Think of transport technology to support ultra-high bandwidth and/or ultra-low latency

draft-han-6man-in-band-signaling-for-transport-qos

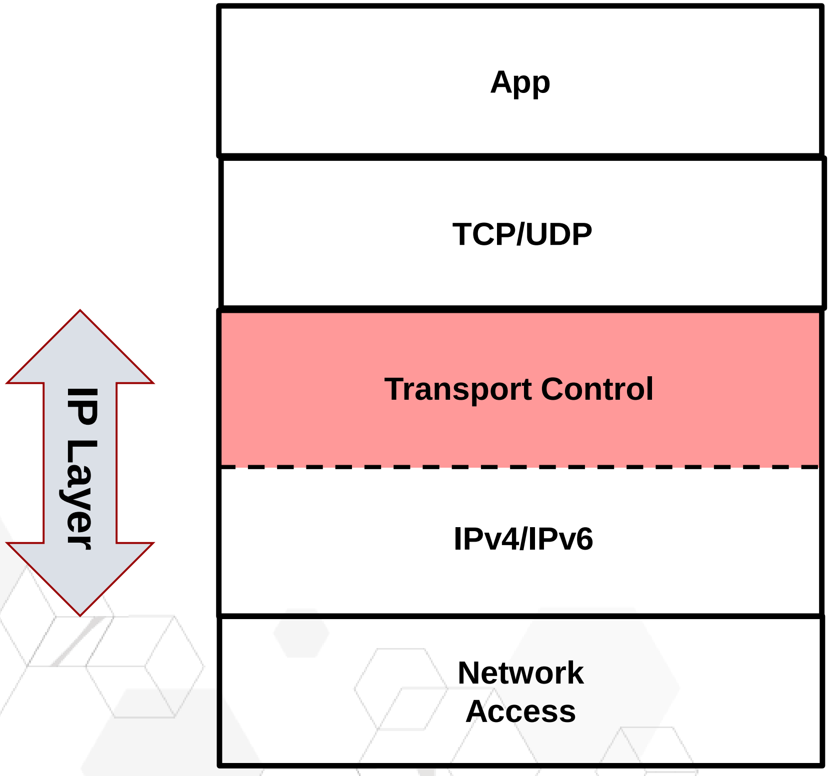
Huawei USA, Future Network Lab: Lin Han

Vodafone: Kevin Smith

Goals of this draft

- **Motivations**
  - Transport service with on-demand QoS
  - For applications that current transport cannot provide satisfactory support
- **Ideas**
  - More involvement of Network device for transport with QoS
  - Granularity: flow(s)
  - Simpler protocols
- **Design Targets**
  - End user or application can directly use the new service
  - The new service can coexist with the current transport service and is backward compatible.
  - Application adaptive QoS
  - The service provider can manage the new service.
  - Performance and scalability targets of new service are practical for vendors to achieve.
  - The new service is transport agnostic. Both TCP, UDP and other transport protocols on top of IP can use it

Transport control sub-layer



Transport Control Functions:

- In-band Signaling by data packet
  - Signaling msg carried in the TCP packet (IP header)
  - QoS HW programming infor for device on path
  - QoS programming state returned to src
- Congestion control
  - Detect congestion state on devices on path
  - Congestion state returned to src
- IP Path OAM
  - Path property detection, static, dynamic
  - Diagnosis

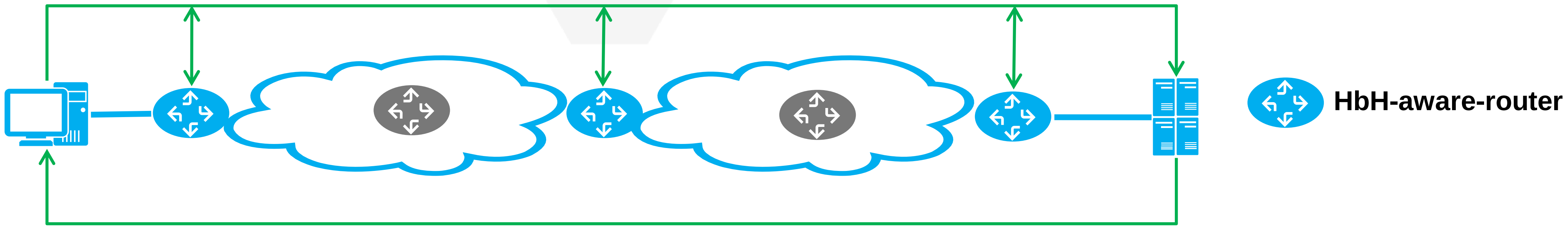
NPU for Signaling processing, QoS forwarding and QoS state keeping

