ETSI GR NGP 010 V1.1.1 (2018-09)

Next Generation Protocols (NGP); Recommendation for New Transport Technologies

Disclaimer

The present document has been produced and approved by the Next Generation Protocols (NGP) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

DGR/NGP-0010

Keywords

Next Generation Protocol, QoS, transport

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Next Generation Protocols (NGP).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The present document focuses on new transport technology for next generation architectures toward 5G and beyond. The basic concept is to enhance the best-effort based IP network to QoS capable IP network. The goal is to provide the QoS for the upper layer protocols. The work aims to examine and propose recommendations to improve and simplify the network infrastructure to support QoS for different transport protocols. In addition, the present document may require the development of new protocols and or modification of existing protocols.

Introduction

Recently, more and more new applications for Internet are emerging. These applications have a common requirement to the Internet that is their required bandwidth is very high and/or latency is very low compared to traditional applications like most of web browser and video streaming applications.

For example, AR or VR applications may need at least 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.1].

Some future applications expect that Internet can provide a up bounded latency (minimized latency) service, such as tactile network [i.2]. To these applications, the latency will determine their user experience or application quality, so it is critical that the maximum latency for application is bounded within values application has requested.

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.

The present document will brief the current IP transport and QoS technologies, and analyse the limitations to support above new applications.

A frame work for new transport technology based on QoS enabled IP network will be reported. As an example, detailed design and experiments for TCP are given.

The frame work also lists other areas, topics and issues that need more study to achieve the complete solution.

1 Scope

The present document reports the analysis of current transport technologies for Internet, especially TCP, the limit of different variants for TCP and other transport protocols, and then proposes a framework for new transport technology for IP network. TCP is exemplified for the detailed design and prove of concept experiments.

In the design, both control plane and data plane are discussed. It includes the control mechanism, message type, key message parameters, hardware capability, forwarding state, host congestion control and traffic management.

In the experiments, the POC product and its realization are discussed; test results, scalability and performance are analysed.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] Draft-han-iccrg-arvr-transport-problem-01 (work in progress): "Problem Statement: Transport Support for Augmented and Virtual Reality Applications", L. Han, and K. Smith, March 2017.

[i.2] Proceedings of European Wireless 2015; 21th European Wireless Conference: "Towards the Tactile Internet: Decreasing Communication Latency with Network Coding and Software Defined Networking", J David Szabo, 2015.

NOTE: Available at https://ieeexplore.ieee.org/iel7/7147658/7147659/07147730.pdf.

[i.3] DEC Research Report TR-301: "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems", R. Jain, 1984.

NOTE: Available at http://www1.cse.wustl.edu/~jain/papers/ftp/fairness.pdf.

[i.4] Andreas Benthin, Stefan Mischke, University of Paderborn: "Bandwidth Allocation of TCP", 2004.

[i.5] IETF RFC 2581: "TCP Congestion Control", M. Allman, V. Paxson and W. Stevens, April 1999.

NOTE: Available at https://www.rfc-editor.org/info/rfc2581.

[i.6] L. Peterson: "TCP Vegas: New Techniques for Congestion Detection and Avoidance - CiteSeer page on the 1994 SIGCOMM paper", 1994.

[i.7] S. Ha, I. Rhee and L. Xu: "CUBIC: A New TCP-Friendly High-Speed TCP Variant", 2008.

[i.8] Draft-sridharan-tcpm-ctcp-02 (work in progress): "Compound TCP: A New TCP Congestion Control for High-Speed and Long Distance Networks", M. Sridharan, K. Tan, D. Bansal and D. Thaler, November 2008.

[i.9] Radhika Mittal, Vinh The Lam, Nandita Dukkipati, Emily Blem, Hassan Wassel, Monia Ghobadi, Amin Vahdat, Yaogong Wang, David Wetherall, David Zats: "TIMELY: RTT-based Congestion Control for the Datacenter", 2010.

NOTE: Available at http://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p537.pdf.

[i.10] Draft-falk-xcp-spec-03 (work in progress): "Specification for the Explicit Control Protocol (XCP)", A. Falk, Jul 2007.

[i.11] Nandita Dukkipati, Ph.D. Thesis, Department of Electrical Engineering, Stanford University: "Rate Control Protocol (RCP): Congestion control to make flows complete quickly", 2007.

NOTE: Available at http://yuba.stanford.edu/~nanditad/thesis-NanditaD.pdf.

[i.12] Draft-ietf-tcpm-dctcp-03 (work in progress): "Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters", S. Bensley, L. Eggert, D. Thaler, P. Balasubramanian, and G. Judd, November 2016.

[i.13] Draft-ietf-aqm-pie-10 (work in progress): "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", R. Pan, P. Natarajan, F. Baker, and G. White, September 2016.

[i.14] Draft-ietf- aqm-codel-06 (work in progress): "Controlled Delay Active Queue Management", K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, December 2016.

[i.15] Draft-ietf-aqm-fq-codel-06 (work in progress): "The FlowQueue-CoDel Packet Scheduler and Active Queue Management Algorithm", T. Hoeiland-Joergensen, P. McKenney dave.taht@gmail.com, J. Gettys and E. Dumazet, March 2016.

[i.16] Lavanya Jose, Mohammad Alizadeh, George Varghese, Nick McKeown, Sachin Kattie: "High Speed Networks Need Proactive Congestion Control", 2016.

NOTE: Available at http://web.stanford.edu/~lavanyaj/papers/perc-hotnets15.pdf.

[i.17] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh,Van Jacobson: "BBR Congestion Control", 2016.

NOTE: Available at https://www.ietf.org/proceedings/97/slides/slides-97-iccrg-bbr-congestion-control-02.pdf.

[i.18] Mo Dong, University of Illinois at Urbana-Champaign, Hebrew University of Jerusalem: "PCC: Re-architecting Congestion Control for Consistent High Performance", 2014.

NOTE: Available at https://arxiv.org/abs/1409.7092.

[i.19] Jonathan Perry: "Fastpass: A Centralized "Zero-Queue" Datacenter Network", 2014.

NOTE: Available at http://fastpass.mit.edu/Fastpass-SIGCOMM14-Perry.pdf.

[i.20] Matthew Mathis, Pittsburgh Supercomputing Center: "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", 1997.

NOTE: Available at https://cseweb.ucsd.edu/classes/wi01/cse222/papers/mathis-tcpmodel-ccr97.pdf.

[i.21] Wei Bao, The University of British Columbia, Vancouver, Canada, IEEE Globecom 2010 proceedings: "A Model for Steady State Throughput of TCP CUBIC", 2010.

NOTE: Available at https://www.researchgate.net/publication/224211021_A_Model_for_Steady_State_Throughput_of_TCP_CUBIC.

[i.22] IETF RFC 2475: "An Architecture for Differentiated Services".

NOTE: Available at https://www.rfc-editor.org/info/rfc2475.

[i.23] IETF RFC 1633: "Integrated Services in the Internet Architecture: an Overview".

NOTE: Available at https://www.rfc-editor.org/info/rfc1633.

[i.24] IETF RFC 8200: "Internet Protocol, Version 6 (IPv6) Specification".

NOTE: Available at https://www.rfc-editor.org/info/rfc8200.

[i.25] Draft-harper-inband-signalling-requirements-00 (work in progress): "Requirements for In-Band QoS Signalling", J. Harper, January 2007.

[i.26] Draft-roberts-inband-qos-ipv6-00 (work in progress): "In-Band QoS Signaling for IPv6", L. Roberts and J. Harford, July 2005.

[i.27] IETF RFC 4782: "Quick-Start for TCP and IP".

NOTE: Available at https://www.rfc-editor.org/info/rfc4782.

[i.28] IETF RFC 5971: "GIST: General Internet Signalling Transport".

NOTE: Available at https://www.rfc-editor.org/info/rfc5971.

[i.29] IETF RFC 2460: "Internet Protocol, Version 6 (IPv6) Specification".

NOTE: Available at https://www.rfc-editor.org/info/rfc2460.

[i.30] IETF RFC 6275: "Mobility Support in IPv6".

NOTE: Available at https://www.rfc-editor.org/info/rfc6275.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

deterministic IP: term contrast to best-effort IP and intend to represent new IP that has QoS support for bandwidth and minimum latency

NOTE: It is similar to the objectives of IETF Detnet WG.

in-band signaling: control information sent within the same band or channel used for user data

IP flow: data flow identified by the source, destination IP address, the protocol number, the source and destination port number

IP path: route that IP flow will traverse

NOTE: IP path 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.

out-of-band signaling: control information sent over a different channel, or even over a separate network

QoS channel: forwarding channel that the QoS is guaranteed so to provide additional QoS service to the normal IP forwarding

NOTE: A QoS channel can be used for one or multiple IP flows depends on the granularity of in-band signaling.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACK	Acknowledge
ACL	Access Control List
AIMD	Additive-Increase/Multiplicative-Decrease

API	Application Program Interface
AQM	Active Queue Management
AR	Augmented Reality
ATN	Access Transport Network
BBR	Bottleneck Bandwidth and RTT
BGP	Board Gateway Protocol
BRAS	Broadband Remote Access Server
BRS	Burst Size
CDF	Cumulative Distribution Function
CIR	Committed Information Rate
CPU	Central Process Unit
CSFQ	Core-Stateless Fair Queuing
DCTCP	Data Center TCP
DHCP	Dynamic Host Configuration Protocol
DIP	Deterministic IP
DNS	Domain Name Service
DOS	Denial Of Service
DPI	Deep Packet Inspection
DSCP	Differentiated Services Code Point
Dst-EH	IPv6 Destination Extension Header
EH	IPv6 Extension Header or Extension Option
EPC	Evolved Packet Core
FI	Flow Identification
HbH-EH	IPv6 Hop-by-Hop Extension Header
HbH-EH-$_{\text{aware node}}$	Network nodes that are configured to process the IPv6 Hop-by-Hop Extension Header
HOPOPT	Hop Option
HW	Hardware
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IP	Internet Protocol
IW	Initial Window
LDP	Label Distribution Protocol
MEC	Mobile Edge Computing
MPLS	Multi-Protocol Label Switching
MPTCP	Multi-Path TCP
MS	Multi-Segment
MSS	Multi-Segment Size
NPU	Network Process Unit
NSIS	Next Steps In Signaling
OAM	Operation And Management
OS	Operating System
PCC	Performance-oriented Congestion Control
PDN	Packet Data Network
PERC	Proactive Congestion Control Algorithm
PGW	PDN Gateway
PIE	Proportional Integral controller Enhanced
PIR	Peak Information Rate
PLR	Packet Loss Ratio
POC	Prove Of Concept
QoS	Quality of Service
RCP	Rate Control Protocol
RFC	Request for Comments
RMCAT	RTP Media Congestion Avoidance Techniques
RSVP	Resource Reservation Setup Protocol
RTCP	Real Time Control Protocol
RTP	Real-time Transport Protocol
RTT	Round Trip Time
SCTP	Stream Control Transmission Protocol
SIS	Service ID Size
SLA	Service Level Agreement
SP	Service Provider

SYN	Synonym
TC-ACK	TCP acknowledgement packet
TCP	Transport Control Protocol
TM	Traffic Management
TOR	Top-Of-Rack
UDP	User Datagram Protocol
VR	Virtual Reality
WFQ	Weighted Fair Queuing
WG	Working Group
XCP	eXplicit Control Protocol

4 Introduction

4.1 IP and Transport Technologies

This clause briefs the IP and transport protocol and technologies.

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

- 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 (QoS) to end-user or application, such as bandwidth, packet loss, latency, jitter, etc. Due to this fact, the transport layer or application will have its own control mechanism for congestion and flow to obtain the reliable and satisfactory service to cooperate with the under layer network quality.

- 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 with maximized fairness [i.3]. The process includes scheduling, queuing and forwarding for all IP flows equally. Thus, the transport layer is supposed to behave nicely and friendly to make sure all flows will only obtain its own faired share of resource, and no one could consume more resource and no one could be starved.

Clause 4.2 briefs the analysis of current transport related technologies including TCP, UDP, DiffServ, IntServ, and MPLS. The major focus is TCP since it is the most widely used and the most complicated transport protocol.

4.2 TCP Solution Analysis

4.2.1 TCP Overview and Evolution

As a most popular and widely used transport technology, TCP is the most popular transport protocol in Internet. TCP traffic is actually dominating Internet from the birth of Internet. It is key to analyse TCP to get any conclusion for the current transport technology, and give any new proposal. This clause will brief the TCP, its variations and some key characteristics.

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 host 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 will wait for an acknowledgment and window update from the receiving host.

