Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 14, 2018

L. Han, Ed.
G. Li
B. Tu
X. Tan
F. Li
R. Li
Huawei Technologies
J. Tantsura

K. Smith
Vodafone

October 11, 2017

IPv6 in-band signaling for the support of transport with QoS
draft-han-6man-in-band-signaling-for-transport-qos-00

Abstract

This document proposes a method to support the IP transport service that could guarantee a certain level of service quality in bandwidth and latency. The new transport service is fine-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 14, 2018.

Table of Contents

1. Introduction

Recently, more and more new applications for Internet are emerging. These applications have a common part that is 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, AR or VR applications may need a couple of hundred Mbps bandwidth (throughput) and a low single digit ms latency. Moreover, the difference of mean bit rate and peak bit rate is huge due to the compression algorithm [I-D.han-iccrg-arvr-transport-problem].

Some future applications expect that network can provide a bounded latency service, such as tactile network [Tactile].

With the technology development in 5G and beyond, the wireless access network is also rising the demand for the Ultra-Reliable and Low-Latency Communications (URLLC), this also leads to the question if IP transport can provide such service in Evolved Packet Core (EPC) network. IP is becoming more and more important in EPC when the Multi-access Edge Computing (MEC) for 5G will require the cloud and data service moving closer to eNodeB.

Following sections will brief the current transport and QoS technologies, and analyze the limitations to support above new applications.

A new approach that could provide QoS for transport service will be proposed. The scope and criteria for the new technology will also be summarized.

1.1. IP and Transport Technologies

The traditional IP network can only provide the best-effort service. The transport layer (TCP/UDP) on top of IP are based on this fundamental architecture. The best-effort-only service has influenced the transport evolution for quite long time, and results in some widely accepted assumptions and solutions, such as:

1. The IP layer can only provide the basic P2P (point to point) or P2MP (point to multi-point) end-to-end connectivity in 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.

1.2. TCP Solution Analysis

As a most popular and widely used transport technology, TCP traffic is dominating in Internet from the born of Internet. It is important to analyze the TCP. This section will brief the TCP, its variation, and some key factors.

1.2.1. TCP Overview and Evolution

The major functionalities of TCP are flow control and congestion control.

The flow control is based on the sliding window algorithm. In each TCP segment, the receiver specifies in the receive window field the amount of additionally received data (in bytes) that it is willing to

buffer for the connection. The sending host can send only up to that amount of data before it must wait for an acknowledgment and window update from the receiving host.

The congestion control is algorithms to prevent the hosts and network device fall into congestion state while trying to achieve the maximum throughput. There are many algorithm variations developed so far.

All congestion control will use some congestion detection scheme to detect the congestion and adjust the rate of source to avoid the congestion.

No matter what congestion control algorithm is used, traditionally, all TCP solutions are pursuing three targets, high efficiency in bandwidth utilization, high fairness in bandwidth allocation, and fast convergence to the equilibrium state. [TCP_Targets]

Recently, with the growth of new TCP applications in data center, more and more solutions were proposed to solve bufferbloat, incast problems typically happened in data center. These solutions include DCTCP, PIE, CoDel, FQ-CoDel, etc. In addition to the three traditional targets mentioned above, these solutions have another target which is to minimize the latency.

1.2.2. TCP Solution Variants

There are many TCP variants and optimization solutions since TCP was introduced 40 years ago. We have collected major TCP variants including typical traditional solution and some new solutions proposed recently.

The traditional solutions:

These solutions are implemented on host only. They use different congestion detection and inference mechanism, either based on packet loss, RTT or both, to dynamically adjust the TCP window to do the congestion control, such as: TCP-reno [RFC2581], TCP-vegas [TCP-vegas], TCP-cubic [TCP-cubic], TCP-compound [I-D.sridharan-tcpm-ctcp], TIMELY [TIMELY], etc

The explicit rate solutions:

These solutions do not use the traditional black box mechanism executed at host to infer the TCP congestion status, instead, they rely on the rate calculation on routers to let host adjust accordingly. Both network devices and hosts must be changed. Typical solutions are: XCP [I-D.falk-xcp-spec], RCP [RCP]. Note, we put XCP and RCP as TCP here is referring to the scenario when XCP and RCP are used with TCP

The AQM solutions:
These solutions use AQM (Active Queue Management) techniques on routers to control the buffer size, thus control the congestion and minimize the latency indirectly. Both network devices and hosts must be changed. They include: DCTCP [I-D.ietf-tcpm-dctcp], PIE [I-D.ietf-aqm-pie], CoDel [I-D.ietf-aqm-codel], FQ-CoDel [I-D.ietf-aqm-fq-codel], etc.

The new concept solutions:
Unlike above categories, these solutions use completely new concepts and methods to either accurately calculate, or figure out the optimized rate and latency of TCP, such as: PERC [PERC], BBR [BBR], PCC [PCC], Fastpass [Fastpass], etc

1.2.3. Throughput Constraint

For the traditional TCP optimization solutions, the efficiency target is to obtain the high bandwidth utilization as much as possible to approach the link capacity. The link utilization is defined as the total throughput of all TCP flows on a network device to the network bandwidth for links.

For individual TCP flow, its actual throughput is not guaranteed at all. It depends on many factors, such as TCP algorithm used, the number of TCP flows sharing the same link, host CPU power, network device congestion status, delay in transmission, etc.