- Network processor (NPU) combines the advantages of ASIC and CPU
  - Tbl Lookup, Packet switch/fwd
  - Packet processing: QoS, ACL, DIP, Firewall, etc
  - General processing: Arithmetic, Hashing, read/write, etc
- State refreshed by data packet
  - QoS state erased if no packet for a configurable time

IPv6 in-band signaling

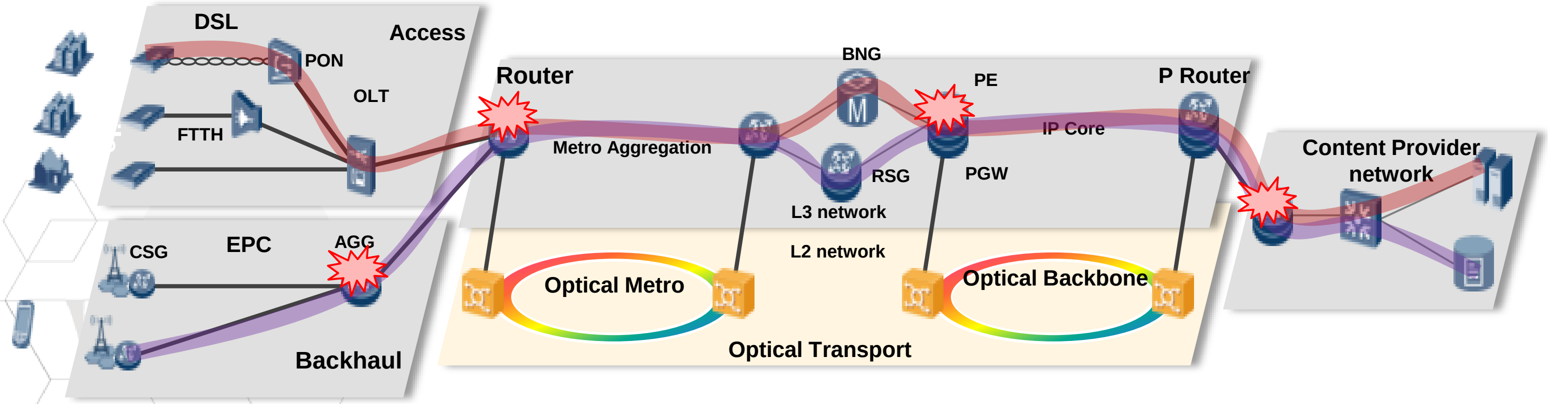
IPv6 hdr	HbH ext hdr	Payload

QoS dir →



IPv6 hdr	Dst ext hdr	Payload

