flow identification in Packet Forwarding is same as the QoS channel establishment by Setup message. It is to forward a packet with a specified QoS process if the packet is identified to be belonging to specified flow(s).

There are two method used in data forwarding to identify flows:

1. Hardware was programmed to use tuples in IP header implicitly. This is indicated by that the "SIS" is zero or the Service ID is not used. When a packet is received, its tuples are looked up according to the value of "FI". If there is a QoS table has match for the packet, the packet will be processed by the QoS state found in the QoS table. This method does not need any EH added into the data packet unless the data packet function to collect the QoS forwarding state on the path.
2. Hardware was programmed to use service ID to identify flows. This is indicated by that the "SIS" is not zero. When a packet is received, the service ID associated with the hop is retrieved and looked up for the QoS table. If it has match for the packet, the packet will be processed by the QoS state entry found in the QoS table.

[5.3.](#) QoS Forwarding State Detection and Failure Handling

QoS forwarding may fail due to different reasons:

1. Hardware failure in HbH-EH-aware node.
2. IP path change due to link failure, node failure or routing changes; And the IP path change has impact to the HbH-EH-aware node.
3. Network topology change; and the change leads to the changes of HbH-EH-aware nodes.

Application may need to be aware of the service status of QoS guarantee when the application is using a TCP session with QoS. In order to provide such feature, the TCP stack in the source host can detect the QoS forwarding state by sending TCP data packet with

reaches the destination host, the host will copy the forwarding state into a Forwarding State Report message, and send it with another TCP packet (for example, TCP-ACK) in reverse direction to the source host. Thereafter, the source host can obtain the QoS forwarding state on all HbH-EH-aware nodes.

A host can do the QoS forwarding state detection by three ways: on demand, periodically or constantly.

After a host detects that there is QoS forwarding state failure, it can repair such failure by sending another Setup message embedded into a HbH-EH of any TCP packet. This repairing can handle all failure case mentioned above.

If a failure cannot be repaired, host will be notified, and appropriate action can be taken, see [section 7.1](#)

[6.](#) Details of Working with Transport Layer

The proposed new IP service is transport agnostic, which means any transport layer protocol can use it.

[6.1.](#) Working with TCP

Considering TCP as the most widely used transport layer protocol, this document uses TCP as an example of transport protocol to show how it works with the proposed IP service.

The following is the list of messages for signaling and associated data forwarding.

- o Setup: This is for the setup of QoS channel through the IP path.
- o Bandwidth: This is the required bandwidth for the QoS channel. It has minimum (CIR) and maximum bandwidth (PIR).
- o Latency: This is the required latency for the QoS channel, it is the bounded latency for each hop on the path. This is not the end to end latency.

- o Burst: This is the required burst for the QoS channel, it is the maximum burst size.
- o Authentication: This is the security message for a in-band signaling.
- o OAM: This is the Operation and Management message for the QoS channel.

- o Setup State Report: This is the state report of a setup message.
- o Forwarding State: This is the forwarding state message used for data packet.
- o Forwarding State Report: This is the forwarding state report of a QoS channel.

There are three scenarios of QoS signaling for TCP session setup with QoS

1. Upstream: This is for the direction of client to server. A application decides to open a TCP session with upstream QoS (for uploading), it will call TCP API to open a socket and connect to a server. The client host will form a TCP SYN packet with the HbH-EH in the IPv6 header. The EH includes Setup message and Bandwidth message, and optionally Latency, Burst, Authentication and OAM messages. The packet is forwarded at each hop. Each HbH-EH-aware nodes will process the signaling message to finish the following tasks before forwarding the packet to next hop:
 - * Retrieve the QoS parameters to program the Hardware, it includes: FL, Time, Bandwidth, Latency, Burst
 - * Update the field in the EH, it includes: Hop_number, Total_latency, and possibly Service ID List

