QoS enhancements for UHR-follow-up

Date: 2024-03-09

Authors:

Name	Affiliations	**Address**	**Phone**	**email**
Dibakar Das	Intel			
Dmitry Akhmetov				
Laurent Cariou				
Dave Cavalcanti				
Necati Canpolat				
Robert Stacey				

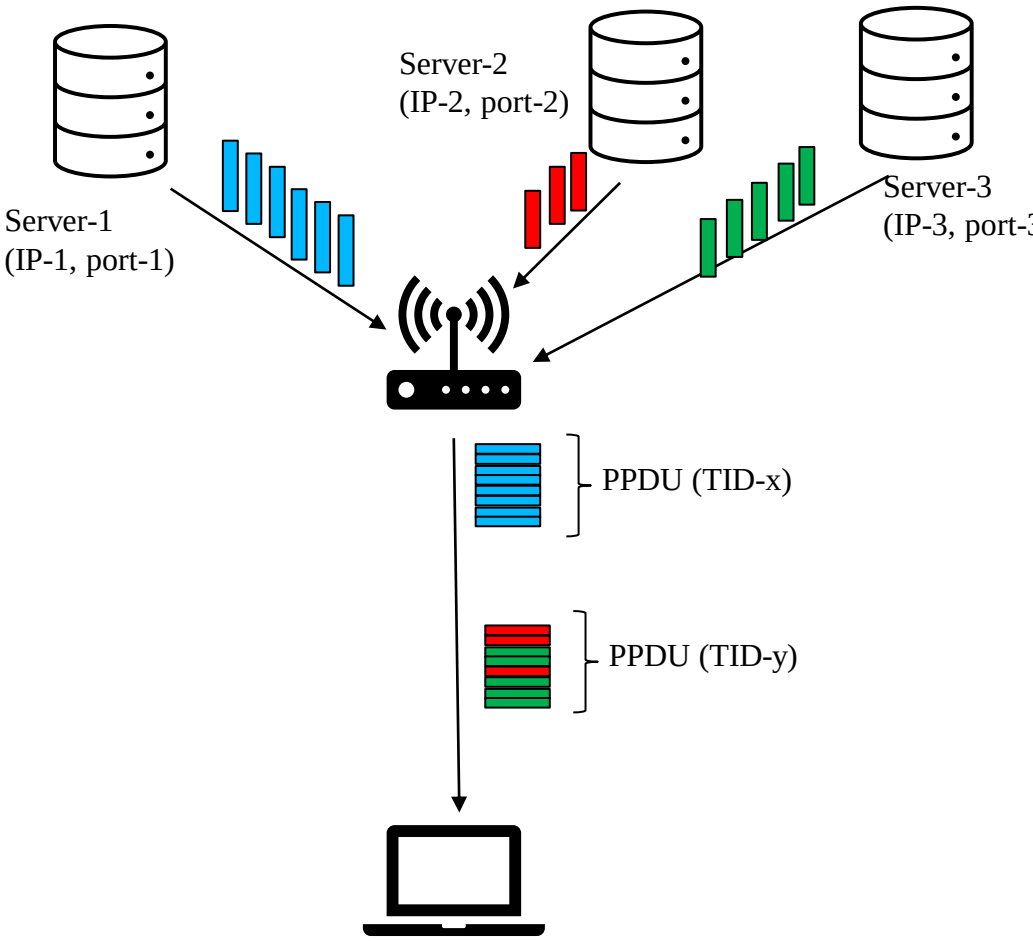
Introduction

- **In 11be, some members highlighted the scalability problem with SCS since multiple flows are mapped to same TID (e.g., 11-20-1686) leading to possible HOL blocking problem at layer-2.**
- **To recap, the HOL blocking problem at layer-2 arises because an older packet waiting retransmission and mapped to a given TID would prevent newer packets that have been correctly received at layer-2 to be forwarded up.**
  - This is not an issue where the newer packet belongs to same traffic flow AND at the application/transport layer the processing of the newer packet has dependence on that of the older packet (i.e., expect in-order delivery).
  - However, for cases when that's not true (see next slide), HOL blocking may affect user-experience by preventing chances of faster processing.
- **Previously in 11-23-0697-uhr we proposed a solution to HOL blocking problem .**
- **In this document we provide further details on motivation and evaluations.**

Use-cases overview

- **There are situations where different traffic streams could be mapped to same TID between two STAs such as multiple voice/ video sessions, time-sensitive flows, file transfer instances etc.**

- **In some cases, there maybe additional information (e.g., IP tuple, Port #, metadata) available at layer-2 to identify the different streams.**



Use-cases overview (contd.)

- **In other cases layer-2 may not have enough information to demultiplex streams because its encapsulated at higher layer (e.g., QUIC connection consisting of multiple streams).**
- **In addition, there are situations where partial delivery of packets belonging to same flow is also useful. For example,**
  - when application layer FEC [1] is applied in which case the receiver may make use of whatever packets are received.
  - An older packet that arrived after a deadline and the newer frames are not dependent on the old packet.
  - Potential packet loss concealment techniques in webrtc that uses AI to fill in missing packets [3]

Demultiplexing streams at layer-2

- For the case when the transmitter STA (e.g., an AP) has enough information to demultiplex streams and each of those streams require in-order (“IO”) delivery (e.g., TCP), the HOL blocking can be resolved if we can put them on to different TIDs.
- Clearly, this is a problem if the number of such streams exceed the number of TIDs that can today be mapped to an AC.
- Proposal: the UHR spec can allow more than 2 TIDs to be mapped to an AC when needed.