For traditional TCP, the real throughput for a flow is limited by three factors: The 1st one is the available maximum throughput at the physical layer, accounting for maximum theoretical bandwidth, network load, buffering configuration, maximum segment size, signal strength, etc; The another is related to congestion control algorithm; The 3rd is related to the TCP fairness principle. Below we will analyze the last two factors.

1.2.3.1. By Algorithm

No matter what algorithm is used, The TCP throughput is always related to some flow and network characteristics, such as the RTT (Round Trip Time) and PLR (packet loss ratio). For example, TCP-reno throughput is shown in the formula (3) in [Reno_throughput]; And TCP-cubic throughput is expressed in formula (21) in [Cubic_throughput].

This limit will prevent the link capacity to be utilized by all TCP flows. Each TCP flow may only get a few portion of the link bandwidth as the real throughput for application. Even there is one TCP flow in a link, the throughput for the TCP could be way below the link capacity for a network which RTT and PLR are high.

1.2.3.2. By Fairness Principle

TCP fairness is a de facto principle for all TCP solutions. By this rule, each router will process all TCP flows equally and fairly to allocate the required resource to all TCP flows. Different Fair Queuing algorithms were used, such as Packet based Round Robin, Core-Stateless Fair Queuing(CSFQ), WFQ, etc. The targets of all algorithms are to reach the so called max-min fairness [Fairness] of TCP in terms of bandwidth.

TCP fairness played an important and critical role in saving internet from collapse caused by congestions since TCP was introduced.

The analysis [RCP] on page 35 has given the formula of the fair share rate at bottleneck routers, the rate or throughput is capped for applications which required bandwidth are not satisfied under the rule of fairness.

1.2.4. Latency Constraint

TCP fairness will not process some TCP flows differently with others, or there is no TCP micro-flow handling.

As described above, for the traditional solutions and explicit rate solution, the latency is not considered as a target, thus no latency guarantee at all.

For AQM solutions and some new concept solutions which try to control the buffer bloat or flow latency, it can only provide the statistic bounded latency for all TCP flows. The latency is related to the queue size and other factors. And the real latency for specific flow(s) is not deterministic. It could be very small or pretty large due to the long tail effect if the flow is blocked by other slower TCP flows.

1.2.5. Summary of TCP Solution

The bandwidth and latency can hardly be satisfied simultaneously without micro flow handling and management. While trying to get higher bandwidth, it may lead to more queued packet in router and result in longer latency. While approaching shorter latency, it may cause the queue under run, and lead to the lower bandwidth.

As a summary, to support some special TCP applications that are very sensitive to bandwidth and/or latency, we need to handle those TCP flows differently with others, and the TCP fairness must be relaxed for these scenarios.

It must be noted that the fairness based transport service could satisfy most of the applications, and it is the most efficient and economical way for hardware implementation and the network bandwidth efficiency.

When providing some TCP flows with differentiated service, the traditional transport service must be able to coexist with the new service. The resource partitioning between different service is a operation and management job for service provider.

1.3. Other Solution Analysis

DiffServ
: DiffServ [DiffServ] or Differentiated services is a network architecture that specifies a simple, scalable and coarse-grained mechanism for classifying and managing network traffic and providing QoS on modern IP networks. DiffServ is designed to support the QoS of aggregated traffic and normally is deployed in Service Provider networks. End user application cannot directly use DiffServ.

IntServ
: IntServ [IntServ] or integrated services specifies more fine-grained QoS, which is often contrasted with DiffServ's coarse-grained control system. IntServ definitely can support the applications requiring special QoS guarantee if it is deployed in a network, supported by Host OS and integrated with application. However, IntServ works on a small-scale only. When you scale up the network, it is difficult to keep track of all of the reservations and session states. Thus, IntServ is not scalable. Another problem of IntServ is it is not application driven, tedious provisioning cross different network must be done earlier. The provisioning is slow and hard to maintain.

MPLS-TE
: MPLS-TE can provide aggregated QoS or fine-grained QoS service for different class of traffic. Similar to DiffServ, MPLS-TE is majorly used for service providers network. It requires extra protocol sets like LDP, MPLS-TE, etc to operate. It is not practical to extend MPLS-TE to end user's desktop.

1.4. New approach

1.4.1. IP Transport with quality of service

Semiconductor chip technology has advanced a lot for last decades, the widely used network process can not only forward the packet in line speed, but also support fast packet processing for other

features, such as Qos for DiffServ/MPLS, Access Control List (ACL), fire wall, Deep Packet Inspection (DIP), etc. To treat some TCP/IP flows differently with others and give them specified resource are feasible now by using network processor.

Network processor is also able to do the general process to handle the simple control message for traffic management, such as signaling for hardware programming, congestion state report, OAM, etc.

This document proposes a mechanism to provide the capability of IP network to support the transport layer with quality of service. The solution is based on the QoS implemented in network processor. the proposal of the document is composed of two parts:

1. Control plane, it explains a transport control sub-layer for IP, the details of control mechanism.

2. Data plane, the realization of QoS in data forwarding, QoS and error handling.

1.4.2. Design targets

The new transport service is expected to satisfy following criteria:

1. End user or application can directly use and control the new service

2. The new service can coexist with the current transport service and is backward compatible.

3. The service provider can manage the new service.

4. Performance and scalability targets of new service are practical for vendors to achieve.

5. The new service is transport agnostic. Both TCP, UDP and other transport protocols on top of IP can use it

1.4.3. Scope and assumption

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 transport with QoS is aimed to be supplementary to the regular transport service. At the current situation, It is targeted for the applications that are bandwidth and/or latency

sensitive. It is not intended to replace the TCP 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 to extend the mechanism for inter-domain scenarios. Thus, the security and other inter-domain requirements are not critical. The basic security is good enough, 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 the application that really needs it, the number of supported applications/users are under controlled and cannot be unlimited. So, the scalability requirement for the new service is limited.