When the server receives the TCP SYN, the Host kernel will also check the HbH-EH while punting the TCP packet to the TCP stack for processing. If the HbH-EH is present and the Report bit is set, the Host kernel must form a new Setup State Report message, all fields in the message must be copied from the Setup message in the HbH-EH. When the TCP stack is sending the TCP-SYNACK to

the client, the kernel must add the Setup State Report message as a Dst-EH in the IPv6 header. After this, the IPv6 packet is complete and can be sent to wire; When the client receives the TCP-SYNACK, the Host kernel will check the Dst-EH while punting the TCP packet to the TCP stack for processing. If the Dst-EH is present and the Setup State Report message is valid, the kernel must read the Setup State Report message. Depending on the setup state, the client will operate according to description in [section 7.1](#)

2. Downstream: This is for the direction of server to client. A application decides to open a TCP session with downstream QoS (for downloading), it will call TCP API to open a socket and connect to a server. The client host will form a TCP SYN packet with the Dst-EH in the IPv6 header. The EH includes Bandwidth

message, and optionally Latency, Burst messages. The packet is forwarded at each hop. Each hop will not process the Dst-EH. When the server receives the TCP SYN, the Host kernel will check the Dst-EH while punting the TCP packet to the TCP stack for processing. If the Dst-EH is present, the Host kernel will retrieve the QoS requirement information from Bandwidth, Latency and Burst message, and check the QoS policy for the user. If the user is allowed to get the service with the expected QoS, the server will form a Setup message similar to the case of client to server, and add it as the HbH-EH in the IPv6 header, and send the TCP-SYNACK to client. Each HbH-EH-aware nodes on the path from server to client will process the message similar to the case of client to server. After the client receives the TCP-SYNACK, The client will send the Setup State Report message to server as the Dst-EH in the TCP-ACK. Finally the server receives the TC-ACK and Setup State Report message, it can send the data to the established session according to the pre-negotiated QoS requirements.

3. Bi-direction: This is the case that the client wants to setup a session with bi-direction QoS guarantee. The detailed operations are actually a combination of Upstream and Downstream described above.

After a QoS channel is setup, the in-band signaling message can still be exchanged between two hosts, there are two scenarios for this.

1. Modify QoS on the fly: When the pre-set QoS parameters need to be adjusted, the application at source host can re-send a new in-band signaling message, the message can be embedded into any TCP packet as a IPv6 HbH-EH. The QoS modification should not impact the established TCP session and programmed QoS service. Thus, there is no service impacted during the QoS modification. Depending on the hardware performance, the signaling message can be sent with TCP packet with different data size. If the performance is high, the signaling message can be sent with any TCP packet; otherwise, the signaling message should be sent with small size TCP packet or zero-size TCP packet (such as TCP ACK). Modification of QoS on the fly is a very critical feature for the so called "Application adaptive QoS transport service". With this service, an application (or the proxy from a service provider) could setup an optimized CIR for different stage of application for the economical and efficient purpose. For example, in the transport of compressed video, the I-frame has big size and cannot be lost, but P-frame and B-frame both have smaller size and can tolerate some loss. There are much more P-frame and B-frame than I-frame in videos with smooth changes and variations in images [[I-D.han-iccr-g-arvr-transport-problem](#)].

Based on this characteristics, application can request a relatively small CIR for the time of P-frame and B-frame, and request a big CIR for the time of I-frame.

2. Repairing of the QoS channel: This is the case the QoS channel was broken and need to be repaired, see [section 5.3](#).

[6.2](#). Working with UDP and other Protocols

There are other transport layer protocols, such as UDP, QUIC and SCTP, and for these protocols similar strategy as TCP can be applied. The to establish a closed-loop for the transport control.

For protocols with natively bi-directional control mechanism such as SCTP, only some QoS control functionalities for the protocol need to be added. The mechanism for TCP can be borrowed for such job. There will be the QoS setup for one directional data stream, and QoS setup state report for another directional data stream. The protocol may also have functionalities in the stack to handle the adjustment of

the behaviour for different QoS setup and setup states.

For protocols that natively lack the feed-back control mechanism to form a closed-loop such as UDP, this mechanism needs to be added into the streams. There are two options to realize this:

1. Modify the protocol itself to have some state machine to establish the closed-loop for the protocol. This can be done in the kernel of the OS by modifying the protocol stack.
2. Modify the user data stream to introduce the closed-loop scheme, this becomes as application work. It is up to application to add or modify codes for the state machine of the closed-loop control.