Mapping multiple TIDs to an AC

- **Today typically TID = UP and UPs are restricted from 0-7. Also, there is explicit mapping from those 8 UPs to 4 ACs.**
- **Option 1: keep current UP to AC mapping but define mapping from TIDs 8-14 to an AC.**
  - Only subset of TIDs 8-14 may be mapped and only when there is a flow that needs it (e.g., during an active SCS stream).
- **Option 2: allow a temporarily unused TID between 0-7 to be mapped to a different AC than that's derived from default mapping.**
  - Could be dynamic as well (i.e., only during an active SCS stream).

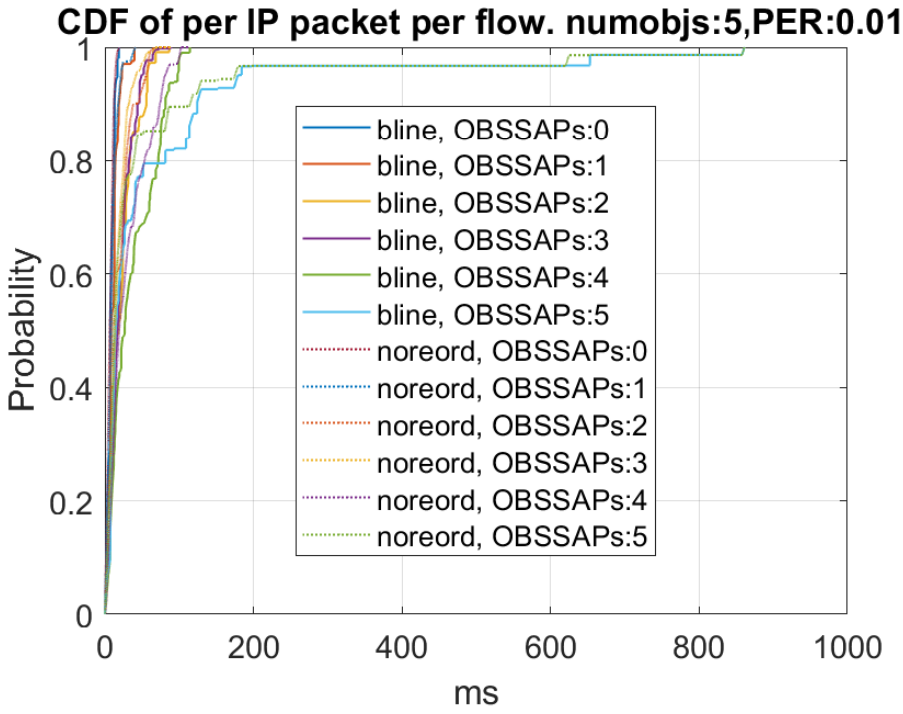
Allowing out-of-order delivery (recap from 11-23-0697)

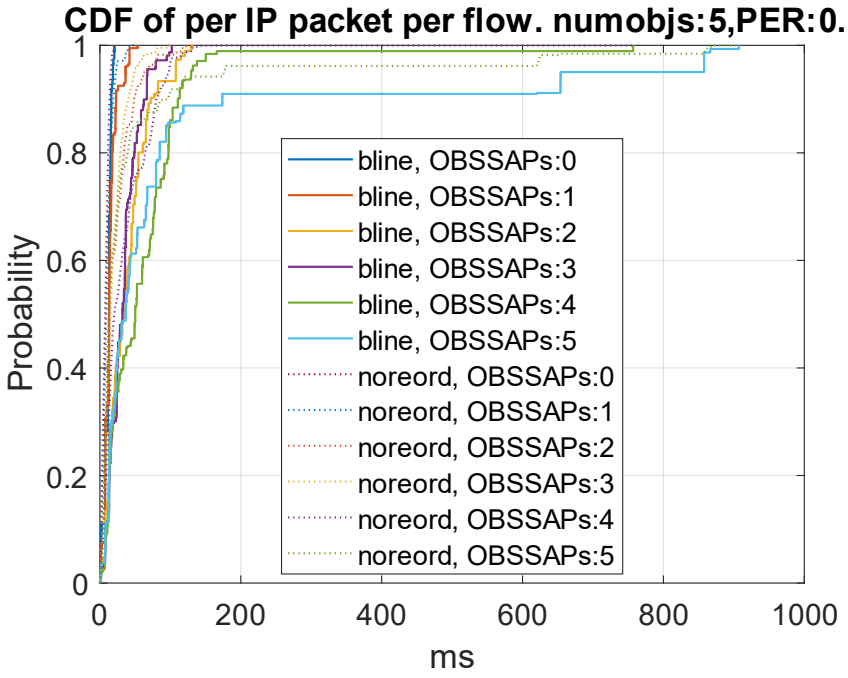
- **For the cases when demultiplexing streams at layer-2 is not feasible but possible at upper layer (e.g., QUIC, RTP etc.) and when the upper layer can handle some packet reordering, we propose to allow the MAC-SAP to deliver out-of-order ("OOO") packets optionally for a subset of TIDs.**
  - Maybe limit to 1 or 2 TIDs. Also, may use only when a flow that can benefit from OOO delivery is established (i.e., following a corresponding SCS negotiation).
- **This is similar to out-of-order delivery by PDCP feature in 3GPP.**
- **Frame replay detection:**
  - Today the in-order delivery sequence allows for easy frame replay detection (i.e., just check if the PN number is greater than last highest PN number).
  - For out-of-order delivery, the frame replay detection needs to change (e.g., by maintaining a sliding scoreboard of PNs).
  - Need a way to separate the PN space used by frames requiring OOO delivery vs ones that require IO.