← report



Experimental POC System– Based on the ATN box and NPU SD5131 with TM

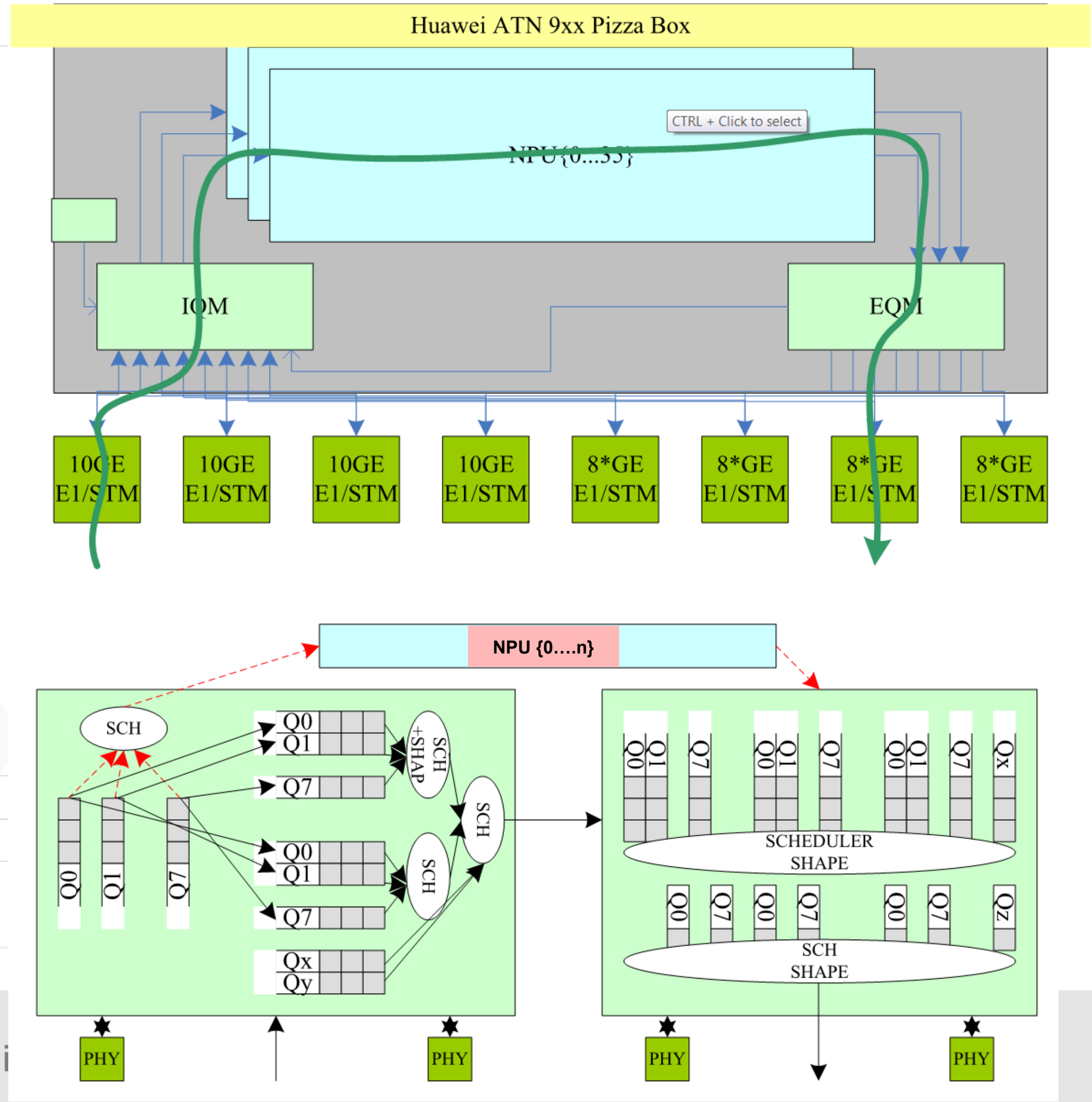
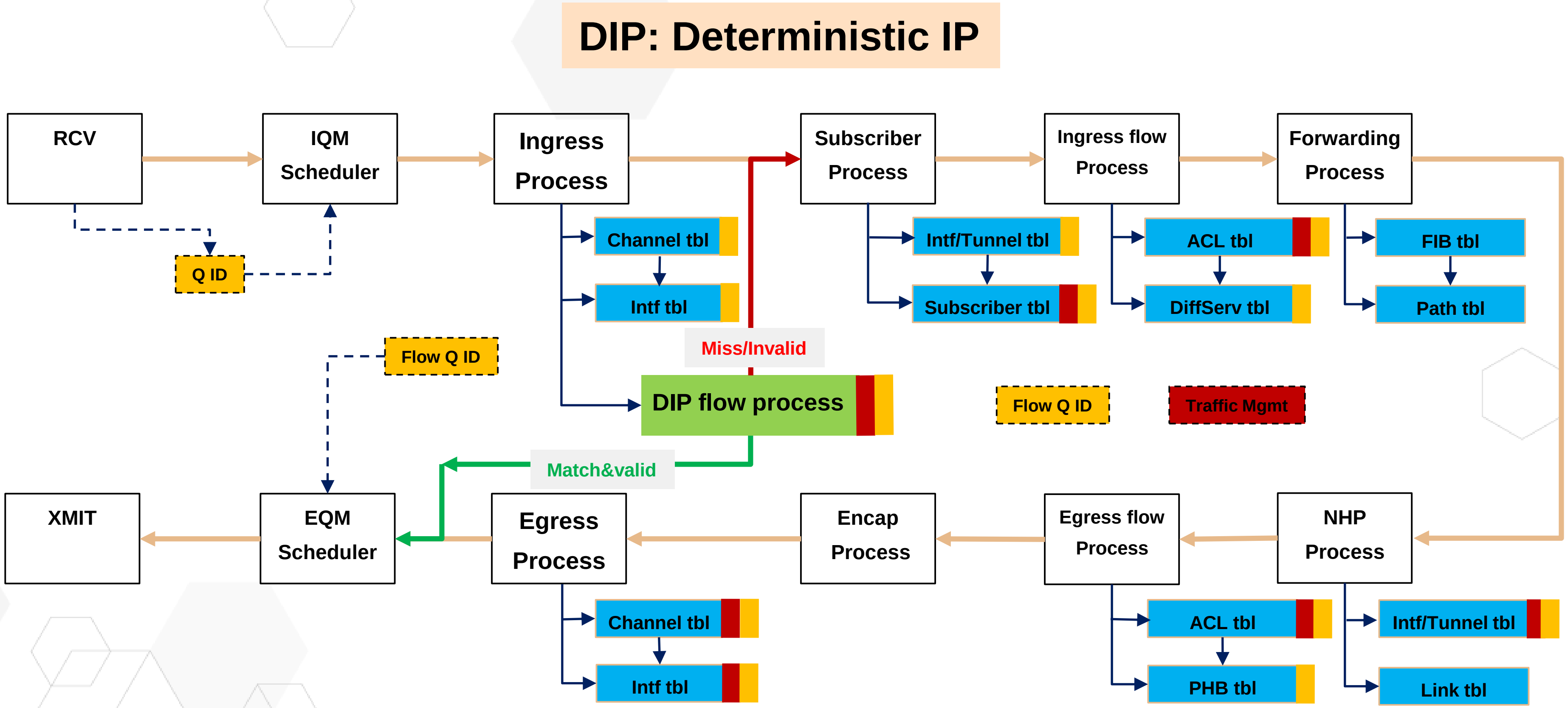
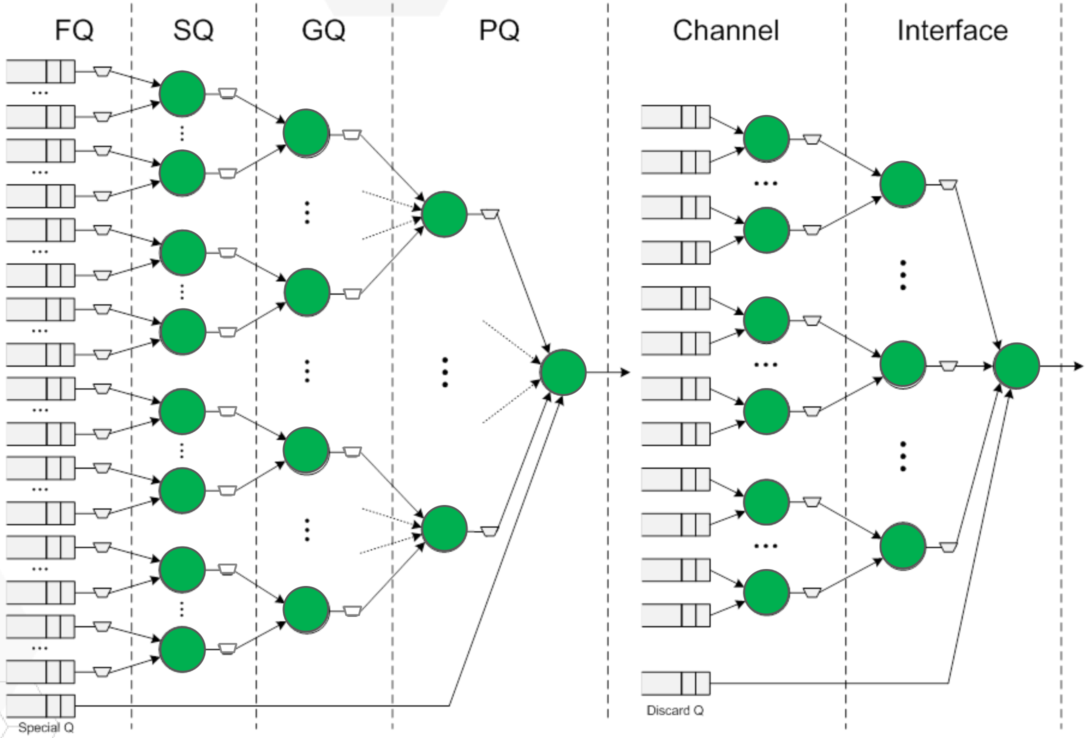