7. Additional Considerations

This document only covers the details of setting up a path with QoS using IPv6, and TCP is used as an example of transport layer protocol to achieve flow level service. Only basic scenarios are covered, and there are lots of open issues to be researched. The following is a non-comprehensive list, and they can be addressed in separate drafts.

7.1. User and Application driven

The QoS transport service is initiated and controlled by end user's application. Following tasks are done in host:

1. The detailed QoS parameters in signaling message are set by end user application. New socket option must be added, the option is

a place holder for QoS parameters (Setup, Bandwidth, etc.), Setup State Report and Forwarding State Report messages.

2. The Setup State Report and Forwarding State Report message received at host are processed by transport service in kernel. The Setup State Report message processed at host can result in the notification to the application whether the setup is successful. If the setup is successful, the application can start to use the socket having the QoS support; If the setup is failed, the application may have three choices:

* Lower the QoS requirement and re-setup a new QoS channel with

new in-band signaling message.

- * Use the TCP session as traditional transport without any QoS support.
- * Lookup the service provider for help to locate the problem in network.

7.2. Traffic Management in Host

In order to better accommodate this new IP in-band service, the OS on a host may be changed in traffic management related areas. There are two parts for traffic management to be changed: one is to manage traffic going out a host's shared links, and the other is congestion control for TCP flows.

1. For current traffic management in a host, all TCP/UDP sessions will share the bandwidth for all egress links. For the purpose to work with the differentiated service provided by under layer network in bandwidth and latency, the kernel may allocate expected resource to applications that are using the QoS transport service. For example, kernel can queue different packets from different applications or users to different queue and schedule them in different priority. Only after this change, some application can use more bandwidth and get less queuing delay for a link than others.
2. The congestion control in a host manages the behavior of TCP flow(s). This includes important features like slow start, AIMD, fast retransmit, selective ACK, etc. To accommodate the benefit of the QoS guaranteed transport service, the congestion control can be much simpler [[I-D.han-tsvwg-cc](#)]. The new congestion control is related to the implementation of QoS guarantee. Following is a simple congestion control algorithm assuming that the CIR is guaranteed and PIR is shared between flows:

- * There is no slow start, the TCP can start sending traffic at the rate of CIR.
- * The AIMD is kept, but the range of the sawtooth pattern should be maintained between CIR and PIR.

- * Other congestion control features can be kept.

[7.3.](#) Heterogeneous Network

When an IP network is connected with a non-IP network, such as MPLS or Ethernet network, the in-band signaling should also work in that network to achieve an end-to-end connection. The behavior, protocol and rules in the interworking with non-IP network is out of the scope of this draft, and further research needs to be done to solve the problem.

[7.4.](#) Proxy Control

It is expected that in a real service provider network, the in-band signaling will be checked, filtered and managed at proxy routers. It serves the following purposes:

1. A proxy can check if the in-band signaling from an end user meets the SLA compliance. This adds extra security and DOS attack prevention.
2. A proxy can collect the statistics for user's TCP flows and check the in-band signaling for accounting and charging.
3. A proxy can insert and process appropriate in-band signaling for TCP flows if the host does not support this new feature. This can provide backward compatibility, also enable the host to use the new feature.

[8.](#) IANA Considerations

This document defines a new option type for the Hop-by-Hop Options header and the Destination Options header. According to [[RFC8200](#)], the detailed value are:

Hex Value	Binary Value			Description	Reference
	act	chg	rest		
0x0	00	0	10000	In-band Signaling	Section 4 in this doc

Figure 3: The New Option Type

1. The highest-order 2 bits: 00, indicating if the processing IPv6 node does not recognize the Option type, skip over this option and continue processing the header.
2. The third-highest-order bit: 0, indicating the Option Data does not change en route.
3. The low-order 5 bits: 10000, assigned by IANA.