Evaluations (Scenario 1)

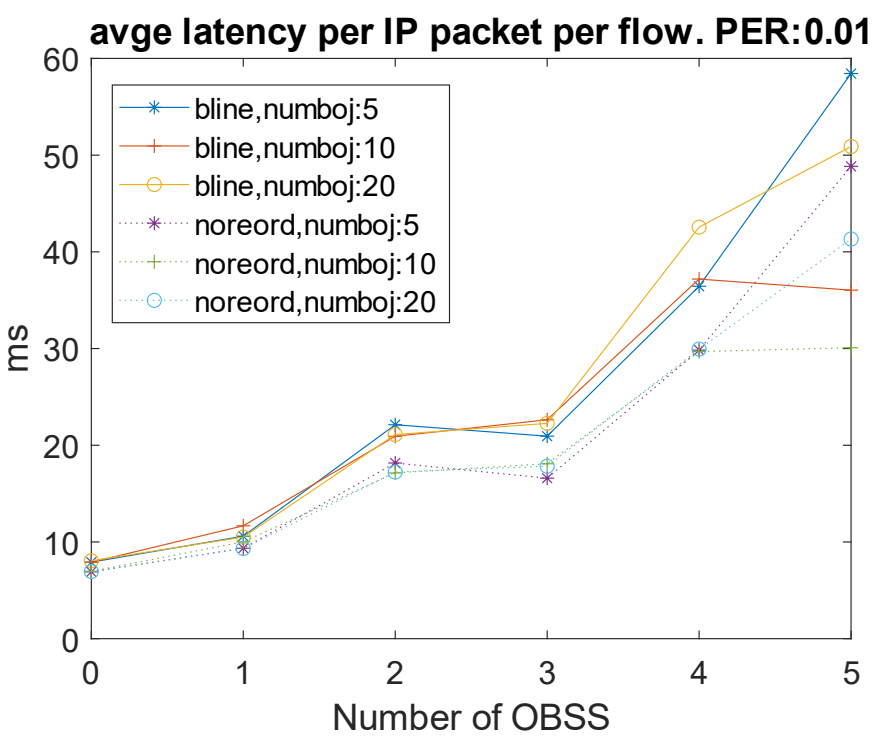
- **Network:**
  - 1 DUT (i.e., wifi-8) AP with one DUT STA and DL flow. Single link, BW = 40 MHz, MCS-5, NSS-2.
  - N OBSS APs (where N = 0, 1,…5).
  - TXOP at DUT and OBSS AP = 6 ms.
- **Traffic and KPI:**
  - At DUT, model very simplistic HTTP response traffic
    - generate N objects of size M bytes one time where set of (N, M) = {(5, 80K), (10,40K), (20,20K)}.
    - Interleaved packets modeling network jitter/delay.
    - Measure latency for each IP packet of an object as well overall latency for each object.
      - For example, this could model progressively loading different parts of a webpage.
  - OBSS traffic is full buffer.
- **Link level PER: (1%, 10%).**

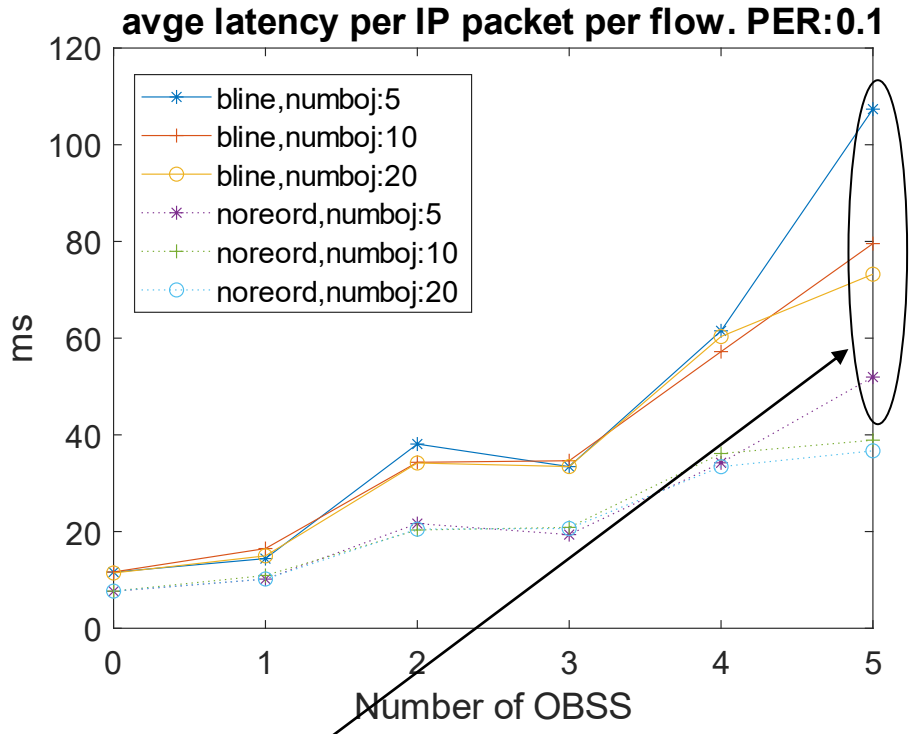
Results (scenario 1)





Results (scenario 1 contd.)