The congestion control is the algorithm 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 state and adjust the rate of source to avoid the congestion.

No matter what congestion control algorithm is used, all classical TCP solutions are pursuing three targets [i.4]:

1) Higher efficiency in bandwidth utilization.

2) More fairness in bandwidth allocation.

3) Faster convergence to the equilibrium state.

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

4.2.2 TCP Solution Variants

There are many TCP variants and optimization solutions since TCP was introduced 40 years ago. Below lists the major TCP variants including typical classical solution and some contemporary solutions proposed recently:

- The classical 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 [i.5], TCP-vegas [i.6], TCP-cubic [i.7], TCP-compound [i.8], TIMELY [i.9], 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 notify host to adjust accordingly. Both network devices and hosts need to be changed in software and/or hardware. Typical solutions are: XCP [i.10], RCP [i.11].

NOTE: XCP and RCP are described for 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 or queuing, thus control the congestion and minimize the latency indirectly. Both network devices and hosts may need to be changed in software and/or hardware. They include: DCTCP [i.12], PIE [i.13], CoDel [i.14], FQ-CoDel [i.15], etc.
- The new concept solutions:
  - Unlike above categories, the category of these solutions use completely new concepts and methods to either accurately calculate, or figure out the optimized rate and latency for TCP, such as: PERC [i.16], BBR [i.17], PCC [i.18], Fastpass [i.19], etc.

4.2.3 TCP Throughput Constraints

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 ratio of the total throughput of all TCP flows on a network device to the network bandwidth of all 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 IP (including TCP, UDP and all other type of IP protocols) flows sharing the same link, host CPU power, network device congestion status, physical propagation delay in transmission, etc.

For traditional TCP, the real throughput for a flow is limited by three factors: The 1st factor is the available maximum throughput at the physical layer. It is related to the maximum theoretical bandwidth, network load, buffering configuration, maximum segment size, signal strength, etc. The 2nd one is related to congestion control algorithm. The 3rd is related to TCP fairness principle. Clauses below will analyse the last two factors, and the 1st factor is pretty steady and there is not much the transport technology can do:

1) By Algorithm:

   - No matter what algorithm is used, TCP throughput is always related to 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 [i.20]. And TCP- cubic throughput is expressed in formula (21) in [i.21].

   - 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 for the case that there is only one TCP flow in a link, the throughput of the TCP flow could be way below the link capacity for a network which RTT and PLR are high. By the formula (3) in [i.20], the real throughput can be easily calculated for any RTT/PLR values for TCP-reno.

2) By Fairness Principal:

   - TCP fairness [i.6] is a de facto principle for all TCP solutions currently. 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 [i.3] 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 RCP analysis [i.11] 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.

4.2.4 TCP Latency Constraints

According to the principal TCP fairness, network device 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.

4.2.5 Summary of TCP Solution

The bandwidth (throughput) 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 throughput.

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

It should 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 efficiency of network bandwidth utilization.

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

4.3 UDP Solution Analysis

UDP protocol is a non-reliable transport technology. It can only provide the multiplexing functionality for the data streams by the port number allocated at each end host, and there is no flow control and congestion control to provide the reliability and optimized bandwidth for UDP application.

At network devices doing IP forwarding in a network, there is no distinction between TCP, UDP or other transport protocol unless there is special configuration to do so. Forwarding engine will treat all IP flows (TCP, UDP or other protocols) in the same way. The fairness principal and its consequence will also apply to UDP flows. That means the throughput that an application can obtain is also limited by the fairness principal described in clause 4.2.3.

Recently, there are efforts to introduce the congestion control for UDP, IETF RMCAT WG (RTP Media Congestion Avoidance Techniques) has been trying to use the congestion algorithm at end host to avoid the congestion for UDP used in multicast media streams. For example, there are two work in progress: "Coupled congestion control for RTP media" and "NADA: A Unified Congestion Control Scheme for Real-Time Media". The essence of the congestion control in UDP is similar to TCP. They try to detect the congestion state on network by feedback (RTCP) and adjust the RTP (over UDP) stream rate.

4.4 Other Solution Analysis

- DiffServ:
  - DiffServ [i.22] 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. It cannot support the case that a specified TCP is expecting QoS. Moreover, end user application cannot directly use DiffServ.
- IntServ:
  - IntServ [i.23] 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 the scale of a network increases, 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 needs to 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 provider's 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.

4.5 New Transport Technology Overview

4.5.1 Fundaments

What is New Transport Technology for IP based network? This is the 1st question the present document should answer. From the analysis above, following conclusions can be obtained:

1) The new transport technology should aim to provide service that all current transport technologies (based on the best-effort-only IP network) could not provide. The current transport technology (TCP/UDP/others), no matter what form and what variants, cannot break the barrier caused by best-effort-only service in IP layer, so it cannot provide the user-demanded throughput and latency to applications.

2) New transport technology will be naturally born if IP based Internet is changed from best-effort-only to multiple service capable network.

With the above conclusion, the present document suggests that a new transport technology is based on a new IP network that could provide QoS for upper layer protocols (TCP/UDP).

NOTE: The new transport technology proposed by the present document is for IP network. All IP architecture and protocols, such as the basics of IP packet, routing protocols and IP forwarding mechanism are kept for the backward compatibility. The present document is belonging to IP theme in NGP.

In the present document, the new transport technology is composed of two major parts:

- Make the best-effort-only IP to be new IP that is able to provide multiple service support (bandwidth guarantee, or bounded latency guarantee).
- Enhance the existing transport technology (TCP/UDP/SCTP/QUIC, etc.) to be adaptive to the new IP to obtain the benefits of QoS.

4.5.2 Design Guidance

Based on the above analysis, some high level guidance and directions of new transport technology can be predicted to achieve the target to support the emerging application requirement:

1) **The network device should be involved deeply:**

   - Classical TCP uses the host-only congestion control approach, it cannot accurately detect the congestion state quickly and cannot control the resource allocation for different flows. New TCP variants such as AQM, DCTCP have proved the benefit and advantages of using network device for congestion control and minimized latency for all TCP flows. Moreover, more involvement of network device, more benefits can be obtained to control the transport service. This can be demonstrated by the difference between DiffServ and IntServ, IntServ can allocate the resource for individual flows, so, its QoS granularity is finer than DiffServ, and the QoS control is more efficient and satisfactory.

2) **The fairness principal should be relaxed:**

   - The current fairness principal prevents the differentiation of processing and resource allocation for different flows. That is one of the initiatives and reasons that MPTCP is used to achieve higher bandwidth since one TCP session can only obtain the bandwidth that is limited by the algorithm and fairness principal as analysed in the clause 4.2.3. Only after the relaxation of fairness principal, different flows can be processed differently to obtain different network resource. Also, the relaxation can give us the flexibility to ONLY process required flows differently while keeping other flows processed like before. This can dramatically reduce the scalability requirement for the whole network.

3) **The control plane should be as simple as possible to obtain the high scalability and performance:**

   - From the lesson of history, the control protocol for new transport technology should not take the same approach as RSVP, RSVP-TE, etc. since those protocols cannot satisfy the scalability and performance requirement for Internet. If the new transport technology can be used cross different domains, or even for the Internet, the protocol should be very simple.

4) **The data plane should be based on IP:**

   - IP is the basic protocol for Internet, IP based data plane will make two things much easier, one is the wide acceptance and deployment of new technology, and another is the application on host can directly use the new transport service. If non-IP data plan is used, it is inevitable that other protocols are required and maintained, for example, MPLS needs to run LDP. Currently, there is no sign that any public host likes to accept the protocol other than IP.

5) **Latest hardware technology should be used:**

   - Semiconductor chip technology has advanced a lot for last decades, the widely used network process (NPU) cannot 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 (DPI), etc. To 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.

4.5.3 Design Targets

The new transport service is expected to satisfy the 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. It is for applications that current transport technology could not support, and not to replace traditional TCP/UDP for scenarios that TCP/UDP work well.

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

4.5.4 Assumptions

To limit the scope of the present document and simplify the design and solution, the following constraints are given:

1) IP network architecture are not changed, this includes the basics of IP packet format, IP routing architecture, IP forwarding mechanism.

NOTE: The extension header described in clauses 5.5.1 and 5.6 does not change the basics of IPv6 and IPv4 packet format respectively.

2) The transport service 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 or UDP completely that have been proved to be efficient and successful for current applications.

3) The framework focus on one administrative domain, but 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.

4) 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.

5) The new service should 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.

4.5.5 Architecture of Framework

The framework is composed of following parts:

- Network Control Plane Framework, it explains a transport control sub-layer for IP in network device, the details of control mechanism.

- Network Data Plane Framework, the realization of QoS in data forwarding, QoS and error handling in network device.

- Host Congestion Control, the congestion control mechanism in host to utilize the new transport technology.

5 Network Control Plane Framework

5.1 Introduction

This clause will describe the framework for control plane for new transport technology. The framework is composed of following clauses:

- Clause 5.2 describes a new sub-layer to control the transport protocol for QoS and associated features.
- Clause 5.3 will talk about the IP in-band signaling and different granularity.
- Clause 5.4 will discuss the details of control mechanism.
- Clause 5.5 lists the details of approach for IPv6 [i.24].
- Clause 5.6 gives the proposals for IPv4.

5.2 Sub-layer in IP for Transport Control

In order to provide new features for transport layer on top of IP, it is very useful to introduce an additional sub-layer, layer 3.5, between layer 3 (IP) and layer 4 (TCP/UDP). The new layer belongs to IP and provides the Transport Control functionalities, and is present ONLY when the transport control is needed. The sub-layer is the most important part of the control plane for new transport technology.

Figure 1 illustrates a new stack with the sub-layer 3.5.

```
+=========================+
|           APP           |
+=========================+
|         TCP/UDP         |
+=========================+             -+
|      Transport Ctl      | <- Layer 3.5 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+              |-> New Layer 3
|           IP            |              |
+=========================+             -+
|     Network Access      |
+=========================+
```

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

The new sub-layer is always bounded with IP layer, and can provide following functionalities for upper layer, such as:

1) **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.

2) **Congestion control:**

   - The congestion state in each device on a path can be detected and notified to the source of IP 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.

3) **IP Path OAM:**

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

IPv6 extension header [i.24] can be used to realize the sub-layer functionalities for IPv6.

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

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

Clause 5.6 will propose to extend IPv4 header to realize the complete control plane functionality as IPv6.

5.3 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.

IP in-band signaling concept is not new. DiffServ is a kind of IP in-band signaling. In DifferServ, IP header DSCP fields of user data will carry the message for differentiated Service, different DSCP codes represent different service for the data stream.

IP in-band signaling is the key function for the new sub-layer, and is the fundament of the new transport technology. It will provide the critical control functions for transport protocols to achieve the QoS.

In the present document, four types of "in-band signaling" are introduced for different signaling granularity:

1) **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.

2) **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 and destination address, i.e. signaling messages for all TCP/UDP flows between the same client and same server (only one address for client and one for server).

3) **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.

4) **DiffServ level In-band Signaling:**

   - The control message and data packet share the same DSCP value. This is the situation that the signaling is for the 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.

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:

1) **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.

2) **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.

3) **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.25]. And the in-band QoS signaling for IPv6 was simply discussed in [i.26]. Unfortunately, both works did not continue to achieve its original goals. Actually, TCP quick start standard IETF RFC 4782 [i.27] was partially from the requirement of [i.25]. Also, the NSIS WG was trying to solve some problems raised in in-band signaling, but the outcome of the protocol IETF RFC 5971 [i.28] does not prevail in industry.

The present document not only discusses detailed solution for in-band signaling, but also tries to address issues applied to all protocols, such as security, scalability and performance. Finally, experiments with proprietary hardware and chips are given in clause 9.

5.4 Control Mechanism

5.4.1 Protocol Driven In-band signaling

By the criteria of protocol owner, there are two categories of In-band signaling, one is user protocol driven and another is control protocol driven.

1) User protocol driven:

- For this category, user can initialize an in-band signaling by L4 user protocol, TCP or UDP. In-band signaling message is embedded into user protocol or user data packet. For example, TCP three-hand shaking message (TCP-SYN, TCP-SYNACK, and TCP-ACK) or even TCP data can carry the in-band signaling information. Normally, user protocol driven in-band signaling is used for a user's IP flow, and belongs to flow-level in-band signaling.

2) Control protocol driven:

- Network administrator can initialize a separate protocol to invoke the in-band signaling. The in-band signaling message is carried into a new protocol, and the new protocol share some common information for IP header with user data. For example, a separate administer controlled TCP session (control session) can be used to setup QoS path for aggregated TCP or UDP traffic. The control session uses the TCP message to carry the in-band signaling information. It is normally used for address level and transport level in-band signaling.