4. The new service must coexist with the regular transport service in the same hardware, and backward compatible. Also, a transport flow can switch without the service interruption between the regular transport support and new service.

2. Terminology

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

2.1. Definitions

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 the sending of 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, a IP flow is identified by the source, destination IP address, the protocol number, the source and destination port number.

IP path
: A IP path is the route that IP flow will traverse. It could be the shortest path determined by routing protocols (IGP or BPG), or the explicit path decided by another management entity, such as a central controller, or Path Computation Element (PCE) Communication Protocol (PCEP), etc

QoS channel
: A forwarding channel that the QoS is guaranteed, it provides an additional QoS service to the normal IP forwarding. A QoS channel can be used for one or multiple IP flows depends on the granularity of in-band signaling.

Cir
: Committed Information Rate, this is the guaranteed bandwidth

Pir
: Peak Information Rate. this is the up limit bandwidth. Whether a flow can reach the PIR depends on the implementation. To use resource more efficiently, the system normally does not guarantee the PIR, but allow the sharing of resource between flows.

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. Control plane

3.1. Sub-layer in IP for transport control

In order to provide some new features for the upper 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 the 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.

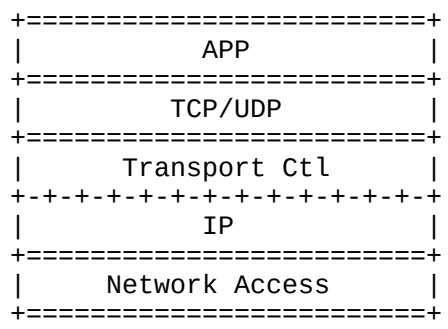


Figure 1: The new stack with a sub-layer in Layer 3

The new sub-layer is always bound with IP layer and can provide a 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.

IPv6 can realize the sub-layer easily by the IPv6 extension header [RFC8200].

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.

The document will focus on the solution for IPv6 by using different IPv6 extension header.

The control plane of the propose comprises of IP in-band signaling, and the detailed control mechanisms.

3.2. IP In-band signaling

There is no definition for IP in-band signaling. From the point of view of similarity to traditional telecommunication technology, the In-band signaling for IP is that the IP control messages are sharing some common header information as the data packet.

In this document, we introduce three types of "in-band signaling" for different signaling granularity:

Flow level In-band Signaling
: The control message and data packet share the same flow identification. 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. For the flow level in-band signaling, the signaling is for the individual IP flow, and there is no aggregation at all.

Address level In-band Signaling
: The control message and data packet 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
: The control message and data packet 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 is the situation that the signaling is for the aggregated TCP or UDP flows that started and terminated at the same IP addresses.

Using In-band signaling, the control message can be embedded into any data packet, 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.

Note, the requirement of IP in-band signaling was proposed before by John Harper [I-D.harper-inband-signalling-requirements]. And the in-band QoS signaling for IPv6 was simply discussed in [I-D.roberts-inband-qos-ipv6]. Unfortunately, both works did not continue.

This document not only gives detailed solution for in-band signaling, but also try to address issues raised for the previous proposal, such as security, scalability and performance. Finally, experiments with proprietary hardware and chips are given in a presentation.

3.3. Control mechanism

The in-band signaling must be cooperated with a control method to achieve the QoS control. There are two categories of control, one is the closed-loop control and another is the open-loop control.

1. Closed-loop control is that the in-band signaling is sent in one direction and the feedback will return in the reverse direction. For example, the closed-loop control can be achieved by inserting the signaling information into a data packet sent in one direction, and the feedback information is carried in the data packet in reverse direction. The transport service with bi-direction data flow can use this mechanism, such as TCP and point-to-point UDP. In closed-loop control, a signaling message in one direction is processed at each router on the path. When the signaling message reaches the destination, the signaling message is processed by the protocol stack in the host, and the report information is generated. The report information is then

embedded into the flow data packet in the reverse direction and return to the host of the signaling source.

2. Open-loop control is that the in-band signaling is sent periodically in one direction without any feedback. The transport service with uni-direction data flow can use this mechanism, such as multicast by UDP. The transport service with bi-directional data flow can also use this mechanism when the simplicity of the control is wanted, i.e. no control feedback needed.

For both closed-loop and open-loop control, the signaling message for one direction is for the QoS programming for the direction. For example, the TCP-SYN or TCP data packet from client to server can carry the in-band signaling message to program the QoS for the direction of client to service. TCP-SYNACK or TCP data packet from server to client can carry the in-band signaling message to program the QoS for the server to client direction

Due to the nature that symmetric IP path between any source and destination cannot be guaranteed, in closed-loop control, the feedback information may take the different path as the in-band signaling path. The in-band signaling must not depend on the feedback information to accomplish the signaling work, such as the programming of hardware. This is one of the difference between in-band signaling and RSVP protocol.

For this document, we will only discuss the detailed mechanism for closed-loop control for TCP.

3.4. IPv6 Approach

The IPv6 In-band signaling could be realized by using the IPv6 extension header.