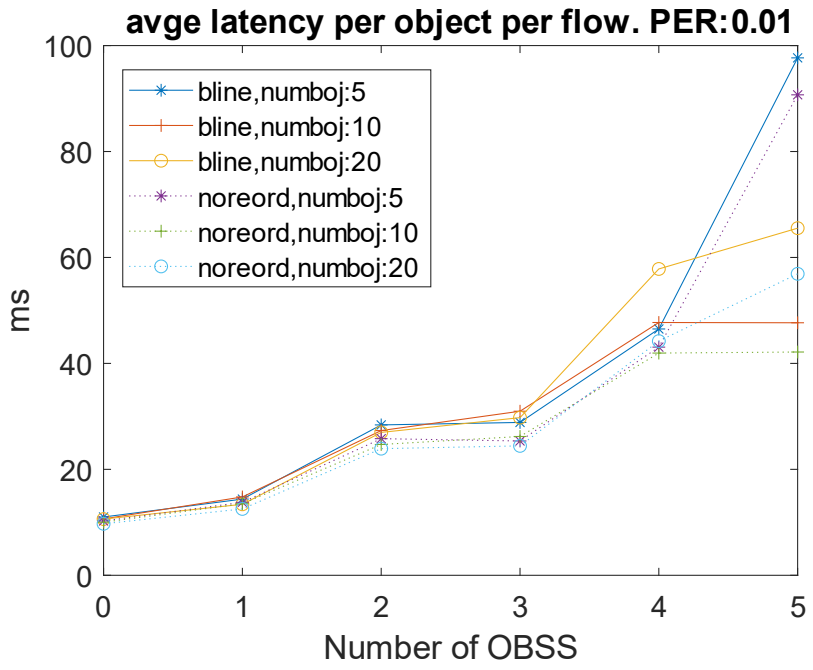


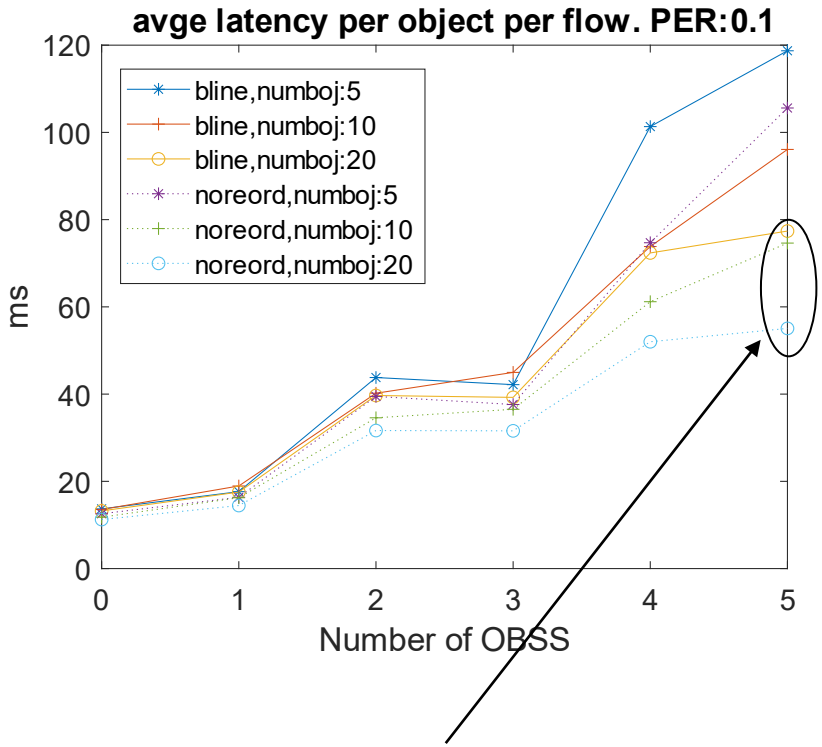
Clearly, higher OBSS congestion => more channel access delay for DUT AP.

For 1% and 10% PER cases, OOO delivery (labelled as “noreord”) clearly provides significant latency reduction per IP packet allowing parallel, partial processing.

For example, from > 100ms with PER = 10% to ~50ms with 5 objects.

Results (scenario 1 contd.)