This document also defines a 4-bit subtype field, for which IANA will create and will maintain a new sub-registry entitled "In-band signaling Subtypes" under the "Internet Protocol Version 6 (IPv6) Parameters" [[IPv6 Parameters](#)] registry. Initial values for the subtype registry are given below

Internet-Draft

ResRev for IP QoS

October 2019

Type	Mnemonic	Description	Reference
0	SETUP	Setup object	Appendix B
1	BANDWIDTH	Bandwidth object	Appendix B
2	BURST	Burst object	Appendix B
3	LATENCY	Latency object	Appendix B
4	AUTH	Authentication object	Appendix B
5	OAM	OAM object	Appendix B
6	FWD STATE	Forward state	Appendix B
7	SETUP REPORT	Setup state report	Appendix B
8	FWD REPORT	Forwarding state report	Appendix B

Figure 4: The In-band Signaling Sub Type

9. Security Considerations

It is important to guarantee that the resource reservation is used by authenticated users, and false signaling should not be accepted or processed. The following aspects may be considered:

Authentication of user

If an user is interested in using this new service, the user should sign up to a service provider. Service provider should do the proper authentication check for a new user, and establish account for the user.

After the sign up, a user should provide a security key to the service provider through a secured channel (https, registered mail, etc.), or the key could be generated and given to user by the service provider. Service provider should distribute the security key of the user to different network device. More

specifically, the security key should be distributed securely to all HbH-EH-aware nodes for an open network, or the proxy for a closed network.

Proxy

Han, et al.

Expires April 17, 2020

[Page 24]

Internet-Draft

ResRev for IP QoS

October 2019

Proxy or gateway is the 1st network device connecting to customer's devices (Host, phone, etc.) that can generate the signaling for resource reservation. The functionality of the Proxy is to check if the signaling is allowed to go through SP's network. This can be done by checking the signaling integrity and other info associated with the user, such as the source/destination IP address, the account balance, the user's privilege, etc.

Authentication of signaling message

The signaling for resource reservation should be checked at each HbH-EH-aware nodes or a proxy node.

Service ID is originally used for performance improvement of forwarding with QoS, and it can also provide additional security protection of forwarding resource in data plane. Service ID in each HbH-EH-aware node is to represent an IP flow with programmed QoS service, and it is a local significant number generated by a router to identify a flow that was offered QoS service. So, the router can periodically change the number for the same flow to protect any middle box sniffing for DOS attacking. It can be done by host periodical send out in-band signaling with the same QoS parameters and obtain the new Service ID and Service ID List for the use of next data forwarding.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC2581] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", [RFC 2581](#), DOI 10.17487/RFC2581, April 1999, <<https://www.rfc-editor.org/info/rfc2581>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, [RFC 8200](#), DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

Han, et al.

Expires April 17, 2020

[Page 25]

Internet-Draft

ResRev for IP QoS

October 2019

[10.2](#). Informative References

- [eNodeB] wikipedia, "eNodeB", 2018, <<https://en.wikipedia.org/wiki/ENodeB>>.
- [EPC] 3GPP, "The Evolved Packet Core", 2018, <<http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core>>.
- [HU5G] Huawei, "5G Vision: 100 Billion connections, 1 ms Latency, and 10 Gbps Throughput", 2015, <<http://www.huawei.com/minisite/5g/en/defining-5g.html>>.
- [I-D.falk-xcp-spec]
Falk, A., "Specification for the Explicit Control Protocol (XCP)", [draft-falk-xcp-spec-03](#) (work in progress), July 2007.
- [I-D.han-iccr-g-arvr-transport-problem]
Han, L. and K. Smith, "Problem Statement: Transport Support for Augmented and Virtual Reality Applications", [draft-han-iccr-g-arvr-transport-problem-01](#) (work in progress), March 2017.
- [I-D.han-tsvwg-cc]
Han, L., Qu, Y., and T. Nadeau, "A New Congestion Control in Bandwidth Guaranteed Network", [draft-han-tsvwg-cc-00](#) (work in progress), March 2018.

[I-D.harper-inband-signalling-requirements]

Harper, J., "Requirements for In-Band QoS Signalling", [draft-harper-inband-signalling-requirements-00](#) (work in progress), January 2007.

[I-D.ietf-aqm-codel]

Nichols, K., Jacobson, V., McGregor, A., and J. Iyengar, "Controlled Delay Active Queue Management", [draft-ietf-aqm-codel-06](#) (work in progress), December 2016.