5.4.2 Closed-loop and Open-loop Control by In-band Signaling

By the criteria of control theory, there are two categories of In-band signaling, one is closed-loop control and another is open-loop control.

1) Closed-loop control

- For this category, 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 may be 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

- For this category, 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 is needed.

For both closed-loop and open-loop control, the signaling message for one direction is used 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 should 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.

5.4.3 Scope of Solution

The scope of complete solutions for new transport technology by in-band signaling is pretty big and cannot be completely covered by the present document. Many topics are to be researched further as stated in clause 8. In the present document, only TCP is exemplified for in-band signaling to achieve the goal of new transport technology. According to the criteria described in clauses 5.3, 5.4.1 and 5.4.2, the demonstrated solution in clause 5.5 is a **user protocol driven, closed-loop controlled flow-level in-band signaling method**.

Figure 2 illustrates a closed loop control by IP in-band signaling for the QoS path setup for TCP.

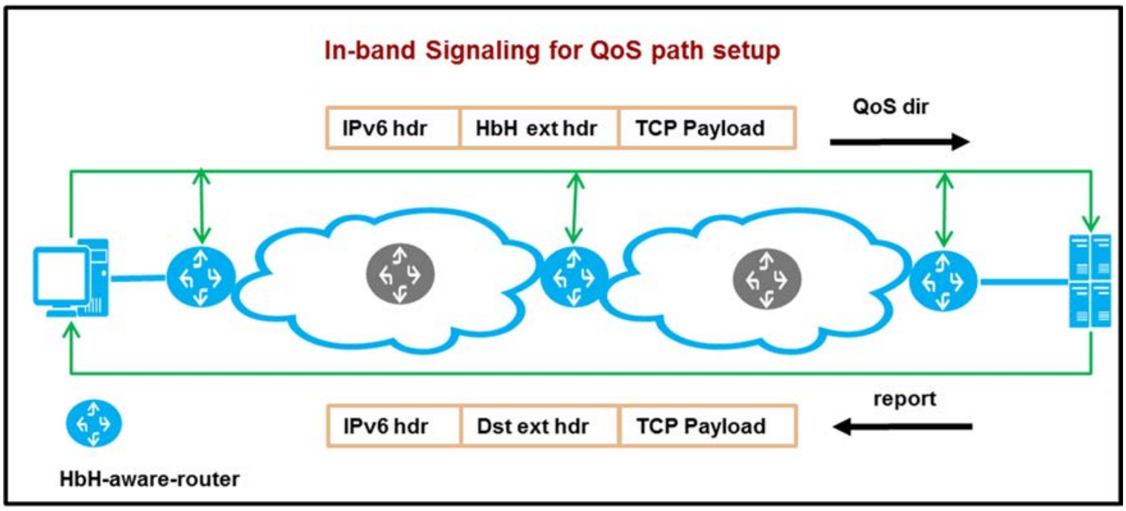


Figure 2: In-band signaling for QoS path setup for TCP

5.5 IPv6 Solution

5.5.1 Overview

The IPv6 [i.24] 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 [i.24]. These nodes are called as HbH-EH-aware nodes in the clauses 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.

5.5.2 Control Scenarios for TCP

The finest grained QoS for TCP is flow level, the present 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 not discussed.

Normally, TCP with flow level QoS feature comprises following basic controls:

1) **QoS Setup:**

   - The QoS 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 setup state, QoS requirements, such as max/min bandwidth, burst size and the latency. 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 may need to 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:

     a) HW state refreshing:

        - The HW state for QoS programming is data driven (see clause 5.5.4.7 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.

     b) 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. Clause 5.5.3.3 will explain more about this scenario.

     c) 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. Clause 6.5 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 behaviour of the application. This is discussed in clause 7.3.

5.5.3 Details of In-band Signaling for TCP

5.5.3.1 Message Type

As a report, the present document does not intend to design the protocol or method in details, such as accurate mechanism including state machine, message format, etc. The present document only proposes some key message types and key message contents. This is to make the description more logical and more convenient for readers to understand.

Although the detailed message format is attached in Annex A, it is only to make readers understand the organization of different fields for the message of in-band signaling, but not to mean the message is finalized.

The final standard for in-band signaling, including the message format and other aspects of control protocol, is up to a standard organization, such as IETF, to design and approve.

The present document introduces following four types of message for in-band signaling and associated data forwarding, the detailed discussion of messages is expressed in clause 5.5.4:

1) **Setup:** This is a set of messages for the setup signaling of QoS channel through the IP path. The QoS forwarding is for the same direction as the Setup messages forwarded. Setup messages may have following sub-messages included:

    a) Setup State: This is the mandatory and the 1st sub-message for setup messages. It has the most key information about the setup. See clause 5.5.4.1 for details of Setup State message.

    b) Bandwidth: This is the required bandwidth for the QoS channel. It has minimum (CIR) and maximum bandwidth (PIR). Bandwidth message is an optional message and can be attached to the Setup State message.

    c) 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. Latency message is an optional message and can be attached to the Setup State message.

    d) Burst: This is the required burst for the QoS channel, it is the maximum burst size. Burst message is an optional message and can be attached to the Setup State message.

    e) Authentication: This is the security message for an in-band signaling. Authentication message is an optional message and can be attached to the Setup State message.

    f) OAM: This is the Operation and Management message for the QoS channel. OAM message is an optional message and can be attached to the Setup State message.

2) **Setup State Report:** This is the state report of a setup message. It majorly contains Setup state information received from the Setup messages. More detailed discussion for Setup State Report is in clause 5.5.4.1.

3) **Forwarding State:** This is the message used for data packet forwarding with QoS guarantee. More detailed discussion for forwarding message is in clause 6.2.1.

4) **Forwarding State Report:** This is the forwarding state report of a packet forwarding. More detailed discussion for Forwarding State Report message is in clause 6.2.2.

5.5.3.2 Basic QoS Setup Scenarios

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

1) Upstream:

- This is for the direction of client to server. Figure 2 demonstrates the path setup and data forwarding for upstream. An 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 messages and it may include Setup State message, 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:

    a) Retrieve the QoS parameters to program the Hardware, it includes: Flow identification method, the life time of QoS forwarding state, Bandwidth, Latency, Burst, etc.

b) Update the field in the EH, it includes: State for each hop index, 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 should form a new Setup State Report message, all fields in the message should be copied from the Setup messages in the HbH-EH. When the TCP stack is sending the TCP-SYNACK to the client, the kernel should 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 should read the Setup State Report message. Depending on the setup state, the client will operate according to new congestion control mechanism described in clause 7.3.

- Figure 3 demonstrates the scenario of the path setup and data forwarding for upstream from client to server.

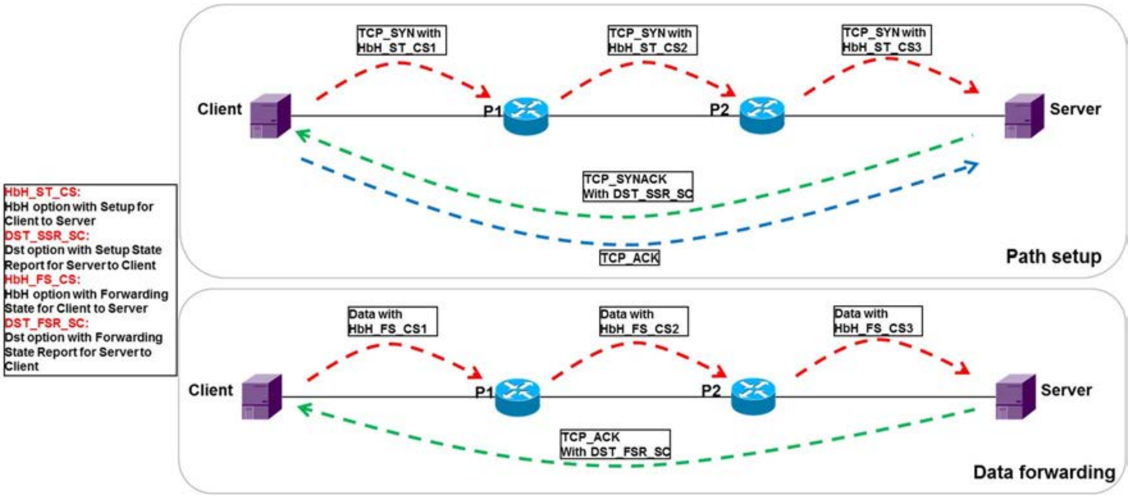


Figure 3: QoS path setup and data forwarding for the upstream from client to server

2) Downstream:

- This is for the direction of server to client. Figure 4 demonstrates the path setup and data forwarding for downstream. An 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 messages 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 QoS path according to the pre-negotiated QoS requirements.

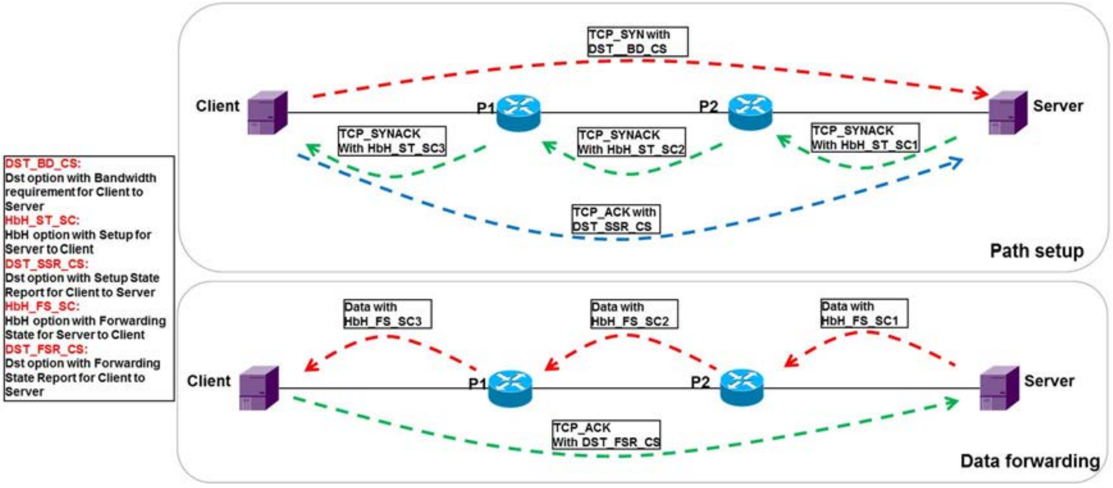
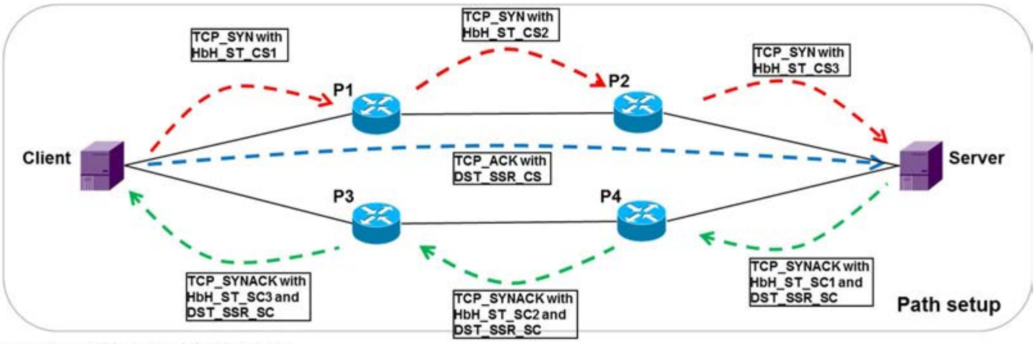

DST_BD_CS:
Dst option with Bandwidth requirement for Client to Server
HbH_ST_SC:
HbH option with Setup for Server to Client
DST_SSR_CS:
Dst option with Setup State Report for Client to Server
HbH_FS_SC:
HbH option with Forwarding State for Server to Client
DST_FSR_CS:
Dst option with Forwarding State Report for Client to Server
TCP_SYN with DST_BD_CS
Client
P1
P2
Server
TCP_SYNACK With HbH_ST_SC3
TCP_SYNACK With HbH_ST_SC2
TCP_SYNACK With HbH_ST_SC1
TCP_ACK with DST_SSR_CS
Path setup
Data with HbH_FS_SC3
Data with HbH_FS_SC2
Data with HbH_FS_SC1
Client
P1
P2
Server
TCP_ACK With DST_FSR_CS
Data forwarding

Figure 4: QoS path setup and data forwarding for the downstream from server to client

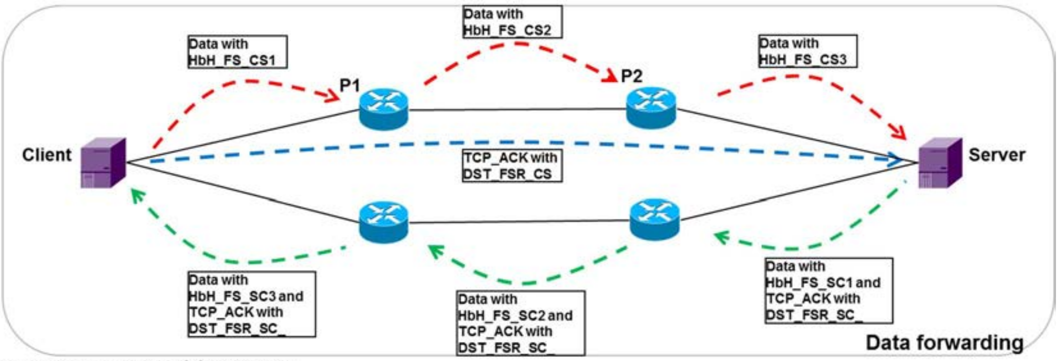
3) **Bi-direction:**

- 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. Figures 5 and 6 show the path setup and data forwarding for bi-direction.