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]. We call this nodes as HbH-EH-aware nodes in document below. It 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.

3.4.1. Basic Control Scenarios for TCP

The finest grained QoS for TCP is flow level, this document will only focus on the solution of the flow level in-band signaling and its data plane. Other two types, address level and transport level QoS for TCP are briefly discussed in section 5.3.

The feature of TCP with flow level QoS comprises following control scenarios:

1. Setup: The setup is combined with the TCP 3-hand shaking, or any two directional TCP packets. When used with TCP 3-hand shaking, the 1st signaling embedded into HbH-EH is sent with TCP-SYN. It will be processed at HbH-EH-aware nodes on the path from source to destination. The signaling message includes the QoS requirements, such as max/min bandwidth, burst size, the latency, and the setup state. The setup state message is updated at HbH-EH-aware nodes to include the QoS programming and provisioning result and the necessary hardware reference information for IP forwarding with QoS. The 2nd signaling message is the TCP-SYNACK from server side, it includes the setup report message encoded as the Dst-EH. The setup report message is from the 1st TCP-SYN which represents the setup results on all HbH-EH-aware nodes on the path. The setup can even be started after TCP is established whenever the QoS service is required.

2. Dynamic control: this scenario is for the situation that previous QoS programming must be refreshed, modified or re-programmed. Normally, the signaling message can be embedded into HbH-EH for any TCP data packet or TCP-ACK packet. There are couple cases that the dynamic control is needed.

   HW state refreshing
      The HW state for QoS programming is data driven (see Section 4.1 for details). Its state will be refreshed if there is a data packet received. If there is no data received for a pre-configured time, the HW programming will be erased and the resource will be released.

   HW programming modification
      The HW QoS parameters can be modified if a new in-band signaling message is received and the embedded parameters are different with the old one that was used to program the HW. Section 3.4.2 will explain more about this scenario.

HW programming repairing
The IP path may be changed due to rerouting, link or node failures. This may result in the HW QoS programming failure. To repair any QoS programming failure, the new in-band signaling message can be embedded into any data packet and sent to the destination. All hops on the new path will be reprogramed with the QoS parameters. Section 4.4 has more detailed discussion.

3. Congestion Control: For TCP protocol, if IP layer can provide a certain level of quality service guarantee, the congestion control algorithm will be impacted a lot. As for what is the new congestion control, it depends on the quality service implementation in hardware and the behavior of the application. This is simply discussed in section 5.2.

3.4.2. Details of In-band Signaling for TCP

This document introduces following type of message for in-band signaling and associated data forwarding, the detailed format of messages is expressed in Section 6,

- Setup: This is for the setup of QoS channel through the IP path.
- Bandwidth: This is the required bandwidth for the QoS channel. It has minimum (CIR) and maximum bandwidth (PIR).
- 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.
- Burst: This is the required burst for the QoS channel, it is the maximum burst size.
- Authentication: This is the security message for a in-band signaling.
- OAM: This is the Operation and Management message for the QoS channel.
- Setup State Report: This is the state report of a setup message.
- Forwarding State: This is the forwarding state message used for data packet.
- 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 Mapping Index 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 5.1

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 impcted 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-iccrg-arvr-transport-problem]. Based on this characteristics, application can request a relatively small CIR for the time of P-frame and P-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 4.4.

3.5. Key Messages and Parameters in Control Protocol

The detailed message format is described in the section 6, the detailed explanation of key messages and parameters are below:

3.5.1. Setup and Setup State Report messages

Setup is the message used for following purpose:

- Setup the QoS channel for a TCP when the TCP session is establishing.

- Dynamic Control of the QoS channel for a established TCP session. See section 3.4.1

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 appropriate TCP packet and will be processed at each HbH-EH-aware node. For the simplicity, performance and scalability purpose, we can configure some hop to do the processing and some hops do not. For different QoS requirement and scenarios, different criteria can be used for the configuration of the hop to be HbH-EH-aware node, below are some factor to consider:

- 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. The throttle router is the device that a interested TCP session cannot get the enough bandwidth to support its application. The regular throttle routers include the BRAS (broadband remote access server) in broadband access network, the PGW (PDN Gateway) in LTE network, the TOR (Top of Rack) in data center. In more general case, any routers which aggregate traffic may become as a throttle router. Moreover, the direction of congestion must be considered. Normally, the congestion happens on the direction that more than one flows from multiple ingress links are aggregated and sent to one egress link. For other devices that the interested TCP session can get the enough bandwidth do not need to process the hop-by-hop EH.

- 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-cast 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.

3.5.2. OAM

OAM is a special in-band signaling message used for detection and diagnosis. It can be used before and after a QoS channel is established. Before a QoS channel is established, OAM message can be added as a HbH-EH to any IPv6 packet and used to detect:

- IP path properties: Total hop number that is HbH-EH-aware node; The IP address of each HbH-EH-aware node.

- Static properties at each HbH-EH-aware node: Protocol version; Supported Flow identifying methods; Mapping index size; Supported configuration range of bandwidth, latency, forwarding QoS state time.

- Financial properties at each HbH-EH-aware node: Unit price for bandwidth; Unit price for service duration; Price for different latency.

After a QoS channel is established, OAM message can also be added as a HbH-EH to any IPv6 packet and used to detect and diagnose failures:

- IP path dynamic properties: Total end to end latency

- Dynamic properties at ach HbH-EH-aware node: Queue size; Remained bandwidth; Dropped packet number by different reasons.