[I-D.ietf-aqm-fq-codel]

Hoeiland-Joergensen, T., McKenney, P., dave.taht@gmail.com, d., Gettys, J., and E. Dumazet, "The FlowQueue-CoDel Packet Scheduler and Active Queue Management Algorithm", [draft-ietf-aqm-fq-codel-06](#) (work in progress), March 2016.

Han, et al.

Expires April 17, 2020

[Page 26]

Internet-Draft

ResRev for IP QoS

October 2019

[I-D.ietf-aqm-pie]

Pan, R., Natarajan, P., Baker, F., and G. White, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", [draft-ietf-aqm-pie-10](#) (work in progress), September 2016.

[I-D.ietf-detnet-use-cases]

Grossman, E., "Deterministic Networking Use Cases", [draft-ietf-detnet-use-cases-20](#) (work in progress), December 2018.

[I-D.ietf-spring-segment-routing]

Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [draft-ietf-spring-segment-routing-15](#) (work in progress), January 2018.

[I-D.ietf-tcpm-dctcp]

Bensley, S., Eggert, L., Thaler, D., Balasubramanian, P., and G. Judd, "Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters", [draft-ietf-tcpm-dctcp-03](#) (work

in progress), November 2016.

[I-D.roberts-inband-qos-ipv6]

Roberts, L. and J. Harford, "In-Band QoS Signaling for IPv6", [draft-roberts-inband-qos-ipv6-00](#) (work in progress), July 2005.

[I-D.sridharan-tcpm-ctcp]

Sridharan, M., Tan, K., Bansal, D., and D. Thaler, "Compound TCP: A New TCP Congestion Control for High-Speed and Long Distance Networks", [draft-sridharan-tcpm-ctcp-02](#) (work in progress), November 2008.

[IPv6_Parameters]

IANA, "Internet Protocol Version 6 (IPv6) Parameters", 2015, <<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-2>>.

[MEC]

ETSI, "Multi-access Edge Computing", 2018, <<https://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing>>.

[NGP]

ETSI, "Next Generation Protocols (NGP); Recommendation for New Transport Technologies", 2018, <https://www.etsi.org/deliver/etsi_gr/NGP/001_099/010/01.01.01_60/gr_NGP010v010101p.pdf>.

Han, et al.

Expires April 17, 2020

[Page 27]

Internet-Draft

ResRev for IP QoS

October 2019

[QU2016]

Qualcomm, "Leading the world to 5G", 2016, <<https://www.qualcomm.com/media/documents/files/qualcomm-5g-vision-presentation.pdf>>.

[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](#), DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.

[RFC2475]

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", [RFC 2475](#), DOI 10.17487/RFC2475, December 1998, <<https://www.rfc-editor.org/info/rfc2475>>.

- [RFC3175] Baker, F., Iturralde, C., Le Faucheur, F., and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", [RFC 3175](https://www.rfc-editor.org/info/rfc3175), DOI 10.17487/RFC3175, September 2001, <<https://www.rfc-editor.org/info/rfc3175>>.
- [Tactile] JDavid Szabo, et al. Proceedings of European Wireless 2015; 21th European Wireless Conference, "Towards the Tactile Internet: Decreasing Communication Latency with Network Coding and Software Defined Networking", 2015, <<http://fastpass.mit.edu/Fastpass-SIGCOMM14-Perry.pdf>>.
- [TCP-vegas] Peterson, L., "TCP Vegas: New Techniques for Congestion Detection and Avoidance - CiteSeer page on the 1994 SIGCOMM paper", 1994.
- [TCP_Targets] Andreas Benthin, Stefan Mischke, University of Paderborn, "Bandwidth Allocation of TCP", 2004.
- [TIA] TIA 1039 Revision A, "QoS Signaling for IP QoS Support and Sender Authentication", 2015, <https://global.ihs.com/doc/detail.cfm?&csf=TIA&item_s_key=00480715&item_key_date=880431>.

[Appendix A.](#) Acknowledgements

The authors are very grateful to Fred Baker for his valuable contributions to this document.