TCP_SYN with HbH_ST_CS1
TCP_SYN with HbH_ST_CS2
TCP_SYN with HbH_ST_CS3
P1
P2
Client
TCP_ACK with DST_SSR_CS
Server
P3
P4
TCP_SYNACK with HbH_ST_SC3 and DST_SSR_SC
TCP_SYNACK with HbH_ST_SC2 and DST_SSR_SC
TCP_SYNACK with HbH_ST_SC1 and DST_SSR_SC
Path setup

Acronym is same as upstream and downstream

Figure 5: QoS path setup for bi-direction between client and server


Data with HbH_FS_CS1
Data with HbH_FS_CS2
Data with HbH_FS_CS3
P1
P2
Client
TCP_ACK with DST_FSR_CS
Server
Data with HbH_FS_SC3 and TCP_ACK with DST_FSR_SC_
Data with HbH_FS_SC2 and TCP_ACK with DST_FSR_SC_
Data with HbH_FS_SC1 and TCP_ACK with DST_FSR_SC_
Data forwarding

Acronym is same as upstream and downstream

Figure 6: QoS data forwarding for bi-direction between client and server

5.5.3.3 Other Control Scenarios

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 an 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.9]. Based on these 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 clause 6.5.

5.5.4 Key Messages and Parameters in Control Protocol

5.5.4.1 Setup State and Setup State Report messages

Setup State is the master message used for following purpose:

- Setup the QoS channel (with optional bandwidth, latency, etc. messages) for a TCP when the TCP session is establishing.
- Dynamic Control of the QoS channel for an established TCP session. See clause 5.5.3.3.

Setup State message should at least include following information:

- Flow Identification (FI) - Notification to hardware about the flow identification methods. See more details in clause 5.5.4.4.
- Report (R) - If the end host needs to report the received setup state to the source address by the Dst-EH.
- Service ID Size (SIS) - The size of Service ID.
- Program (P) - Program or de-program the hardware for QoS.
- Time (T) - The life time of QoS forwarding state in second.
- Hop number (Hop_num) - The total hop number for all configured HbH-EH-aware node on the path set by host. It should be decremented at each hop (HbH-EH-aware node) after the processing.
- Unit of latency (u) - The unit of latency, ms or us.
- Total Latency (Total_latency) - The total end-to-end latency.
- State for each hop programming - Each bit indicates the programming state for a configured HbH-EH-aware node for QoS programming.
- Service ID List for hops - The chained Service ID list for all configured HbH-EH-aware node, each ID is corresponding the Service ID for the hop.

Setup State message should be used with following optional messages to form a complete Setup message:

- Bandwidth.
- Latency.
- Burst.
- Authentication.
- OAM.

Setup messages are 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, some hop can be configured 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 be considered:

- 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 an 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, and 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 should 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. OAM can be used 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.

Setup State Report message should at least include following information:

- Hop number bit (H) - When a host receives a setup message and form a setup report message, it should 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.

- Unit of latency (u) - The unit of latency, ms or us, copied from received Setup message.

- Total Latency (Total_latency) - The total end-to-end latency, copied from received Setup message.

- State for each hop programming - Each bit indicates the programming state for a configured HbH-EH-aware node for QoS programming, copied from received Setup message.

- Service ID List for hops - The chained Service ID list for all configured HbH-EH-aware node, copied from received Setup message.

5.5.4.2 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 an 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; Service ID 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 an 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.

5.5.4.3 Forwarding State and Forwarding State Report messages

Forwarding State and Forwarding State Report messages are used for data plane, See clauses 6.2.1 and 6.2.2.

5.5.4.4 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 (FI=0), address level (FI=3) and transport level (FI=2, 3). Each level of QoS granularity is realized by corresponding in-band signaling. The present document focuses on the flow level in-band signaling.

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 Service ID) generated and maintained in a router. When "Service ID Size" (SIS) is zero, it means the "Flow identification" (FI) is used for both control plane and data plane. When "SIS" is not zero, it means "FI" is only used in signaling, and the data plane will only use the "Service ID".

There are five types for "Flow identification method" defined in the present document:

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.

5) **DSCP:** flow is identified by the DSCP bits. The associated QoS is DiffServ level, and QoS is guaranteed for all IP flows that have the same DSCP bits. For this case, FI=4.

NOTE: The present document only proposes above flow identification method, and it can cover TCP and UDP. But it may not cover all possibilities such as L4 protocols other than TCP and UDP, for example, for all SCTP flows between a pair of source and destination address. This problem can be easily solved by adding new definition for "FI" field.

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 "Service ID Size" (SIS) is 20 bits.

5.5.4.5 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 State message. It is used to locate the bit position for "Setup State" and the "Service ID" in "Service ID List". The value of "Hop_num" should be decremented at each hop (each HbH-EH-aware node). And at the receive host of the in-band signaling, the Hop_num should be zero.

The source host should 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.

5.5.4.6 Service ID, Service ID Size and Service ID List

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

Service ID Size (SIS) is the size for each service ID in the Service ID List. The source host should know Service ID Size, and setup the initial value in the Setup message. The exact Service ID Size can be detected by the OAM message.

When a router receives an 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 should 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 formed as follows: the 1st router's Service ID is at the list header, and the last router's Service ID is at the list tail. The number of the list element is the Hop_number set at the Setup message by the source sender.

5.5.4.7 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, an 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.

5.5.4.8 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 hash value of a key and all invariable part of a signaling message (Setup message: version, FI, R, SIS, P, Time; Bandwidth message, Latency message, Burst message). The variables in Setup message, such as Hop_num, the "state for each hop index" and the "Service ID List" should not be included in hash value calculation.

The key for the hash calculation is critical for the security. It is only known to the hosts and all HbH-EH-aware nodes. The securely distribution of the key is out the scope of the present document.

5.5.5 Security Consideration

The target of security of control plane is how to guarantee the security of IPv6 in-band signaling in such that a faked or unauthorized in-band signaling message will not be accepted and processed. It is realized by following aspects:

1) **Authentication of user:**

   - For any user, if he is interested to use the new service of new transport, he 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 also 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.

2) Proxy:

- Proxy or gateway is the 1st network device connect to customer's devices (Host, phone, etc.) that can generate the IP in-band signaling. The functionality of the Proxy is to check if the in-band signaling is allowed to go through SP's network. This can be done by checking the signaling integrity stated in 3) and other info associated with the user, such as the source/destination IP address, the account balance, the user's privilege, etc.

3) Signaling message authentication:

- The in-band signaling should be checked at each HbH-EH-aware nodes or a proxy node. The details are described in clause 5.5.4.8.

5.6 IPv4 Solution

IPv4 traffic still dominates the current Internet, so, it is important to have solution for IPv4.

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

The present document proposes another disruptive way to do this.

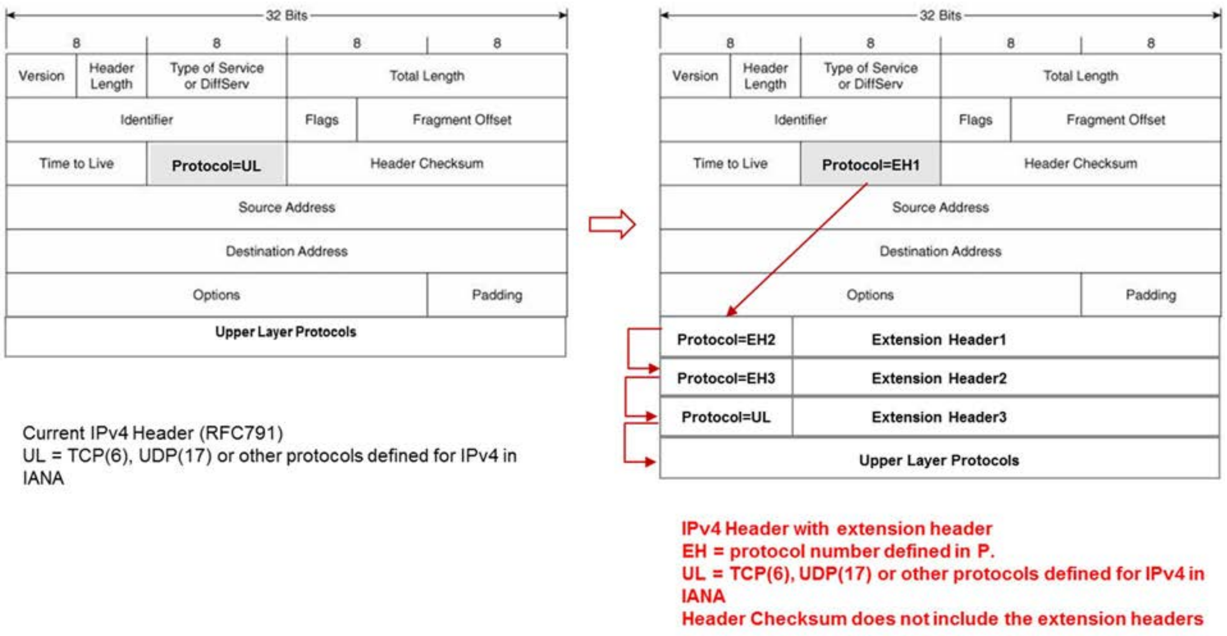
Basically, the present document suggests to extend IPv4 header like IPv6, introducing new extension header for IPv4. Figure 7 illustrates the extended IPv4 header compared with current IPv4 header.

The new protocol number can reuse the protocol number defined for IPv6 as shown in table 1. Currently, those protocol number were defined and used for IPv6 only, they can be used for IPv4 as long as IANA approves.

Off course, the extension of IPv4, the definition of new protocol number and its detailed contents are up to IETF to discuss and consensus.

After IPv4 is extended, the IPv4 in-band signaling scheme and control details for new transport technology will be similar to IPv6. The Hop-by-Hop extension header and Destination extension header (assuming IPv4 has the same extension header name as IPv6) will be used to carry the signaling message.

It is well known that to extend IPv4 header is full of argument in industry and IETF. If IPv6 can completely replace the IPv4 in the near future, then the extension of IPv4 is not required. If IPv4 is still alive for quite long time, then the extension of IPv4 may become more and more necessary.



Current IPv4 Header (RFC791)
UL = TCP(6), UDP(17) or other protocols defined for IPv4 in IANA

IPv4 Header with extension header
EH = protocol number defined in P.
UL = TCP(6), UDP(17) or other protocols defined for IPv4 in IANA
Header Checksum does not include the extension headers

Figure 7: Old IPv4 header and new IPv4 header

Table 1: The protocol number used for IPv4 extension

Protocol number	Keyword	Protocol	Reference
0	HOPOPT	IPv6 Hop-by-hop	IETF RFC 2460 [i.29]
43	IPv6-Route	Routing Header for IPv6	IETF RFC 2460 [i.29]
60	IPv6-Opts	Destination option for IPv6	IETF RFC 2460 [i.29]
135	Mobility Header	Mobility option for IPv6	IETF RFC 6275 [i.30]

6 Network Data Plane Framework

6.1 Basic Capability Requirement

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.

This clause will talk about the basic capability requirement for data plane, and clause 6.2 will discuss the messages used for data plane after the QoS channel is established.