Diagram for Packet Forwarding



Hierarchy of Queuing and Scheduling



FQ: Flow Queue, 8k
SQ: Subscriber Queue, 1k
GQ: Group Queue, 1k
PQ: Port Queue, 128
64Gbps@425MHz

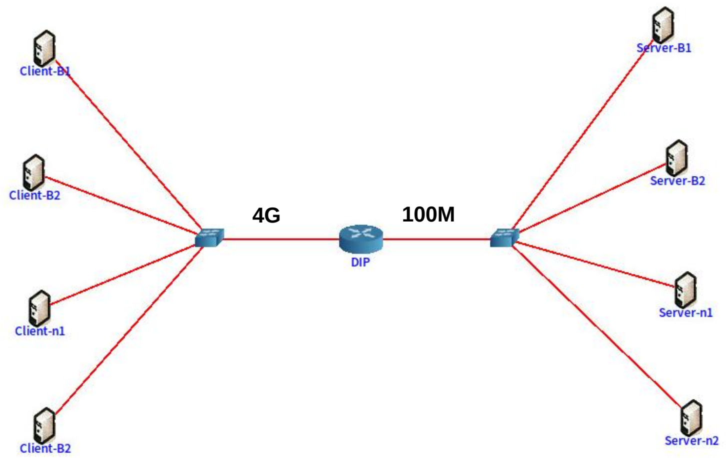
Scheduler:
SP, DWRR…

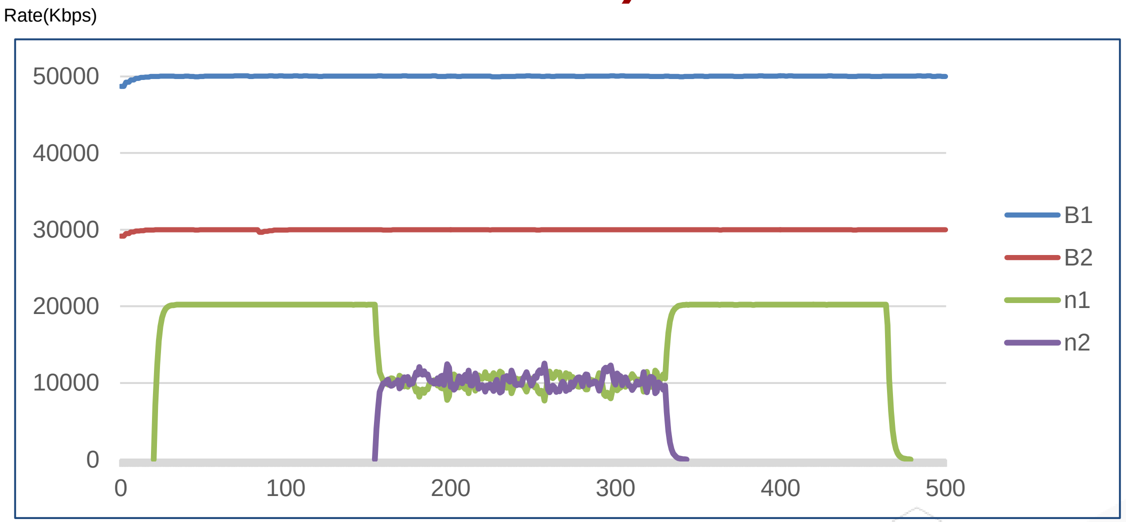
Shaper:
Dual/Single Leaky Bucket Algorithm