- The detailed QoS forwarding failure reason.

3.5.3. Forwarding State and Forwarding State Report messages

Forwarding State and Forwarding State Report messages are used for data plane, See section 4.2.

3.5.4. Flow Identifying Methods

This is a parameter to program the HW for the flow identifying method. It is used for the QoS granularity definition and flow identification for QoS process. The QoS is enforced for a group of flows or a dedicated flow that can be identified by the same flow

identification. The QoS granularity is determined by the flow identification method during the setup and packet forwarding process. There are three levels of QoS granularities: Flow level, Address level and transport level. Each level of QoS granularity is realized by corresponding in-band signaling. The document focus on the flow level in-band signaling, other two level in-band signaling are discussed in the section 5.3.

There are two ways for the flow identifying method. One is by the tuples in IP header, another is by a local significant number (see mapping index) generated and maintained in a router. When "Mapping Index Size" (Mis) is zero, it means the "Flow identification method" (FI) is used for both control plane and data plane. When "Mis" is not zero, it means "FI" is only used in signaling, and the data plane will only use the "Mapping Index".

There are four types for "Flow identification method":

1. Individual Flow: Non-IPSec case: flow is identified by source and destination address, source and destination port number, and protocol number; IPSec case: flow is identified by source and destination address, flow label. For both case, FI = 0; the associated QoS is flow level, and QoS is guaranteed for a dedicated IP flow.

2. TCP flows: flow is identified by source and destination address, and TCP protocol number. The associated QoS is transport level, and QoS is guaranteed for TCP flows that have the same source and destination address. For this case, FI=1.

3. UDP flows: flow is identified by source and destination address, and UDP protocol number. The associated QoS is transport level, and QoS is guaranteed for UDP flows that have the same source and destination address. For this case, FI=2.

4. All flows: flow is identified by source and destination address. The associated QoS is address level, and QoS is guaranteed for all IP flows that have the same source and destination address. For this case, FI=3

The use of local generated number to identify flow is to speed up the flow lookup and QoS process for data plane. The number could be the MPLS label or a local tag for a MPLS capable router. The difference between this method and the MPLS switch is that there is no MPLS LDP protocol running and the IP packet does not need to be encapsulated as MPLS packet at the source host. When the MPLS label is used, the "Mapping Index Size" is 20 bits.

3.5.5. Hop Number

This is a parameter for the total number of hop that is HbH-EH-aware node on the path. it is the field "Hop_num" in Setup message. It is used to locate the bit position for "Setup State" and the "Mapping Index" in "Mapping Index List". The value of "Hop_num" must be decremented at each hop. And at the receive 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 by the OAM message.

3.5.6. Mapping Index, Size and Mapping Index List

Mapping Index is the local significant number generated and maintained in a router, and The "Mapping Index List" is just a list of "Mapping Index" for all hops that are HbH-EH-aware nodes on the IP path.

Mapping Index Size is the size for each mapping index in the Mapping Index List. The source host must know Mapping Index Size, and setup the initial value in the Setup message. The exact Mapping Index Size can be detected by the OAM message.

When a router receives a HbH-EH, it may generate a mapping index for the flow(s) that is defined by the Flow Identifying Method in "FL". Then the router must attach the mapping index value to the end of the Mapping Index List. After the packet reaches the destination host, the Mapping Index List will be that the 1st router’s mapping index as the list header, and the last router’s mapping index as the list tail.

3.5.7. QoS State and life of Time

After the chip 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.

3.5.8. 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, Mis, 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

4. Data plane

To support the QoS feature, there are couple of important requirements and schemes for implementations. These include the basic capability for the hardware, the scheme for the data forwarding, QoS processing, state report, etc.

Section 4.1 will talk about the basic capability for data plane, and section 4.2 will discuss the messages used for data plane after the QoS channel is established.

4.1. Basic Capability

The document only proposes the protocol used for control, and it is independent of the implementation of the system. However, 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). 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 QoS 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.

4.2. Forwarding State and Forwarding State Report

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 (Mis = 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 mapping index as identification for QoS process (Mis != 0).

   * The hardware was programmed to use the tuples in IP header as identification for QoS process (Mis = 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 mapping index 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 to the host in the reverse direction.

4.3. 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 "Mis" is zero or the Mapping index 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. See section 4.3

2. Hardware was programmed to use mapping index to identify flows. This is indicated by that the "Mis" is not zero. When a packet is received, the mapping index 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.

4.4. QoS Forwarding State Detection and Failure Handling

QoS forwarding may be failed 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 Forwarding State message coded as HbH-EH. After the TCP data packet 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 5.1

5. Other Issues

Above document only covers the details for the QoS support of individual TCP session by using the flow level in-band signaling. Due to the extensive scope of in-band signaling, there are many other associated issues for IP transport control. Below lists some of them, and we only brief the solution but do not go to details.

The details of each topic can be expressed in other drafts.

5.1. 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 in-band signaling is 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.

5.2. Traffic Management in Host

In order to accommodate in-band signaling and the QoS transport service, the OS on a host muts 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. Another is congestion control for TCP flows:

1. The current traffic management in a host manages traffic from different TCP/UDP session going out host link(s), in the way similar to routers to send traffic out. 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 will be much simpler. 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 the 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.