The present 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 should be cooperated with capable hardware to provide the desired fine-grained QoS for different transport.

In our experiment to implement the feature for TCP, a network processor is used 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 hardware such as network processor (NPU) without punting to controller CPU (slow path) for process.

2) The QoS forwarding state is kept and maintained in network processor without the involvement or with minimum interference 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 by in-band signaling.

4) The QoS forwarding does not need to be done at the controller CPU, or so called forwarding at 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.

6.2 Key Messages and Parameters in Data Plane

6.2.1 Forwarding State Message

Forwarding State messages are used for the IP packet forwarding for new transport technology. It is intended to tell each HbH-EH-aware router the necessary programming information to forward the TCP packet properly, and also collect the forwarding state from the router.

Forwarding State message should at least include following information:

- Following fields are defined same as in the Setup State message (see clause 5.5.4.1).
  - Flow Identification (FI).
  - Report (R).
  - Service ID Size (SIS).
  - Program (P).
  - Time (T).
  - Hop number (Hop_num).
  - Unit of latency (u).
  - Total Latency (Total_latency).
- Forwarding State for each hop- Each bit indicates the forwarding state for a configured HbH-EH-aware node for QoS forwarding.
- Service ID List for hops - The chained Service ID list for all configured HbH-EH-aware node, each ID is corresponding the Service ID for the hop.

6.2.2 Forwarding State Report Message

Forwarding State Report messages are used for state notification for the IP packet forwarding for new transport technology. It majorly contains forwarding state information received from the forwarding message of a data packet.

Forwarding State Report message should at least include following information:

- Hop number bit (H) - When a host receives a forwarding state message and form a forwarding state report message, it should check if the Hop_num in forwarding 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 Forwarding State message (source of the data) that if the original Hop_num was correct.
- Unit of latency (u) - The unit of latency, ms or us, copied from received Setup State message.
- Total Latency (Total_latency) - The total end-to-end latency, copied from received Setup State message.
- Forwarding State for each hop - Each bit indicates the forwarding state for a configured HbH-EH-aware node for QoS forwarding, copied from Forwarding State message.

6.3 How a Host sends TCP packet

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:

   a) The hardware was programmed to use the tuples in IP header as identification for QoS process (Service ID Size or SIS=0); and

   b) The packet does not function to collect the QoS forwarding state on the path.

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

   a) The hardware was programmed to use the Service ID as identification for QoS process (SIS != 0).

   b) 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 clause 6.5).

Forwarding message is used to carry the Service ID and also update QoS forwarding state for the hops that are HbH-EH-aware nodes.

After forwarding 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.

6.4 Flow Identification in Packet Forwarding

Flow identification in Packet Forwarding is same as in the QoS channel establishment by Setup State 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 methods 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 matched entry in the QoS table 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 clause 6.5.

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 a matched entry is found in the QoS table for the packet, the packet will be processed by the QoS state entry found in the QoS table.

6.5 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 an 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 clause 7.1 for details.

6.6 Security Consideration

The security of data plane is how to guarantee the security of QoS forwarding hardware in such that a faked or unauthorized data packet will not be forwarded **with** QoS.

NOTE: The security scheme cannot protect a faked data packet forwarded **without** QoS, since this becomes the classical security issue for a current network and is out of the scope of the present document.

There are two scenarios for data plane with regarding to the security of data plane:

1) **Unauthorized user data protection:**

    - Unauthorized user data may exhaust the QoS forwarding resource, but this is a classical security issue. As long as a network owner have enough security configuration in network service, such as DHCP, DNS, Firewall, ACL, etc., it will not allow any unauthorized user to use the network and send data to the network; and there is no unauthorized user data into the network to consume the QoS forwarding resource.

2) **Authorized user but unauthorized user data protection:**

    - This is a situation that a legal user is authorized to use the network, i.e. allowed to connect to the network and assigned a valid IP address, but is not authorized use the QoS transport service. For example, a valid user who does not sign up for QoS transport service or his balance is not enough. This can be protected at the Proxy by classical security scheme such as ACL to check if user data may hit a QoS forwarding entry in Forwarding Information Table, and if hit is the user allowed to use such service by checking user's assigned service and balance, etc.

3) **Service ID to protect Denial of Service (DOS) attack:**

    - Malicious DOS attacking is a situation that hackers use a middle box to sniff the normal data stream, and send simulated garbage data with the same IP tuples as those programmed with QoS service. This attack can exhaust the normal QoS forwarding and lead to normal service for normal flow unavailable. Using Service ID can mitigate this attack.

    - Service ID is originally used for performance improvement of forwarding with QoS, it also provides the additional security protection of forwarding resource for data plane. Service ID in each HbH-EH-aware node is to represent an IP flow with programmed QoS service, it is normally any local significant number generated on 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.

7 Host Congestion Control and Traffic Management

7.1 Introduction

Host end user device, no matter if it is a server, a client or a cell phone, it always has two important functionalities that are related to the network resource. One is the congestion control and another is the traffic management for interface of end-user devices:

1) **Congestion Control:**

   - Congestion control in end user device is the mechanism to control user's application to user network resource based on the common industry standards such as TCP-reno, TCP-cubic, etc. User cannot greedily use network without considering the congestion and fairness.

   - The current congestion control algorithm, such as TCP-reno, TCP-cubic, etc., are all based on the assumption that IP network can only provide best-effort service. If IP network can provide multiple service in addition to the best-effort, the congestion control will be impacted, and the current congestion control algorithm all do not work efficiently with the new service. This clause describes the principals in host for the new congestion control.

   - Similar to the history that there are many variants of TCP congestion control algorithm based on the best-effort IP network, it is expected that there will be many solutions for the new congestion control based on the QoS capable IP network. The present document only describes one basic solution to demonstrate what could be the new congestion control, it does not exclude any other alternatives with regarding to the new solution and features based on the QoS capable IP network. The details of the new congestion control is described in clause 7.3.

2) **Traffic Management:**

   - Some end-user device especially big server has complicated hardware architecture and multiple interface, its traffic management function is to control how the interface resource is shared between different users and different TCP/UDP sessions. The classical mechanism is similar to a network device, all network resource will be shared fairly for all user's data stream including TCP, UDP and other IP protocols. The bandwidth allocation between transport TCP/UDP sessions are the same as a router. When an egress link is congested, the bandwidth will be evenly distributed to all sessions. For example, if a server's egress interface is congested due to n client's downloading job, then each client can only get 1/n bandwidth for the interface.

   - To accommodate the new transport service with QoS, the restriction in traffic management to fairly distribute the network resource to all active sessions should be relaxed, this is the same as discussed in 2) of clause 4.5.2. The server's kernel should 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.

   - Since this is similar to the implementation of network device for new transport technology, the present document will not describe it in details.

   - For the new transport technology, some new traffic management functionalities are needed to process the Setup State Report and Forwarding State Report messages:

     ▪ QoS path setup failure: If a QoS path setup is failed due to QoS programming failure, the TCP source host may take one of following actions depending on the configurations:

       - Notify the application about the failure reason and stop.

       - Notify the application about the failure reason; and reduce the QoS expectation to re-try a new TCP session setup with new QoS parameters. If the re-try is still failed for pre-configured times, stop re-try.

    - Notify the application about the failure reason; and re-try a new TCP setup without any QoS requirement, this is actually to provide application a traditional TCP service. If the re-try is still failed for pre-configured times, stop re-try.

  - QoS forwarding failure: If an established QoS TCP session reports QoS forwarding failure, the TCP source host may take one of following actions depending on the configurations:

    - Notify the application about the failure reason and stop.

    - Notify the application about the failure reason, and re-try to repair the failure (with the same TCP service as before) by sending a new TCP session with old QoS parameters. If the re-try is still failed for pre-configured times, stop re-try.

    - Notify the application about the failure reason, and re-try to repair the failure (with traditional TCP service) by sending a new TCP session without any QoS parameters. If the re-try is still failed for pre-configured times, stop re-try.

7.2 Definition of New IP service

When IP provides the best effort service, host only needs to sends data packet to IP network and the packet will be forwarded to the destination address without any further information or instruction from host. The destination address is encoded into the packet IP header. But if IP network can provide QoS service, more parameters are needed for the new service.

The present document suggests when IP provides better service or QoS, the new parameters for the QoS can be:

- Minimum bandwidth the application expects to be guaranteed, or Committed Information Rate (CIR) in bits per second.

- Maximum bandwidth the application may requires, or Peak Information Rate (PIR) in bits per second.

- Maximum burst the application may send to network, or Burst Size (BRS) in bytes.

- Maximum latency the application expects to experience, or Latency in microsecond. This indicates the system will provide the minimized latency service from each hop to TCP.

When a host wants to communicate to another host with QoS, it can setup TCP session with the one or multiple above parameters of Qos.

7.3 New Congestion Control

7.3.1 Overview

With QoS provided by IP network, is the original congestion control tied with TCP still needed? The answer is yes. Even though the existing congestion control algorithms work with the QoS enabled IP network, they cannot maximize the benefits for transport protocol from IP in terms of link utilization, convergence, etc.

In order to maximize the resource utilization, the network device cannot satisfy each application's QoS expectation, normally it only allocates the resource for CIR, or only guarantee the bandwidth requirement of CIR.

The new congestion control algorithm is about how the host sends traffic to adapt to the new behaviour of network in QoS. Following are major aspects for the new congestion control, the details are stated in the clause 7.3.3:

1) Slow-start is different with before, host can send the traffic at the rate of CIR after the session is established.

2) Congestion avoidance and Sliding window algorithm are still used but with new schemes.

3) More accurate detection can be done by OAM to distinguish the congestion loss, random and long term physical failures. As a result, the associated congestion control will be different with classic one.

4) Fast retransmit and fast recovery are based on the new scheme to detect the congestion, random and long term physical failures.

5) There is no **ssthresh** needed, the lowest rate for user to use is the CIR, and the sender can send the traffic with CIR when it is recovered from long term failure.

7.3.2 Congestion and Physical Failure Detection

The traditional TCP can only detect the congestion by some indirect behaviour or parameters, such as packet loss or RTT changes. It cannot distinguish the packet lost by congestion, random and permanent physical failure such as link or node failure.

New TCP has better way to detect the congestion by using OAM. OAM can provide more accurate detection with regarding the congestion and different physical failures. Below are schemes:

1) Before application's TCP start:

- Couple of key network parameters should be detected by OAM, such as hop number (for HbH-Aware-node), RTT, router's capability, average RTT, etc. The average RTT is used to setup the initial window (see clause 7.3.3).
- This can be done by setup a measuring TCP connection. The measuring TCP connection does not have user data, it is only used to measure the key network parameters. It can be done by system or application:

    a) System (at Sender) setup measuring TCP connection to different important destinations to collect the key network parameters; or

    b) Application (at Sender) setup measuring TCP connection to the destination to collect the key network parameters before setup the normal TCP connection.

2) After TCP is established:

a) Sender periodically or consistently embed application's TCP data packet with OAM to detect:

   i) Current buffer depth.

   ii) If buffer depth exceeding the pre-configured threshold, or RED signal.

   The OAM state are reported to source by receiver. The period of sending OAM is configurable.

b) If sender detects packet loss (3 dup ACK) after receiving a OAM buffer RED report, it is likely caused by congestion; otherwise, it is likely caused by random physical failure.

c) If packet lost due to time out, it is likely caused by permanent physical failures.

7.3.3 New Congestion Control Algorithm

Following are the details of new congestion control algorithm, it includes new slow-start, new sliding window algorithm, new fast retransmit and fast recover:

1) Receiver side:

- keeps AdvertisedWND=MaxRcvBuffer - (LastByteRcvd - LastByteRead), and send to Sender.

2) Sender side:

a) Measure the current or average RTT, and calculate two WNDs corresponding to the MinBandwidth and MaxBandwidth. The average RTT is measured before TCP starts, see clause 7.3.2.

b) MinBandwidthWND=CIR *RTT/MSS;
MaxBandwidthWND=PIR * RTT/MSS;

3) Congestion Avoidance at Sender side:

a) At start: /* new slow-start */

   ▪ EffectiveWND=CongestionWND=IW (Initial Window)=MinBandwidthWND.