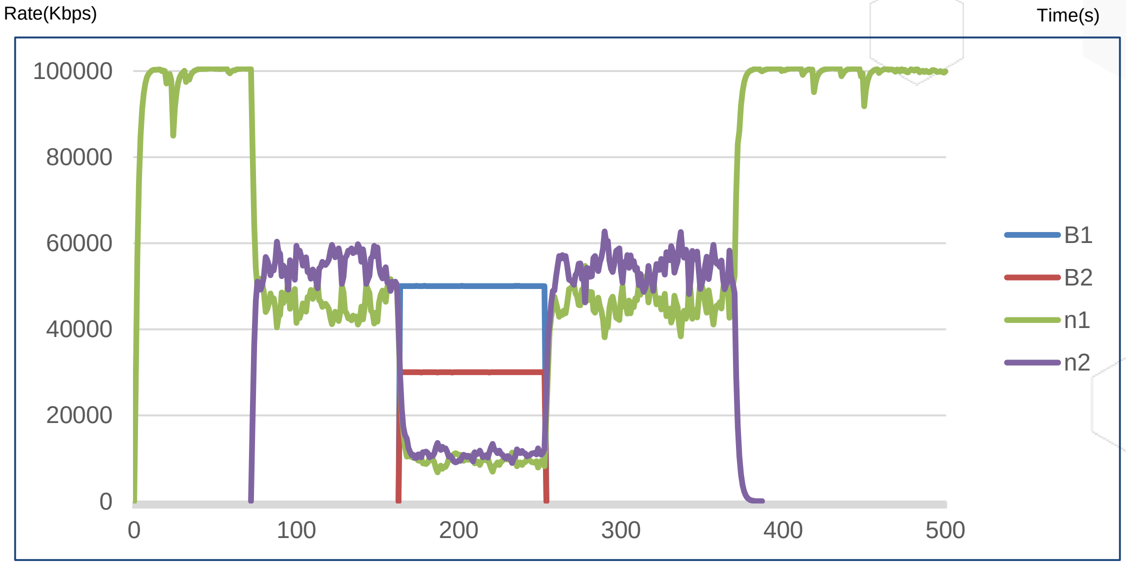
Test Results:

Guaranteed Bandwidth (CIR flows coexistence with traditional TCP)