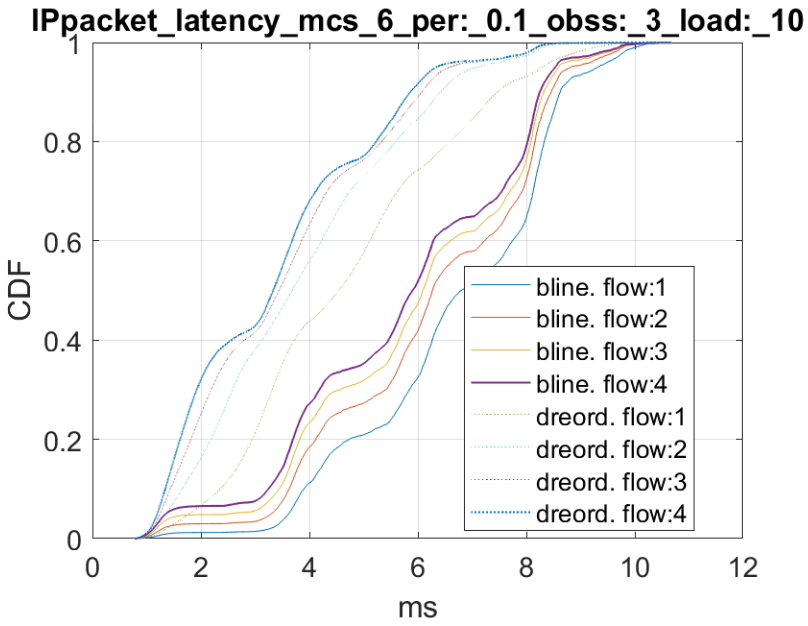


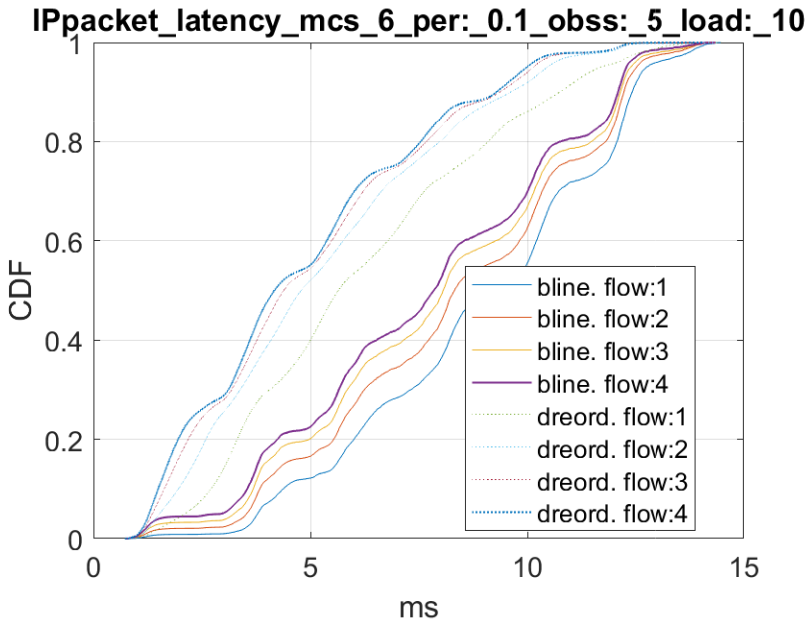
Similarly, overall latency for the whole object also improves significantly (e.g., from 77ms to 55ms when number of OBSS STAs is 5, number of objects is 20 and PER is 10%).

Evaluations (Scenario 2)

- **Network:**
  - 1 DUT AP with one DUT STA and DL flow. Single link, BW = 80 MHz, MCS-7 or 6, NSS-2.
  - N OBSS APs (where N = 3, 5).
  - TXOP at DUT AP = 3 ms.
- **Traffic and KPI:**
  - At DUT, 4 flows modelling 4 point cloud objects [2]
    - Flow-1 to 4 has data rates of 53.5, 21.95, 13.63 and 10.0 Mbps respectively with corresponding video frames generated every 16ms and split into 1400B IP packets.
    - Interleaved packets modeling network jitter/delay.
    - Measure latency for each IP packet of a frame of an object as well overall latency for each frame.
  - Each OBSS AP occupy X % of airtime (where X = 10, 20).
- **Link level PER: (1%, 10%).**

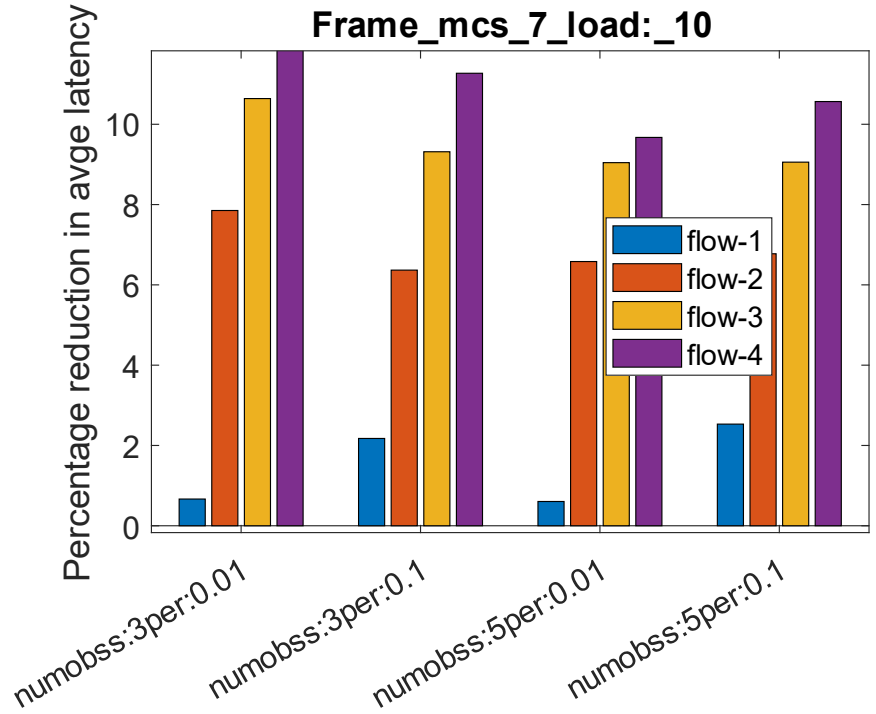
Results (scenario 2 with MCS-6)