We appreciate the following people who made lots of contributions to this draft: Guoping Li, Boyan Tu, and Xuefei Tan, and thank Huawei Nanjing research team led by Feng Li to provide the Product on

Han, et al. Expires April 17, 2020 [Page 28]

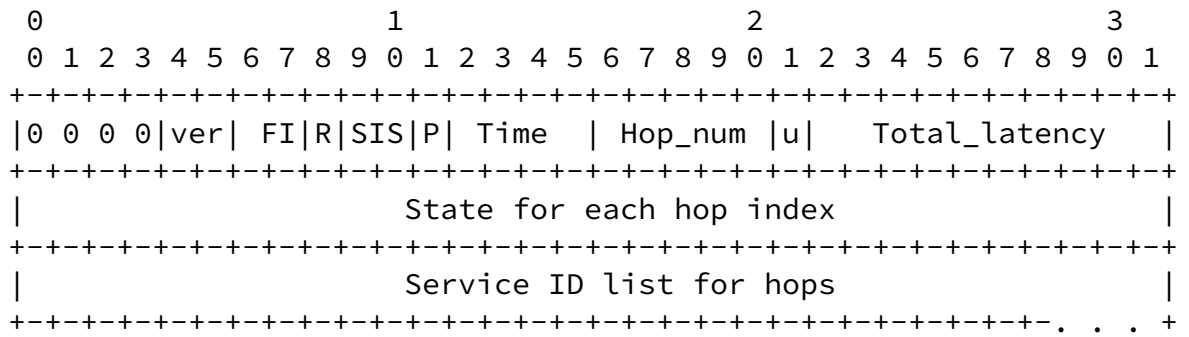
Internet-Draft ResRev for IP QoS October 2019

Concept (POC) development and test, the team members include Fengxin Sun, Xingwang Zhou, and Weiguang Wang. We also like to thank other people involved in the discussion of solution: Tao Ma from Future Network Strategy dept.

[Appendix B.](#) Message Objects

This section defines detailed objects used in different messages.

[B.1.](#) Setup State Object



Type = 0, Setup state;

Version: The version of the protocol for the QoS

FI: Flow identification method,

0: 5 tuples; 1: src,dst,port; 2: src,dst; 3: DSCP

R: If the destination host report the received Setup state to
the src address by Destination EH. 0: dont report; 1: report

SIS: Service ID size; 0: 0bits, 1: 16bits, 2: 20bits, 3: 32bits

P: 0: program HW for the QoS from src to dst;

1: De-program HW for the QoS from src to dst

Time: The life time of QoS forwarding state in second.

Hop_num: The total hop number on the path set by host. It must be decremented at each hop after the processing.

u: the unit of latency, 0: ms; 1: us

Total_latency : Latency accumulated from each hop, each hop will add the latency in the device to this value.

Figure 5: The Setup State Object

Setup state for each hop index: each bit is the setup state on each hop on the path, 0: failed; 1: success. The 1st hop is at the most significant bit.

Internet-Draft

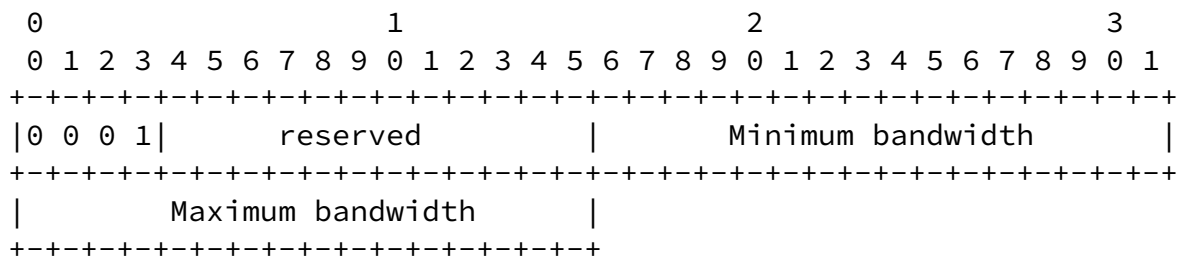
ResRev for IP QoS

October 2019

Service ID list for hops: it is for all hops on the path, each service ID bit size is defined in SIS. The 1st service ID is at the top of the stack. Each hop add its service ID at the correct position indexed by the current hop number for the router.