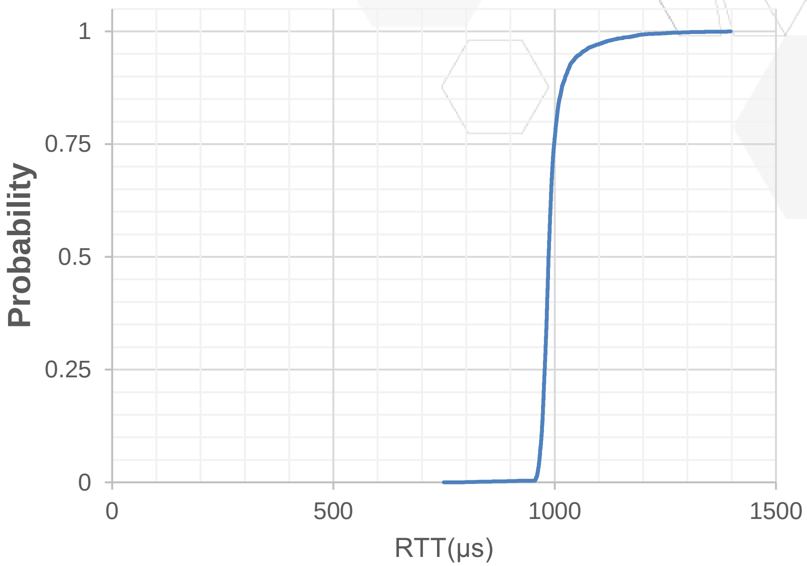
Heavily congested link

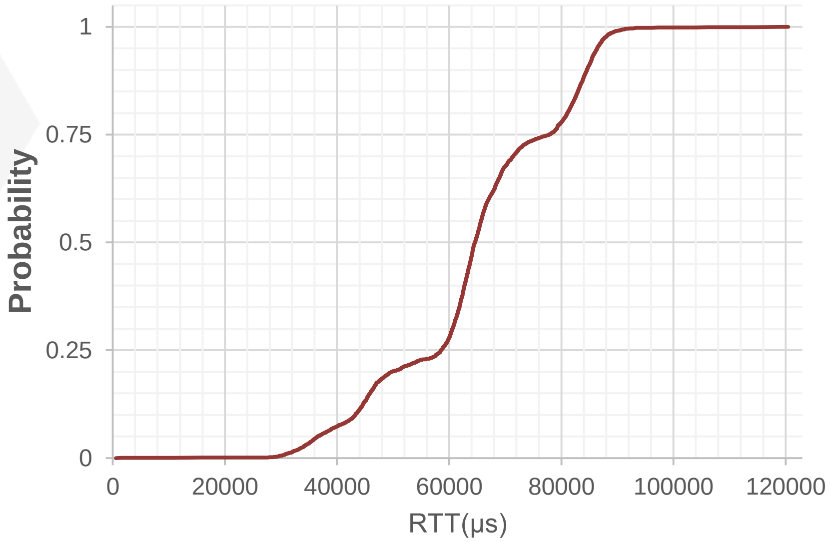


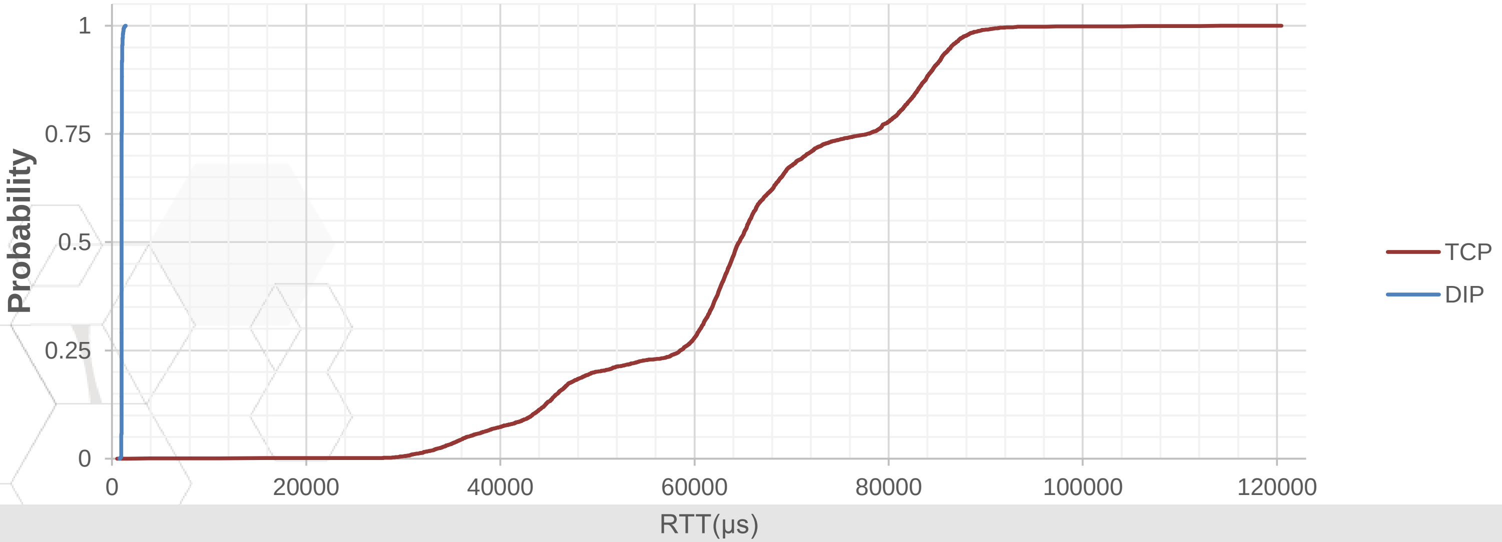




Test Results:
Guaranteed Maximum Latency (CIR flows coexistence with traditional TCP)







Scalability and Performance

- **Scalability**
  - Not targeted for applications that normal TCP works well
  - Assume 100G/NPU; 50% for new session; 100M/session -> 500 sessions
- **Performance**
  - 10ms/hop; 32 hops -> 320 ms/session

Congestion Control

- **What is new CC if network can grantee a certain level of network resource**
  - Guarantee CIR, but not PIR (normal implementation)
  - Many variation and works.
- **The WND is still used, but:**
  - **Receiver** keeps AdvertisedWND = MaxRcvBuffer - (LastByteRcvd - LastByteRead), and send to Sender
  - **Sender** measure the current or average RTT, and calculate two WNDs corresponding to the MinBandwidth and MaxBandwidth
  - MinBandwidthWND = MinBandwidth *RTT ; MaxBandwidthWND = MaxBandwidth * RTT
  - MinWND = min (CongestionWND , AdvertisedWND, MinBandwidthWND);
  - MaxWND = min (CongestionWND , AdvertisedWND, MaxBandwidthWND)
  - EffectiveMinWND = MinWND – (LastByteSent– LastByteAcked); EffectiveMaxWND = MaxWND – (LastByteSent– LastByteAcked);