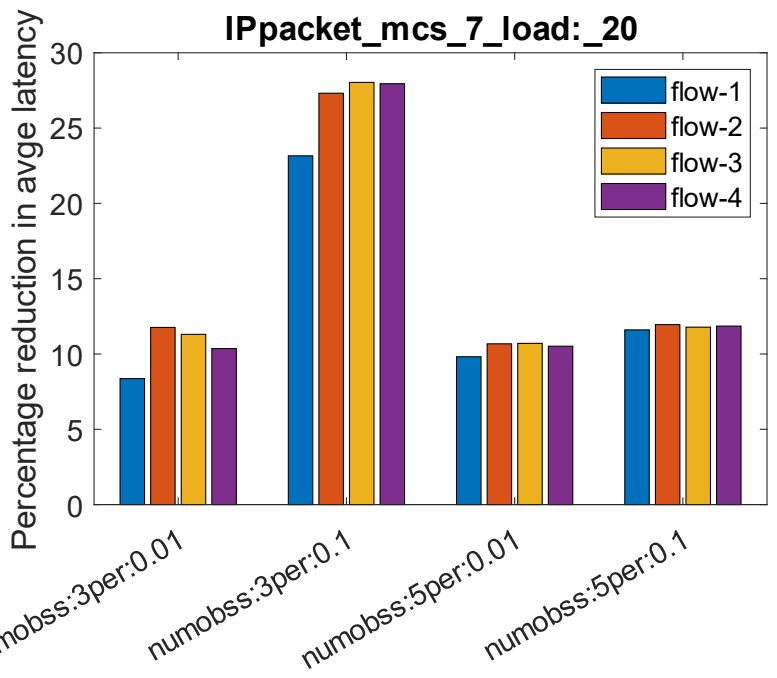




Clear gains until the network below saturation, then the DUT has difficulty delivering all traffic.

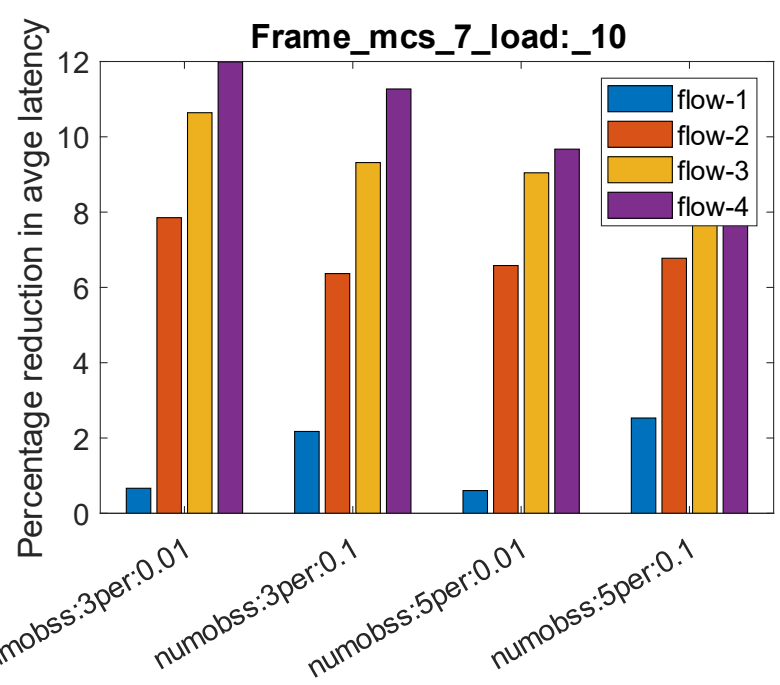
Results scenario 2
average latency per IP packet per frame

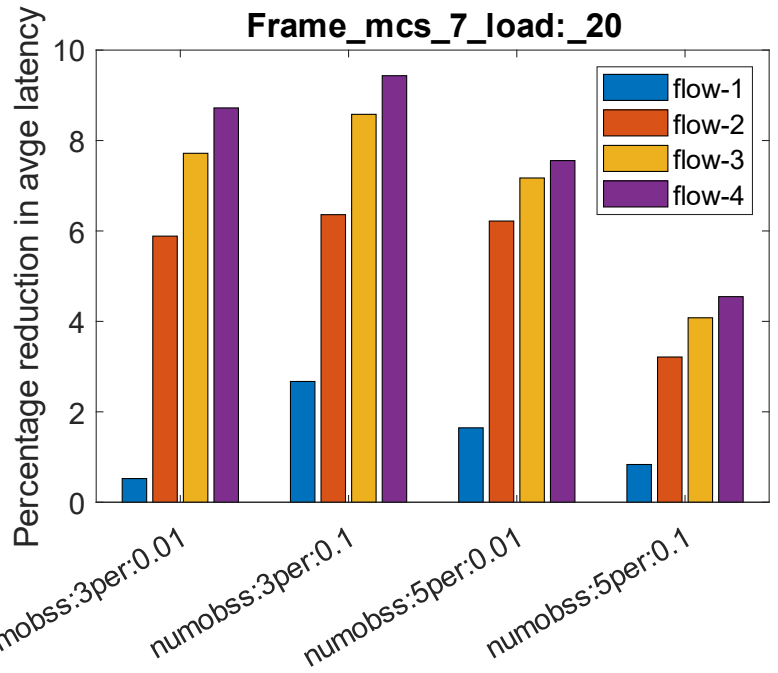




- Significant reduction in latency (e.g., about 20-30% reduction in latency with number of OBSS =3, PER = 10%).
- Low data rate flows (e.g., flow-4) benefits more because they are more likely to be blocked by a packet from a high data rate flow.

Results scenario 2
average latency per frame





- Also, significant reduction in overall frame latency esp. for low data rate flows (~ 10%) .

Possible high-level design to enable OOO delivery

- **Limit to changes only at driver/software level. No change in low level MAC design including BA score-boarding.**
- **No mixing of IO and OOO delivery in same TID at any time.**
- **Separate key generated during 4-way handshake used to encrypt TIDs with OOO traffic => PN space for OOO frames are different from that of IO frames.**
- **No change in reception for TIDs that contain flows with IO traffic.**
- **The PN window should be equal to size of the maximum number of MPDUs for that TID that can be aggregated in a single PPDU i.e., BA scoreboard size for that TID.**
  - Otherwise, we will have scenarios where a retransmitted packet wont be forwarded up for failing the replay detect.
  - Also, transmitter wont send frames beyond the scoreboard length.