5.3. Non-shortest-path

The above method for the transport service with QoS is for the normal IP flows passing along the shortest path determined by the IGP or BGP. However, the IP shortest path may not be the best path in terms of the QoS. For example, the original IP path may not have enough bandwidth for a transport QoS service. The latency of the IP path is not the minimum in the network. There are two problems involved. One is how to find the best path for a QoS criteria, bandwidth or

latency. Another is how to setup the transport QoS for a non-shortest-path.

The 1st problem is out of scope of this document and many technologies have be discovered or in research.

The 2nd problem can be solved by combining the segment routing and in-band signaling. The use of the HbH-EH and Dst-EH is independent of the type of IP path, thus can be used with segment routing for any path determined by source. Note, the HbH-EH-aware nodes may not be different as the explicit IPv6 address in the segment routing header.

5.4. Heterogeneous Network

When IP network is crossing a non-IP network, such as MPLS or Ethernet network, the in-band signaling needs to be interworking with that network. The behavior, protocol and rules in the interworking with non-IP network is not the problem this document will address. More study and research need to be done, and new draft should be written to solve the problem.

5.5. Proxy Control

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

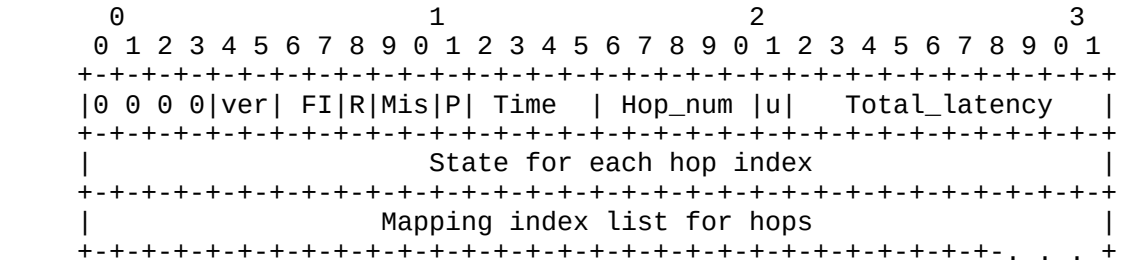
1. Proxy can check if a in-band signaling from end user for the SLA compliance, security and DOS attack prevention.

2. Proxy can collect the statistics for user’s TCP flows and check the in-band signaling for accounting and charging.

3. Proxy can insert and process appropriate in-band signaling for TCP flows that the host does not support the new feature, and this can provide the backward compatibility for host to use the new feature.

6. Message Format

6.1. Setup Msg



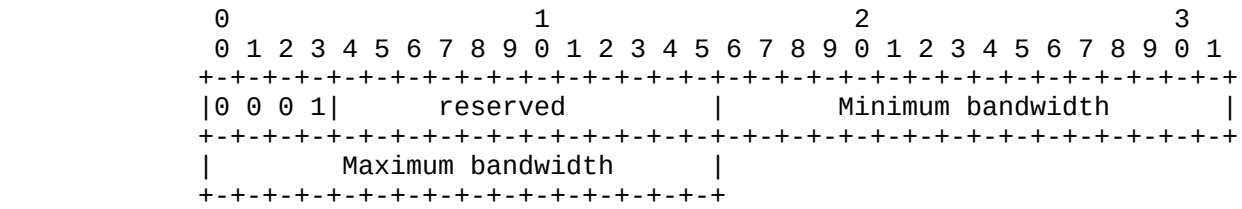
Type = 0, Setup state;
Version: The version of the protocol for the QoS
FI: Flow identification method,
0: 5 tuples; 1: src,dst,TCP; 2: src,dst,UDP; 3: src,dst
R: If the destination host report the received Setup state to the src address by Destination EH. 0: dont report; 1: report
Mis: Mapping index size; 0: 0bits, 1: 16bits, 2: 20bits, 3: 32bits
P: Programming the HW for QoS; 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.
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.
Mapping index list for hops: the mapping index list for all hops on the path, each index bit size is defined in Mis. The 1st mapping index is at the top of the stack. Each hop add its mapping index at the correct position indexed by the current hop number for the router.

Figure 2: The Setup message

The Setup message is embedded into the hop-by-hop EH to setup the QoS in the device on the IP forwarding path. At each hop, if the router is configured to process the header and to enforce the QoS, it must retrieve the hardware required information from the header, and then update some fields in the hader.

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 mapping index list is empty before the 1st hop receives the in-band signaling. Each hop then fill up the associated mapping index into the correct place determined by the index of the hop.

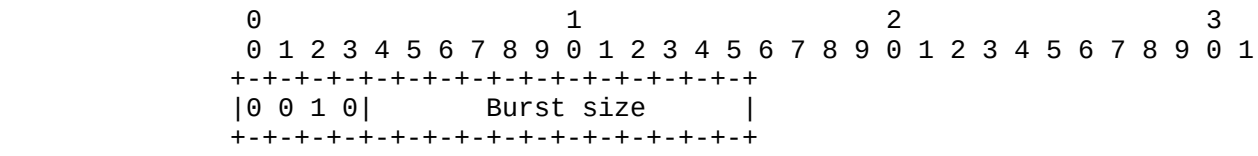
6.2. Bandwidth Msg



Type = 1,
Minimum bandwidth : The minimum bandwidth required, or CIR, unit Mbps
Maximum bandwidth : The maximum bandwidth required, or PIR, unit Mbps