- Host sends traffic with EffectiveWND, the initial RTT can be selected from the history or set by administer.

b) When Sender receives ACK,

```
▪ If (ACK is not dup) { /* no congestion and no loss, new sliding window algorithm */
      Update average RTT, MinBandwidthWND and MaxBandwidthWND
      If (CongestionWND == 1) { /* new fast recovery */
         CongestionWND=MinBandwidthWND; /* new fast recover */
      } Else if (CongestionWND <= MaxBandwidthWND) {
         CongestionWND += 1, /* new AIMD */
      } else {
         CongestionWND=MaxBandwidthWND;
      }
      EffectiveWND=min (CongestionWND , AdvertisedWND) - (LastByteSent- LastByteAcked);
      Host sends traffic with EffectiveWND;
  } else if (ACK is dup and dup count >=3) { /* packet lost, new congestion avoidance algorithm*/
      If (Sender detects RED before) { /* packet lost due to congestion */
         CongestionWND=MinBandwidthWND
         EffectiveWND=min (CongestionWND , AdvertisedWND) - (LastByteSent- LastByteAcked);
         Host sends traffic with EffectiveWND;
      } else { /* packet lost due to physical failure */
         If (no timeout) {/* random physical failure */
              No changes to CongestionWND and EffectiveWND;
              Host sends traffic with EffectiveWND.
         } else { /long term physical failure */
              EffectiveWND=CongestionWND=1
              Host sends traffic with EffectiveWND;
         }
      }
  } else /* ACK is dup but dup count <3, this does not indicate the packet loss */
      No changes to CongestionWND and EffectiveWND;
              Host sends traffic with EffectiveWND.
  }
```

8 Other Issues

8.1 Introduction

Above clauses only cover the details for the QoS support of individual TCP session by using the flow level in-band signaling.

Due to the extensive scope of new transport technology by in-band signaling, there are many other associated issues for IP transport control. Below clauses list some of them. The present document only briefs the solution for each issue but does not go to details. The details of each topic can be expressed in other drafts.

8.2 User and Application Driven, APIs

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

1) New or modified APIs for an OS are needed. These APIs are called by application to setup the TCP or UDP connection with QoS support. These may include the socket and associated APIs, such as **socket, open, connect**, etc.

2) The detailed QoS parameters in in-band signaling is set by end user application. New option should be added to APIs stated in 1), the option is a place holder for QoS parameters (Setup, Bandwidth, etc.), Setup State Report and Forwarding State Report messages.

3) 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:

    a) Lower the QoS requirement and re-setup a new QoS channel with new in-band signaling message.

    b) Use the TCP session as traditional transport without any QoS support.

    c) Lookup the service provider for help to locate the problem in network.

8.3 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 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 the present document and many technologies have been discovered or in research.

The 2nd problem can be solved by combining the segment routing (such as SRv6) 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.

8.4 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 behaviour, protocol and rules in the interworking with non-IP network is not the problem the present document will address.

More study and research need to be done, and new draft should be written to solve the problem.

8.5 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 router. This will serve the following purpose:

1) Proxy can check if an 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.

8.6 UDP and Other Protocols

When In-band signaling is used for other transport protocols, such as UDP, QUICK, SCTP, etc., the similar strategy as TCP can be applied. The key is to establish a closed-loop for the transport control.

For protocols that natively have bi-directional control mechanism such as SCTP, they are simple, and only need to add some QoS control functionalities for the protocol. 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 also has to 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, they need to add this mechanism 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.

8.7 Business Model

The business model is very critical for service provider to provide new transport service to customers. Following is high level principals for the business mode:

1) The new transport service can provide better user experience than traditional transport service for all application and users. It should be in compliance with government regulations.

2) The new service can be designed based on the bandwidth guaranteed or latency guaranteed (bandwidth is also needed for latency guaranteed service) in terms of a quantitative values.

3) The new service should be available to all user and all type of applications using different transport protocols including TCP, UDP, etc.

4) For access network, service provider can make decision if a user is qualified to obtain the service by enforcing ACL on the 1st device connecting to user's device for the in-band signaling data in user's IP packet. The accounting is also executed at the same place to collect the statistics and forwarding to the service provider's accounting system like AAA server.

5) For two connected networks belonging to two service providers. Couple of operations should be done:

   a) Two service provider can setup a SLA for the new service. The SLA will define the rules for the new service. For example, how much total bandwidth service one SP is willing to provide to another SP, it can have details like below. Then, each SP can have a total charge to another SP based on agreed pricing for the each items:

      i) The bandwidth service request for sessions is allowed and provided to another service provider. It may have different unit price for different bandwidth service request. For example, for bandwidth service requests, if its requested bandwidth is between value1 to value2, price per request is P1; and if its requested bandwidth is between value2 to value3, price per request is P2.

      ii) The bandwidth service consumption for sessions from another service provider. It may have different unit price for different bandwidth service consumption. For example, for bandwidth service consumption, if its consumption is between value1 to value2, price per byte is P1; and if its consumption is between value2 to value3, price per byte is P2.

      iii) The latency service request for sessions is allowed and provided to another service provider. It may have different unit price for different latency service request. For example, for latency service request, if its requested latency is between value1 to value2, the price per request is P1; and if its requested latency is between value2 to value3, price per request is P2.

      iv) The latency service consumption for sessions from another service provider. It may have different unit price for different latency service consumption. For example, for latency service consumption, if its consumption is between value1 to value2, price per byte is P1; and if its consumption is between value2 to value3, price per byte is P2.

   b) Each service provider enables some important functionalities at its edge router connected to another provider. The functionality may include:

      i) The security checking for all in-band signaling messages and corresponding TCP data received from another provider's network.

      ii) The statistics about all in-band signaling messages and corresponding TCP data received from another provider's network.

      iii) The security checking for all in-band signaling messages and corresponding TCP data sent to another provider's network.

      iv) The statistics about all in-band signaling messages and corresponding TCP data sent to another provider's network.

   c) The accounting information is exchanged between two providers according to pre-agreed time interval, the corresponding charging is also happened between two providers.

6) For multiple connected networks belonging to different providers, each provider will do same thing as 5) with all its connected provider.

8.8 OAM for Other Scenarios

Clauses 5.5.4.2 and 7.3.2 brief the scenario that OAM technique is used to report the network device state to TCP source for new congestion control. In addition to that, OAM technique can also be used in other scenarios to improve the performance of traditional IP transport protocols. For example:

1) **Receiver uses OAM to inform TCP sender of the wireless link's recovery:**

   In WIFI scenario, interference and attenuation are fairly common and usually lead to the long-time loss of feedback for the remote receiver. This behaviour normally results in RTO in TCP sender. To solve this problem, the sender can monitor the state of the wireless link in the data link layer by checking the OAM information that embeds the states of wireless link. The dynamic OAM state is reported to source by TCP receiver. When TCP source receives an OAM state indicating the wireless link is ok, the TCP source can resume the TCP transmission immediately. The new scheme can speed up the TCP fast recovery when physical link state is not steady.

2) **TCP receiver uses OAM to inform TCP sender of random packet loss:**

As analysed by researchers, FCS (Frame Check Sequence) failure in MAC layer is the 1st reason for packet loss. Actually, its happening frequency is orders of magnitude higher than any other kinds of random packet losses. OAM can be used to distinguish this kind of random loss and notify the TCP sender. In particular, when a "FCS check equipment" (e.g. routers) finds that the FCS check fails, it continues to check the checksum of the IP header in the frame. If the IP header checksum check passed, it can deduce that part of this frame is damaged while the IP header in the frame is well-preserved. Then the router can extract the identification information (e.g. IP address etc.) from the intact header, and embed it in an OAM packet. The OAM is then used to inform the TCP sender that the packet indicated by the identification has been randomly dropped. The early notification of such event can reduce the time for TCP sender to detect a packet loss by traditional way (3 dup ACKs).

8.9 Other Types of In-band Signaling

As mentioned in clause 5.3, there are four types of in-band signaling introduced, but the present document only focuses on the flow-level in-band signaling for TCP. Clause 8.6 is also about the flow-level in-band signaling. Other types of in-band signaling are actually coarser in terms of QoS granularity. They are useful for QoS support for aggregated IP flows and thus valuable for further research.

The mechanism of the control plane is expected to be similar to the flow-level in-band signaling, the major difference is that the user TCP session may not be the only method to carry the in-band signaling message. Another difference is that aggregated IP flows may be encapsulated into a tunnel, and the QoS is enforced to the flow identification of the tunnel.

9 Experiment

9.1 Introduction

In order to verify the theory and the proposed design, some basic experiments have been done for POC (Prove of Concept). Exisiting hardware were used with new software and firmware.

In clause 9.2, the basic architecture and components of the test equipment are discussed.

In clause 9.3, the detailed experiment results are given. To demonstrate the effect of new transport technology, TCP sessions using new technology are tested. Bandwidth and latency of both new and traditional TCPs are compared together. The test is basic and intends to prove the concept that each TCP session can obtain bandwidth guaranteed service or minimum latency guaranteed service.

In clause 9.3, the scalability and performance are analysed to see if the expected design targets are reachable by the new transport technology.

9.2 High Level Hardware, Packet Forwarding and QoS

The experiment uses an access router with about 64Gb/s throughput that has proprietary network processor and traffic management (TM) ICs. The following figures (figures 8, 9 and 10) illustrate the box technical information.

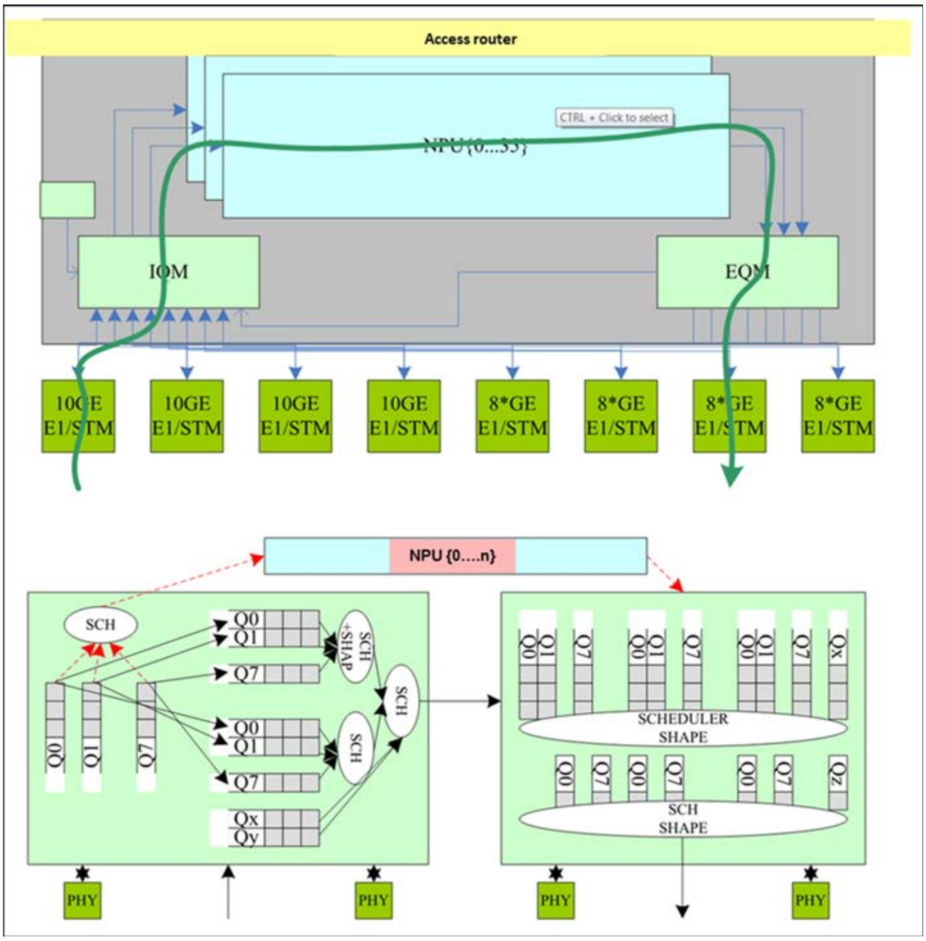


Figure 8: The schematic diagram for the router and its internal component connections between NPU and TM chips