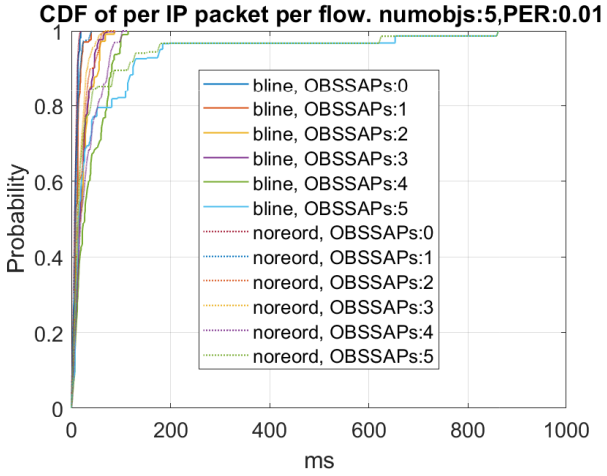
Summary

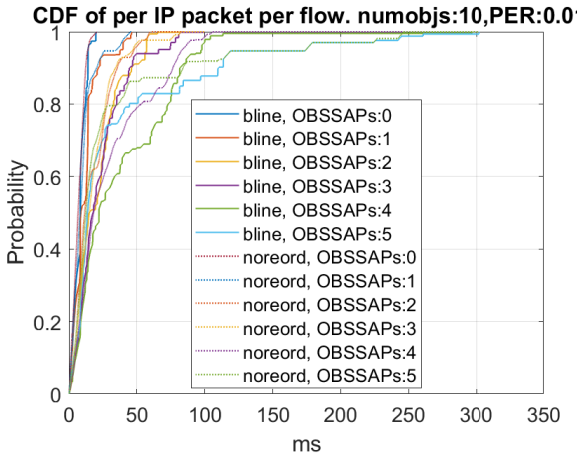
- **Proposed couple of solutions to resolve the HOL blocking issue at MAC layer.**
- **Also, presented some simulation results showing value of resolving this.**

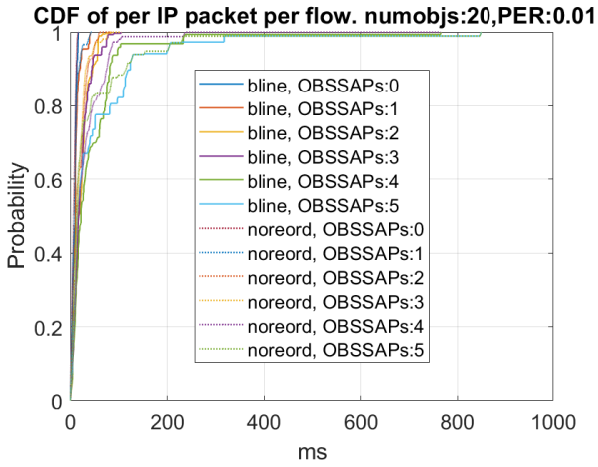
References

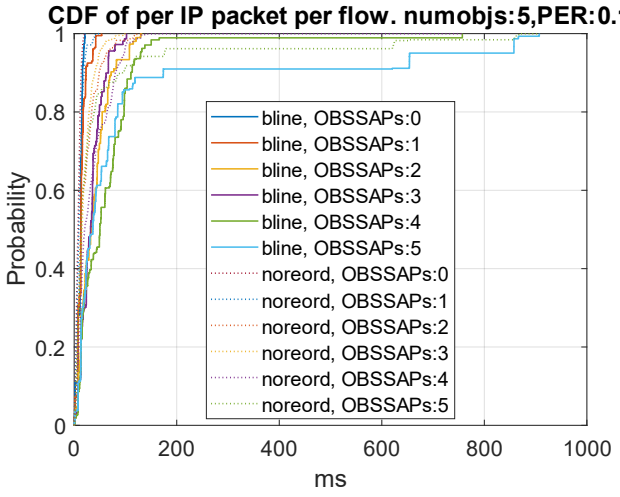
1. 3GPP TR 26.926 V18.1.0 (2023-12)
2. 8I Voxelized Full Bodies—A Voxelized Point Cloud Dataset
3. https://blog.research.google/2020/04/improving-audio-quality-in-duo-with.html

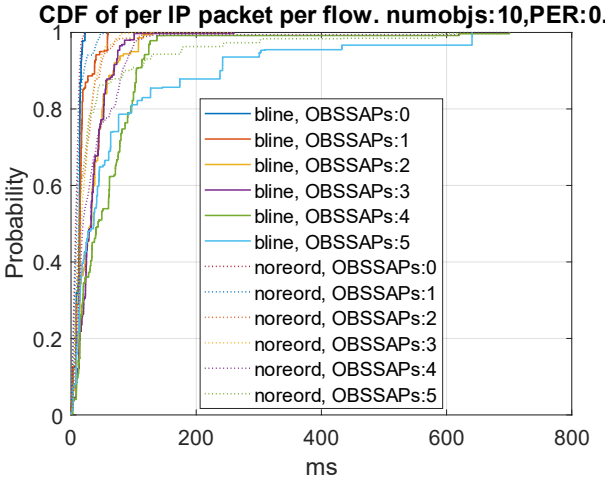
Results (scenario 1)