Figure 3: The Bandwidth message

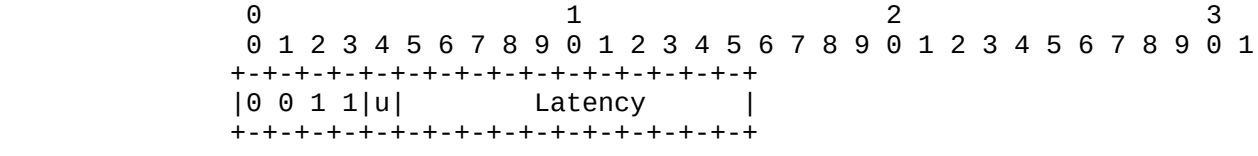
6.3. Burst Msg



Type = 2,
Burst size : The burst size, unit M bytes

Figure 4: The burst message

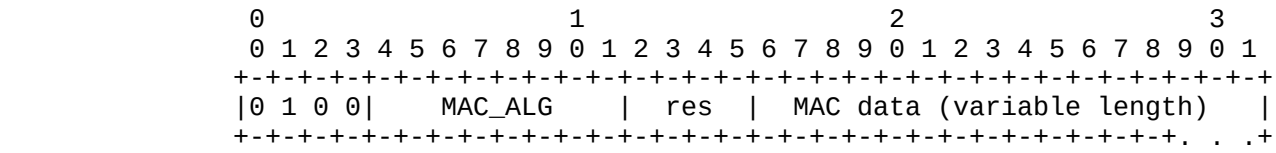
6.4. Latency Msg



Type = 3,
u: the unit of the latency
   0: ms; 1: us
Latency: Expected maximum latency for each hop

Figure 5: The Latency message

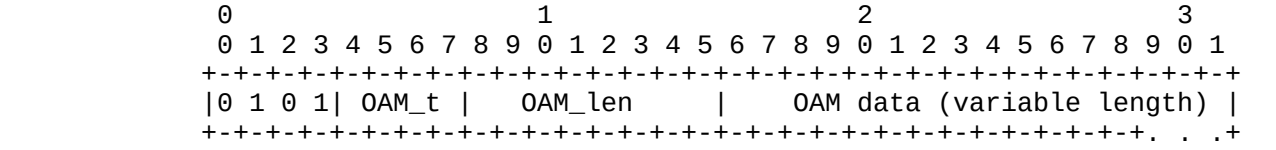
6.5. Authentication Msg



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 6: The Authentication message

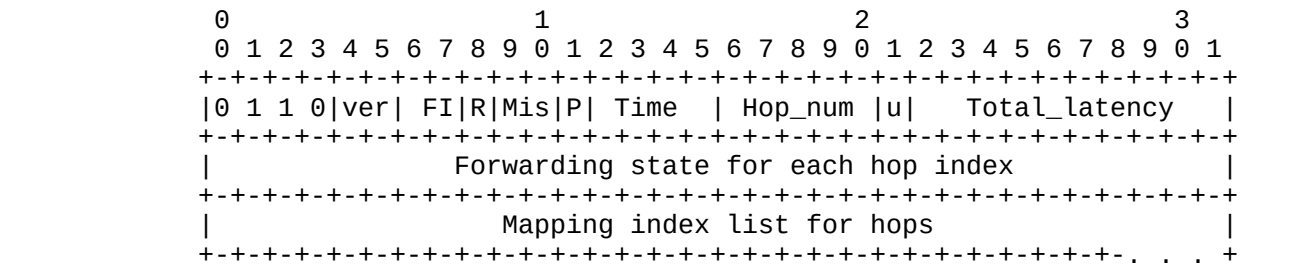
6.6. OAM Msg



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 7: The OAM message

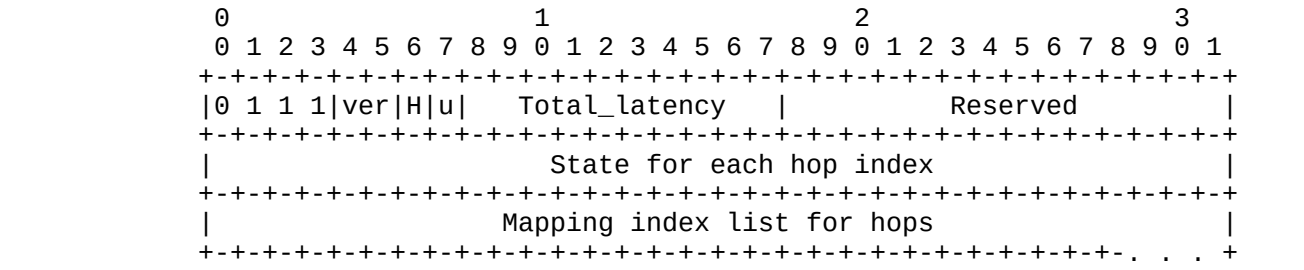
6.7. Forwarding State Msg



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 each hop on the path, 0: failed; 1: success; The 1st hop is at the most significant bit.
Mapping index list for hops: the mapping index list for all hops on the path, each index bit size is defined in Mis. The list is from the setup report message.