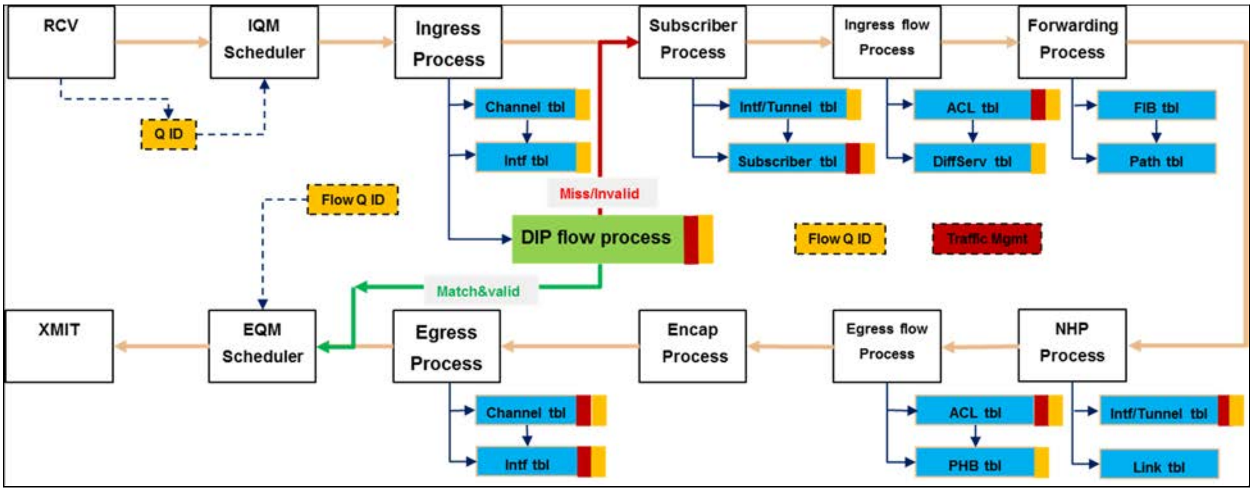


Figure 9: Packet forwarding for different flows, with and without QoS support

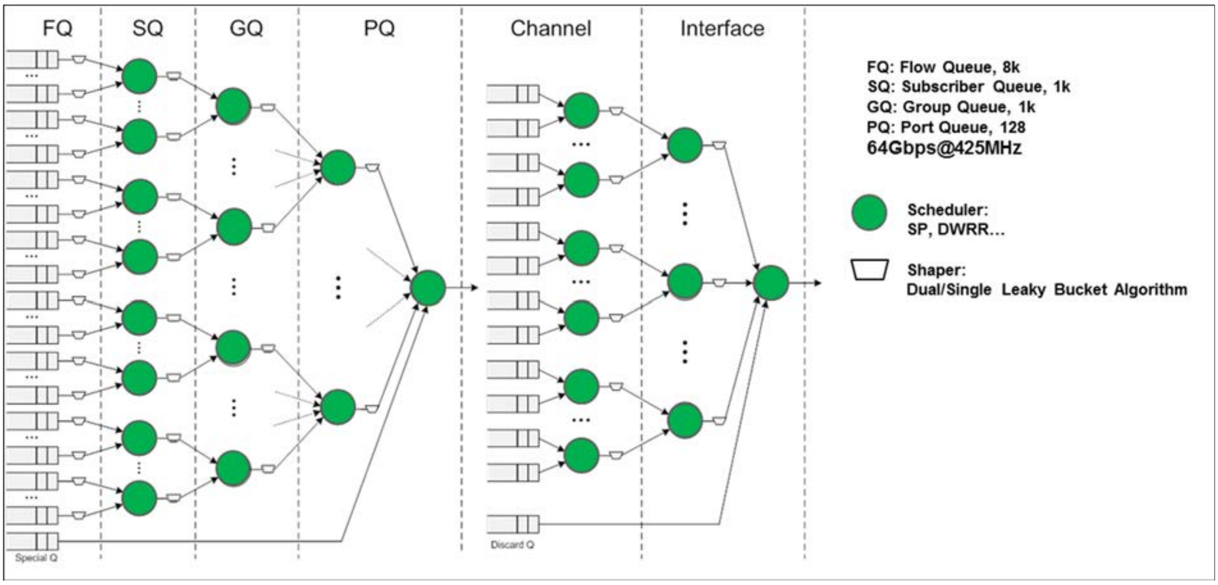


Figure 10: Hierarchy of Queuing and Scheduling in the TM chip

9.3 Experiment Results and Analysis

9.3.1 Test Topology and Configuration

Figure 11 illustrates the test topology, DIP is a new router that can support DIP features.

In order to show the service is guaranteed, a heavily congested network is configured. The DIP ingress traffic rate is at 4G and egress traffic rate is at 100 M.

With traditional transport technology, for all TCP traffic passes through DIP router, each TCP session can only obtain a fraction of bandwidth. It is related to the total number of TCP sessions and the egress bandwidth (100 M).

With new transport technology, new TCP session (DIP flows) could obtain its expected bandwidth or the minimum latency. And most important thing is that the new service is not impacted by the state that router is congested, and this can prove that new service by new transport technology is guaranteed.

The test also demonstrates that the traditional TCP is not impacted, its bandwidth sharing and its long latency symptoms are the same as before. The only difference is the shared bandwidth for traditional TCP is the total link bandwidth reducing the DIP flow expected bandwidth. Moreover, the new and traditional TCP sessions can coexist together. This could prove that design target is achieved.

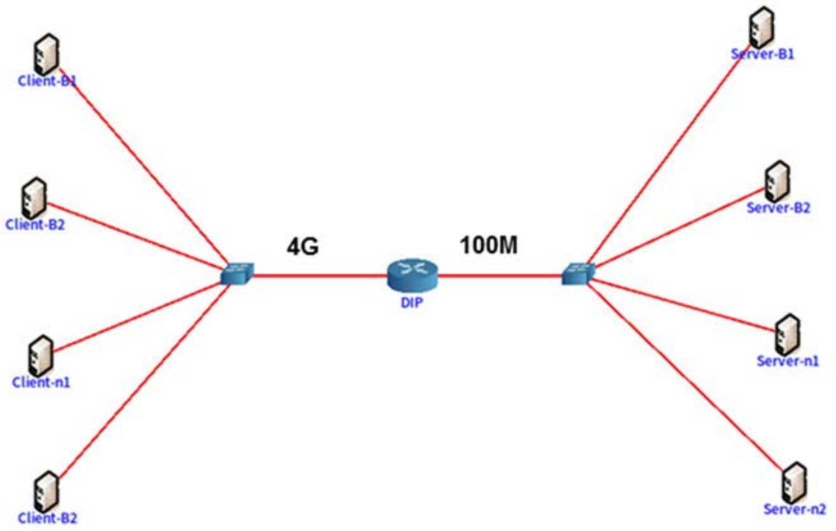


Figure 11: Test topology and configuration

9.3.2 Bandwidth Guaranteed Service

Figures 12 and 13 demonstrate how the bandwidth is allocated to different TCP sessions dynamically. B1 and B2 are new TCP sessions using new transport technology, n1 and n2 are traditional TCP sessions.

Summary of test results:

1) Whenever new TCP sessions start, they obtain the expected bandwidth immediately. Total bandwidth for B1 and B2 are 80 M. The bandwidth allocated to each new TCP session does not change with anything (figure 12 and figure 13).

2) When the traditional TCP starts, it needs time to reach the allocated bandwidth (slow start). The remained bandwidth after allocated to new TCP will be 20 M. This bandwidth will be equally shared between all traditional TCP sessions (figure 12).

3) If network bandwidth is already allocated to traditional TCP, when a new TCP session starts, it will grab bandwidth from traditional TCP and obtain whatever it expected, no matter how many bandwidth is remained (figure 13).

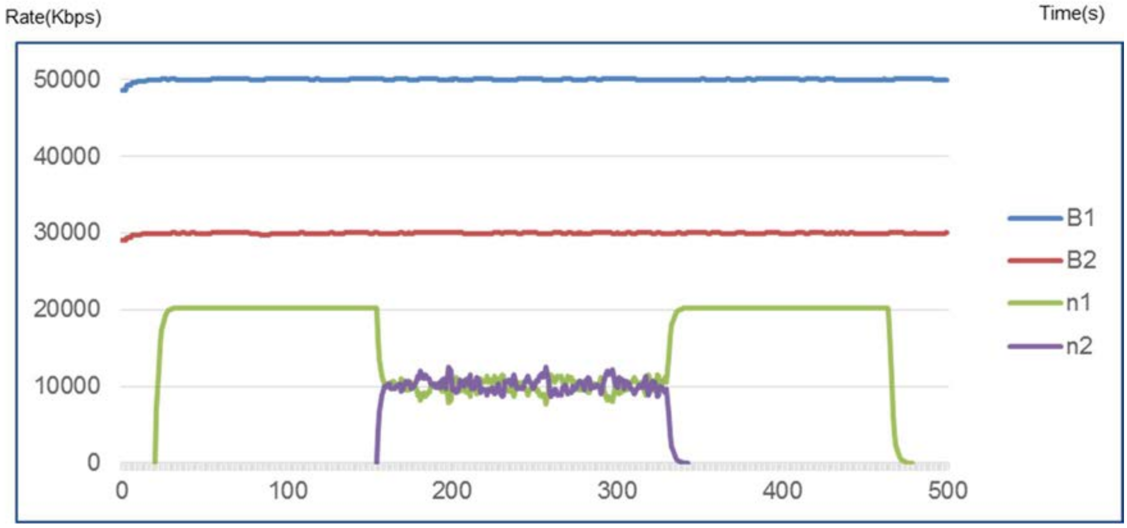


Figure 12: Bandwidth for different TCP sessions, new TCP starts first

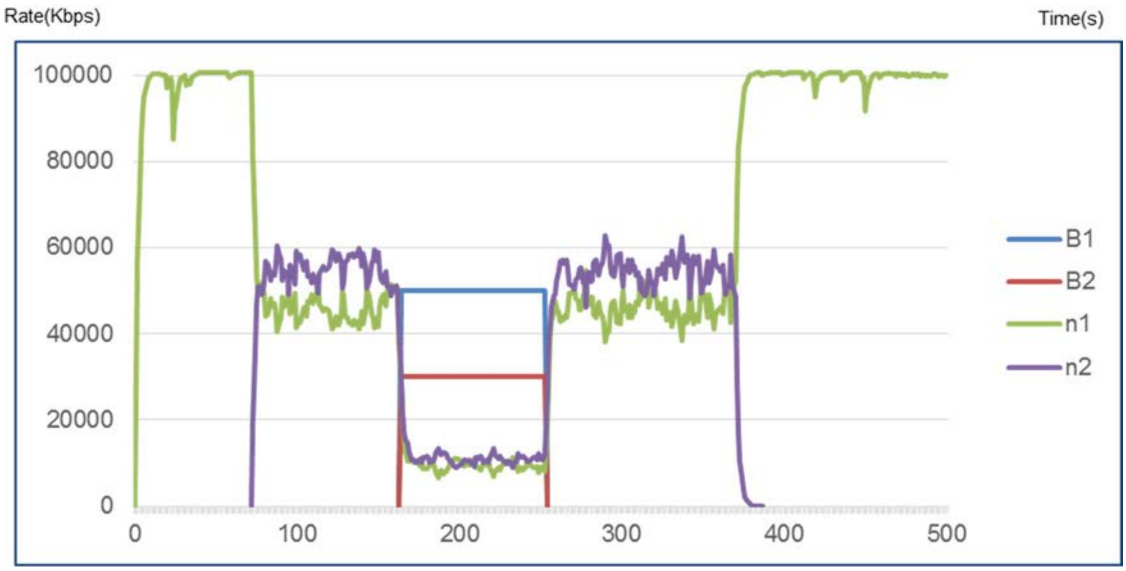


Figure 13: Bandwidth for different TCP sessions, new TCP starts later than traditional TCP

9.3.3 Minimum Latency Guaranteed Service

Figures 14, 15 and 16 demonstrate the CDF (Cumulative Distributed Function) for the latency of different TCP sessions.

Summary of test results:

1) The latency for new TCP session is around 1 mille-second (75 % probability) (figure 14).

2) The latency for traditional TCP session is around 80 mille-second (75 % probability) (figure 15). This is because the router is in congestion and router is configured over buffered to achieve the maximum link utilization.

3) The latency for new TCP session is much less (< 1,25 %) than the traditional TCP sessions (figure 16).