Congestion Control

- **Source rate control:**
  - No slow start, the initial rate is dependent on the MinBandwdith
  - Set the EffectiveWND from the EffectiveMinWND, increase it like TCP-RENO if there is ACK, and until the WND is equal the EffectiveMaxWND, stop
  - How source send traffic
    - Option 1: Sender control the rate through the EffectiveWND
    - Option 2: Sender control the rate by pacing
- **Congestion control:**
  - Congestion and fwd state detection, Packet loss distinguishing
    - OAM detects Remained bandwidth, buffer depth and buffer RED; OAM and fwd state are reported to source by receiver
    - If packet lost after a OAM buffer RED, it is likely caused by congestion; otherwise, it is likely by random physical failure
    - If packet lost due to time out, it is likely caused by permanent physical failure
  - Congestion and failure action
    - **Congestion loss**: Source reduce the sending data size or rate, EffectiveWND = EffectiveMinWND
    - **Random physical failure loss**: Source keeps the rate
    - **Permanent physical failure loss**: Source reduce WND to 1, resend the in-band signaling to repair path.
    - **FWD failure:** Source reduce WND to 1, resend the in-band signaling to repair path

Guaranteed Bandwidth Service

- **DIP flow and TCP flow shares the same egress queue.**
- **Each DIP flow configured with CIR and PIR**
- **System will guarantee each flow's CIR if** $\sum CIR < C$
- **Two scenario,**
- **Congestion: DIP Congestion (**$\sum R_{DIP}^{Ingress} > \sum CIR$**); TCP Congestion (**$\sum R_{TCP}^{Ingress} > (C - \sum R_{DIP}^{Egress})$**)**
- **No DIP and TCP congestion:**
  - All DIP flow is guaranteed to obtain its CIR, and up to PIR; exceeding PIR will be dropped.
  - All TCP flow obtain equally the rate that is excluding the bandwidth of all DIP flows: $R_{TCP}^{Egress} = (C - \sum R_{DIP}^{Egress})/N_{TCP}$
- **DIP Congestion:**
  - All DIP flow is guaranteed to obtain its CIR, may obtain the rate grater than CIR depending on the remained bandwidth
  - Two options depending on the configuration, when DIP congested ($\sum R_{DIP}^{Ingress} > \sum CIR$) :
    1. DIP flow rate grater than its CIR is proportional to its PIR ratio: $R_{extra_i}^{Egress} = \frac{w_{PIR_i}}{\sum w_{PIR}}(C - \sum CIR) - CIR_i$
    2. DIP flow rate grater than its CIR is equally distributed between DIP flows: $R_{extra_i}^{Egress} = (C - \sum CIR)/N_{DIP} - CIR_i$
  - All TCP flow obtain the rate that is excluding the bandwidth of all DIP flows: $R_{TCP}^{Egress} = (C - \sum R_{DIP}^{Egress})/N_{TCP}$
- **TCP Congestion**
  - TCP flow does not impact the DIP to obtain CIR, may reduce the DIP's rate exceeding CIR.
  - TCP flow rate has the same formula as other cases: $R_{TCP}^{Egress} = (C - \sum R_{DIP}^{Egress})/N_{TCP}$

Guaranteed Latency Service

- **All DIP flows share one or multiple high priority queue; all TCP flow share one low priority queue**
- **Maximum latency at each hop can be calculated from the queue size configured, or from the dynamic queue depth detected through OAM.**
- **No DIP and TCP congestion:**
  - DIP and TCP queue depth will be in very low level, thus very low latency
- **DIP Congestion (**$\sum R_{DIP}^{Ingress} > \sum CIR$**)**
  - DIP flow queue built up, its latency can be calculated by the depth of the queue that can be detected, and the maximum latency is determined by the DIP queue size
- **TCP Congestion (**$\sum R_{TCP}^{Ingress} > (C - \sum R_{DIP}^{Egress})$**)**
  - TCP flow does not impact the DIP queue, thus does not impact DIP latency
  - TCP latency can be calculated by the depth of the queue that can be detected, and the maximum latency is determined by the TCP queue size

Q&A

More detailed works in
ETSI NGP (Next Generation Protocol):
https://portal.etsi.org/tb.aspx?tbid=844&SubTB=844