The Setup object is embedded into the hop-by-hop EH to setup the QoS in the device on the IP forwarding path. To keep the whole setup message size unchanged at each hop, the total hop number must be known at the source host. The total hop number can be detected by OAM. The service ID list is empty before the 1st hop receives the in-band signaling. Each hop then fill up the associated service ID into the correct place determined by the index of the hop.

[B.2.](#) Bandwidth Object



Type = 1,

Minimum bandwidth : The minimum bandwidth required, or CIR, unit Mbps

Maximum bandwidth : The maximum bandwidth required, or PIR, unit Mbps

Figure 6: The Bandwidth Object

[B.3.](#) Burst Msg

Internet-Draft

ResRev for IP QoS

October 2019

```

      0                               1                               2                               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0 0 1 0|          Burst size          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type = 2,

Burst size : The burst size, unit M bytes

Figure 7: The burst message

[B.4.](#) Latency Object

```

      0                               1                               2                               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0 0 1 1|u|          Latency          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type = 3,

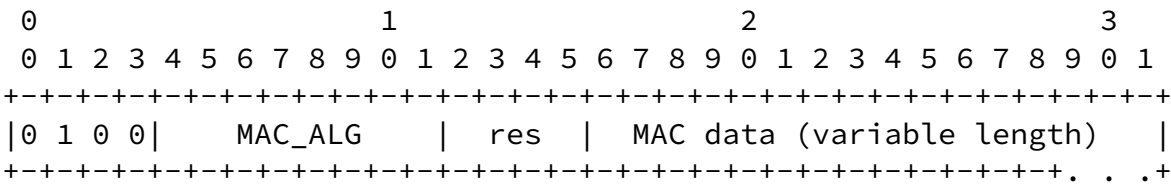
u: the unit of the latency

0: ms; 1: us

Latency: Expected maximum latency for each hop

Figure 8: The Latency Object

B.5. Authentication Object



Type = 4,

MAC_ALG: Message Authentication Algorithm

0: MD5; 1:SHA-0; 2: SHA-1; 3: SHA-256; 4: SHA-512

MAC data: Message Authentication Data;

Res: Reserved bits

Size of signaling data (opt_len): Size of MAC data + 2

MD5: 18; SHA-0: 22; SHA-1: 22; SHA-256: 34; SHA-512: 66

Figure 9: The Authentication Object

B.6. OAM Object

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0 1 0 1| OAM_t |   OAM_len   |   OAM data (variable length) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type = 5,

OAM_t : OAM type

OAM_len : 8-bit unsigned integer. Length of the OAM data, in octets;

OAM data: OAM data, details of OAM data are TBD.

Figure 10: The OAM Object

[B.7.](#) Forwarding State Object

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0 1 1 0|ver| FI|R|SIS|P| Time  | Hop_num |u|   Total_latency   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Forwarding state for each hop index                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Service ID list for hops                                         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type = 6, Forwarding state;

All parameter definitions and process in the 1st row are same in the setup message.

Forward state for each hop index : each bit is the fwd state on

hop on the path, 0: failed; 1: success; The 1st hop is at the most significant bit.

Figure 11: The Forwarding State Object

B.8. Setup State Report Object

[illegible]

H: Hop number bit. When a host receives a setup message and form

Type = 8, Forwarding state report;

H: Hop number bit. When a host receives a Forward State message and form a Forward State Report message, it must check if the Hop_num in Forward State message is zero. If it is zero, the H bit is set to one, and if it is not zero, the H bit is clear.

This will notify the source of Forward State message that if the original Hop_num was set correct.

Following are directly copied from the Forward State message:

u, Total_latency;

Forwarding State for each hop index

Figure 13: The Fwd State Report Object

Authors' Addresses

Lin Han
Futurewei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +10 408 330 4613
Email: lin.han@futurewei.com

Futurewei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: yingzhen.qu@futurewei.com

Lijun Dong
Futurewei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: lijun.dong@futurewei.com

Richard Li
Futurewei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: renwei.li@futurewei.com

Thomas Nadeau
Lucid Vision
Hampton NH 03842
USA

Email: tnadeau@lucidvision.com

Kevin Smith
Vodafone
UK

Email: Kevin.Smith@vodafone.com

Jeff Tantsura
Apstra

Email: jefftant.ietf@gmail.com