Figure 8: The Forwarding State message

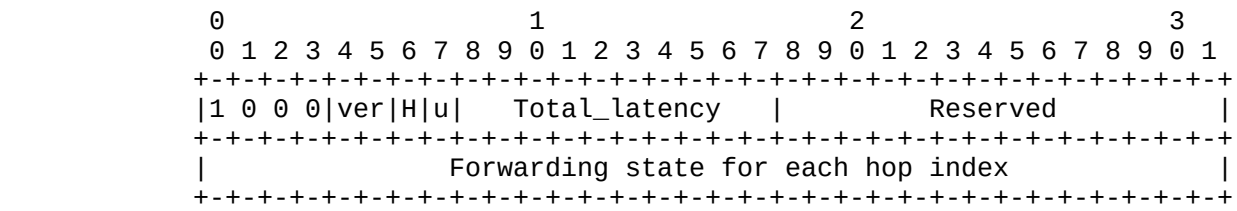
6.8. Setup State Report Msg



Type = 7, Setup state report;
H: Hop number bit. When a host receives a setup message and form a setup report message, it must check if the Hop_num in setup 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 setup message that if the original Hop_num was correct.
Following are directly copied from the setup message:
u, Total_latency;
State for each hop index
Mapping index list for hops.

Figure 9: The Setup State Report message

6.9. Forward State Report Msg



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 10: The Fwd State Report message

7. 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 6 in this doc

Figure 11: 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

Type	Mnemonic	Description	Reference
0	SETUP	Setup message	Section 6.1
1	BANDWIDTH	Bandwidth message	Section 6.2
2	BURST	Burst message	Section 6.3
3	LATENCY	Latency message	Section 6.4
4	AUTH	Authentication message	Section 6.5
5	OAM	OAM message	Section 6.6
6	FWD STATE	Forward state	Section 6.7
7	SETUP REPORT	Setup state report	Section 6.8
8	FWD REPORT	Forwarding state report	Section 6.9

Figure 12: The In-band Signaling Sub Type

8. Security Considerations

There is no security issue introduced by this document

9. Acknowledgements

We like to thank Huawei’s Nanjing research team leaded by Feng Li to provide the Product on Concept (POC) development and test, the team member includes Fengxin Sun, Xingwang Zhou, Weiguang Wang. We also like to thank other people involved in the discussion of solution: Tao Ma from Future Network Streategy dept.

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>.

[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>.

10.2. Informative References

[BBR] Neal Cardwell, et al, Google, "BBR Congestion Control", 2016, <https://www.ietf.org/proceedings/97/slides/slides-97-iccrg-bbr-congestion-control-02.pdf>.

[Cubic_throughput] Wei Bao, et al. The University of British Columbia, Vancouver, Canada, IEEE Globecom 2010 proceedings, "A Model for Steady State Throughput of TCP CUBIC", 2010, <https://www.researchgate.net/publication/224211021_A_Model_for_Steady_State_Throughput_of_TCP_CUBIC>.

[DiffServ] wiki, "Differentiated services", 2016, <https://en.wikipedia.org/wiki/Differentiated_services>.

[Fairness] Jain, R., et al. DEC Research Report TR-301, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems", 1984, <http://www1.cse.wustl.edu/~jain/papers/ftp/fairness.pdf>.

[Fastpass] Jonathan Perry, et al, MIT, "Fastpass: A Centralized ?Zero-Queue? Datacenter Network", 2014, <http://fastpass.mit.edu/Fastpass-SIGCOMM14-Perry.pdf>.

[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-iccrg-arvr-transport-problem]
Han, L. and K. Smith, "Problem Statement: Transport Support for Augmented and Virtual Reality Applications", draft-han-iccrg-arvr-transport-problem-01 (work in progress), March 2017.

[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.

[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-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.

[IntServ] wiki, "Integrated services", 2016, <https://en.wikipedia.org/wiki/Integrated_services>.

[IPv6_Parameters]
IANA, "Internet Protocol Version 6 (IPv6) Parameters", 2015, <https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-2>.

[PCC] Mo Dong, et al, University of Illinois at Urbana-Champaign, Hebrew University of Jerusalem, "PCC: Re-architecting Congestion Control for Consistent High Performance", 2014, <https://arxiv.org/abs/1409.7092>.

[PERC] Lavanya Jose, et al, Stanford University, MIT, Microsoft, "High Speed Networks Need Proactive Congestion Control", 2016, <http://web.stanford.edu/~lavanyaj/papers/perc-hotnets15.pdf>.

[RCP] Nandita Dukkipati, Ph.D. Thesis, Department of Electrical Engineering, Stanford University, "Rate Control Protocol (RCP): Congestion control to make flows complete quickly", 2007, <http://yuba.stanford.edu/~nanditad/thesis-NanditaD.pdf>.

[Reno_throughput]
Matthew Mathis, et al, Pittsburgh Supercomputing Center, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", 1997, <https://cseweb.ucsd.edu/classes/wi01/cse222/papers/mathis-tcpmodel-ccr97.pdf>.

[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-cubic]
Ha, S., Rhee, I., and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant", 2008.

[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.

[TIMELY] Radhika Mittal, et al. Google, Inc., "TIMELY: RTT-based Congestion Control for the Datacenter", 2010, <http://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p537.pdf>.

Authors' Addresses

Lin Han (editor)
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

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

Guoping Li
Huawei Technologies
Beijing
China

Email: liguoping@huawei.com

Boyan Tu
Huawei Technologies
Beijing
China

Email: tuboyan@huawei.com

Xuefei Tan
Huawei Technologies
Beijing
China

Email: tanxuefei@huawei.com

Frank Li
Huawei Technologies
Nanjing
China

Email: frank.lifeng@huawei.com

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

Email: renwei.li@huawei.com

Jeff Tantsura

Email: jefftant.ietf@gmail.com

Kevin Smith
Vodafone
UK

Email: Kevin.Smith@vodafone.com