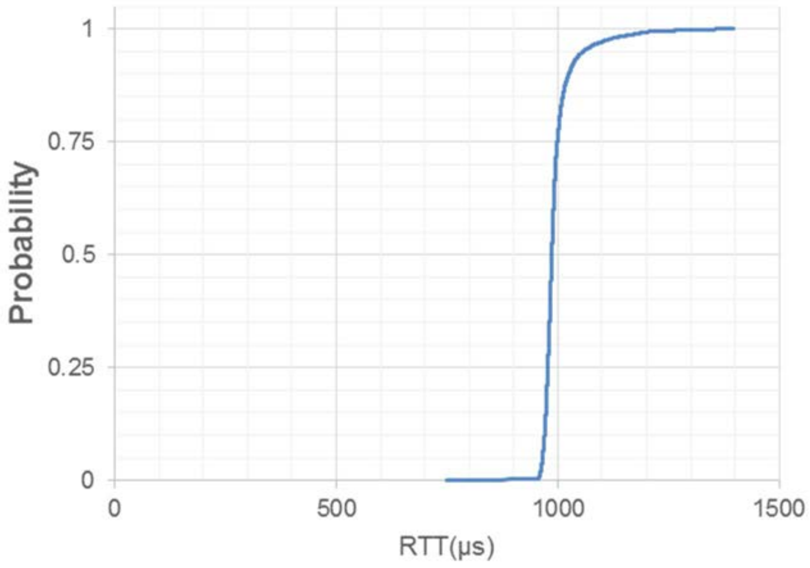


Figure 14: CDF for new TCP sessions

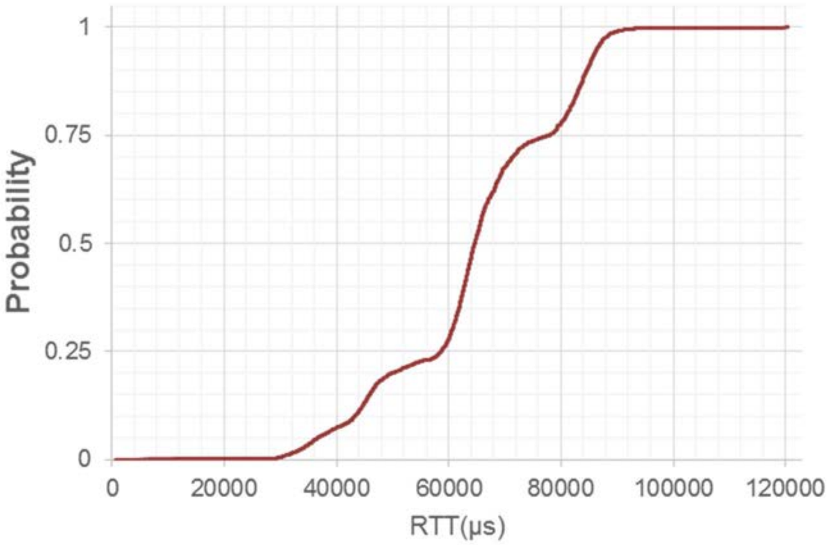


Figure 15: CDF for traditional TCP sessions

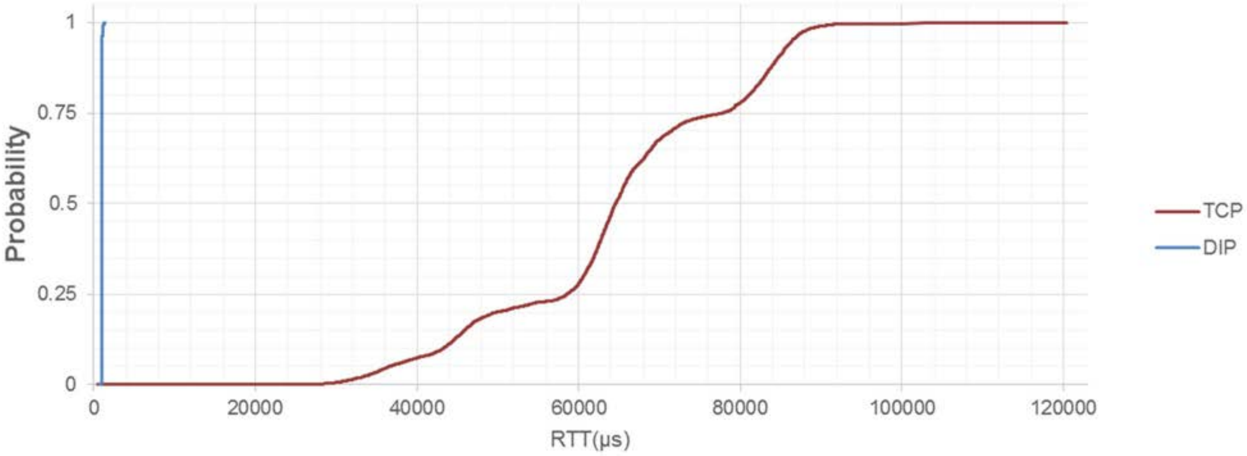


Figure 16: Comparison of CDF for new TCP (DIP) and traditional TCP sessions

9.3.4 Scalability and Performance Analysis

9.3.4.1 Analysis Basics

For the new transport technology, its scalability and performance are key to deployment and acceptance.

The design target for the new transport technology (clause 4.5.3) is considered for the scalability and performance analysis. The new transport technology is designed for the new applications that current transport technology is not able to support, such as ultra-high bandwidth or ultra-low latency scenarios.

In clauses 9.3.4.2 and 9.3.4.3, the scalability and performance will be analysed for different level, port level and system level.

Why the analysis should be analysed for different levels? It is because normally for a modern router, each NPU will support either one or multiple physical ports, the hardware configuration is dependent on the physical port speed. The higher the speed of the port, the less port number one NPU can support. For example, in our experiment, there are multiple ports supported by one NPU. But for the fastest Ethernet port like 400 G, one NPU may only supports one port.

9.3.4.2 Port Level Scalability and Performance

The calculation below will demonstrate if the scalability is acceptable for the tested router and configuration:

- NPU capacity - 72 G.
- Maximum user queue - 1 k.
- Maximum flows 900 - 72/2/900 -> 900 x 40 M/per-flow.

Above calculation indicates for the experiment NPU, it can support up to **900** new TCP session and each TCP session could have 40 M bps bandwidth.

For the latest technology:

- Industry fastest NPU - 400 G.
- 100 M/per-flow; 50 % for new TCP.
- Maximum flows - 200 G/100 M -> 2 000.

Above calculation indicates for the industry fastest NPU, if the half link bandwidth is allowed to new TCP, and assume the TM chip has more than 2 000 queues, it can support up to **2 000** new TCP session and each TCP session could have 100 M bps bandwidth.

Based on the analysis in [i.1], the expected bandwidth for new application such as AR/VR is much higher than 100 M bps, this means the required new TCP session number for each NPU, or target of scalability is much less than above analysis. The current hardware is able to support it.

For performance, it is simpler to analyse. The key of new transport technology is how fast a NPU process the in-band signaling message. Due to the nature of NPU, the processing of signaling message is dependent on the speed of programming of TM chip, it is at microsecond level. Compared with the latency of IP forwarding, it is trivial. Another fact is the NPU performance is almost not degrading with the growth of the number of session in terms of signaling process since the processing is one time job and not as frequent as packet forwarding.

9.3.4.3 System Level Scalability and Performance

One of the key advantages of new transport technology is that it uses hardware accelerated (NPU based) in-band signaling mechanism. This will actually distribute the signaling process into NPU.

Due to the fact that more port or higher throughput for a system, more NPUs are used, and the supported session will automatically increase. This means the system scalability and performance is almost not changing with the growth of the number of transport sessions.

As a comparison, the traditional technology such as integrated service (IntServ) needs to run extra protocol RSVP and the protocol can only be processed by controller CPU. Due to the limit number of controller CPU for a system, the CPU burden will be exponentially increasing with the growth of the number of RSVP sessions, and this results in the bad system scalability and performance for RSVP.

10 Summary

The new transport technology proposed in the present document is based on the QoS enabled IP. QoS enabled IP is achieved by using the hardware accelerated in-band signaling to program hardware for flow-based QoS. It could support emerging application that the current transport technology is not able to support, such as AR/VR applications that need ultra-high bandwidth and/or ultra-low latency. It is transport agnostic and compliance with Net Neutrality, user or application can directly initialize a new TCP session and set the expected QoS parameters. The new transport technology has mechanisms to provide the basic security protection for both control plane and data plane.

There is no fundamental change to how TCP and UDP are used except some enhancements in congestion control.

The current hardware technology is able to support the proposed new transport technology and provide satisfactory scalability and performance.

Table 2 lists key technique characteristics, the purpose and the benefits for the design.

Table 2: The summary of new transport technology

Key techniques	Purpose	Benefits
In-band signaling	• Solve the scalability issues of out-band signaling • Do not use new protocols	• Simpler control plane
Hardware accelerated (NPU based) signaling process	• Distribution of in-band signaling process • Self-maintained state driven by data	• Scalable Control plane with performance
Transport agnostic	• Make it applicable to TCP/UDP, unicast and mcast	• Net neutrality compliance • Simpler upper layer changes • App aware of service level
User driven	• User can initialize the service • User can directly give the expectation	
More fine granularity for QoS	• IP flow-based or aggregated flow-based • Different queuing/scheduling for different QoS matrix combination	• More fine control to specified user flow • Easier to achieve expected QoS • Easier to be adopted by Internet

Annex A: Message Formats

A.1 Setup State Msg

```
  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 0 0|ver| FI  |R|Mis|P| Time  | Hop_num |u|   Total_latency |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     State for each hop index                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Mapping index list for hops                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-. . . +
```

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; 4: DSCP

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 should 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

A.2 Bandwidth Msg

```
 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 0 1|      reserved         |       Minimum bandwidth       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Maximum bandwidth      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type=1

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

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

A.3 Burst Msg

```
 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

A.4 Latency Msg

```
 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

A.5 Authentication Msg

```
 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 0 0|    MAC_ALG    |  res  |  MAC data (variable length)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+. . .+
```

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

A.6 OAM Msg

```
 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 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 to be defined by standard organization

A.7 Forwarding State Msg

```
    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|Mis|P| Time  | Hop_num |u|   Total_latency |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               Forwarding state for each hop index             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Mapping index list for hops                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-. . . +
```

Type=6 Forwarding state

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

A.8 Setup State Report Msg

```
    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 1|ver|H|u|   Total_latency   |          Reserved         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     State for each hop index                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Mapping index list for hops                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-. . . +
```

Type=7 Setup state report

H: Hop number bit. When a host receives a setup message and form a setup report message, it should 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

A.9 Forwarding State Report Msg

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |1 0 0 0|ver|H|u|   Total_latency   |         Reserved          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               Forwarding state for each hop index             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 should 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

Annex B: Standardization

B.1 IANA Considerations

The present document defines a new option type for the Hop-by-Hop Options header and the Destination Options header. According to [i.24], the detailed value are shown in table B.1.

Table B.1: The New Option Type

Hex value	Binary Value			Description	Reference
	act	chg	rest		
0x0	00	0	10 000	In-band Signaling	Annex A

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:** 10 000, assigned by IANA.

The present 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" registry. Initial values for the subtype registry are given below in table B.2.

Table B.2: The In-band Signaling Sub Type

Type	Mnemonic	Description	Reference
0	SETUP STATE	Setup State Msg	A.1
1	BANDWIDTH	Bandwidth Msg	A.2
2	BURST	Burst Msg	A.3
3	LATENCY	Latency Msg	A.4
4	AUTH	Authentication Msg	A.5
5	OAM	OAM Msg	A.6
6	FWD STATE	Forwarding State Msg	A.7
7	SETUP STATE REPORT	Setup State Report Msg	A.8
8	FWD STATE REPORT	Forwarding State Report Msg	A.9

Annex C: Authors & contributors

The following people have contributed to the present document:

Rapporteur:
Lin Han, Huawei Technologies Co., Ltd., lin.han@huawei.com

Other contributors:
Yingzhen Qu, Huawei Technologies Co., Ltd., yingzhen.qu@huawei.com

Lijun Dong, Huawei Technologies Co., Ltd., lijun.dong@huawei.com

Guoping Li, Huawei Technologies Co., Ltd., liguoping@huawei.com

Boyan Tu, Huawei Technologies Co., Ltd., tuboyan@huawei.com

Xuefei Tan, Huawei Technologies Co., Ltd., tanxuefei@huawei.com

Frank Li, Huawei Technologies Co., Ltd., frank.lifeng@huawei.com

Xingwang Zhou, Huawei Technologies Co., Ltd., zhouxingwang@huawei.com

Weiguang Wang, Huawei Technologies Co., Ltd., weiguang.wang@huawei.com

Kevin Smith, Vodafone, kevin.smith@vodafone.com

Annex D:
Change History

Date	Version	Information about changes
11/27/2017	0.0.1	Initial Version
1/20/2018	0.0.2	Add more contents for protocol details
5/30/2018	0.0.3	Add more contents for test results, stable draft
6/22/2018	0.0.4	Add/modify contents for comments, stable draft
7/30/2018	0.0.5	editHelp check, final draft

History

Document history		
V1.1.1	September 2018	Publication