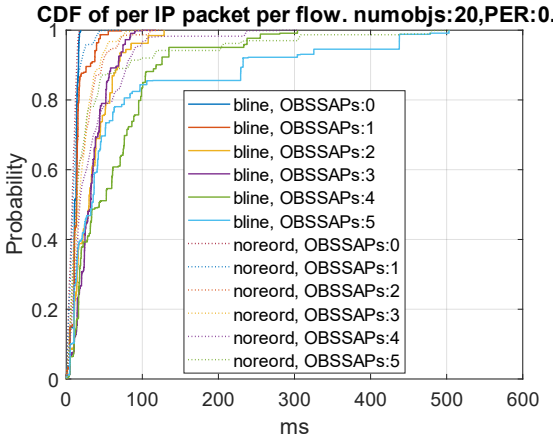




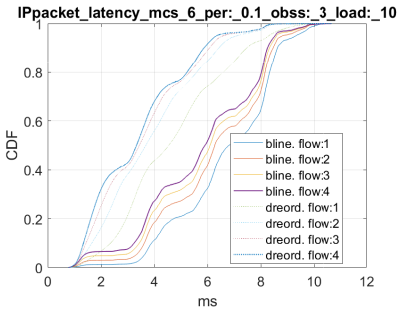


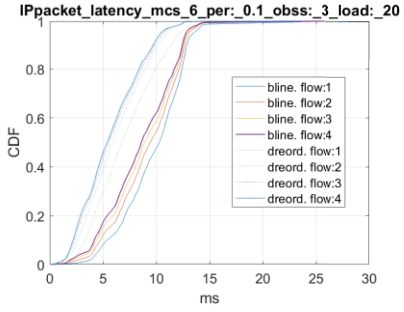


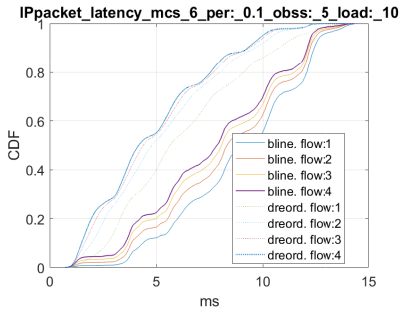


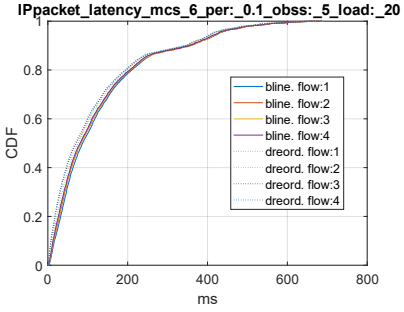


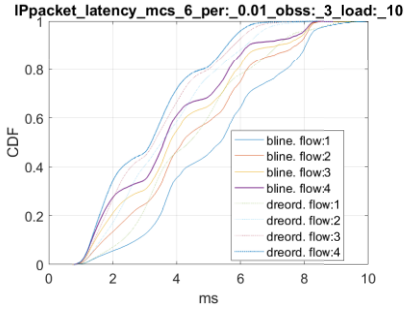
Results (scenario 2 with MCS-6)

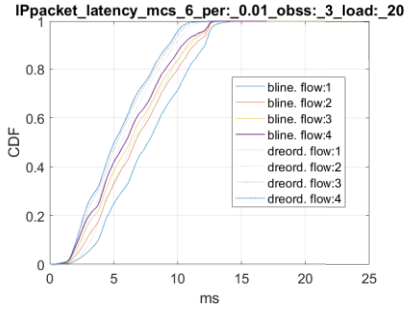


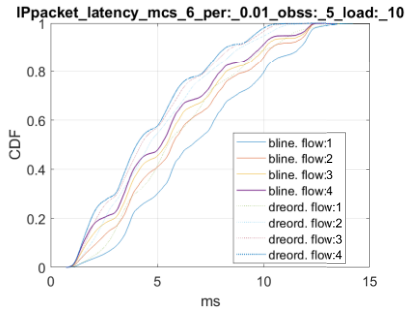


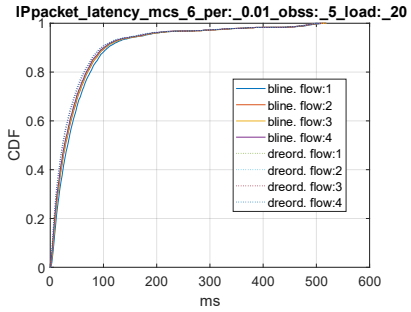






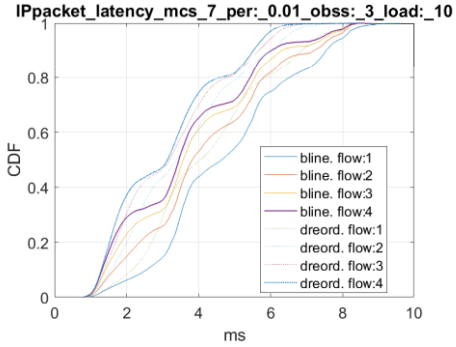


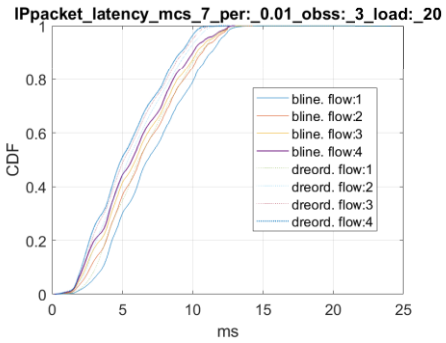


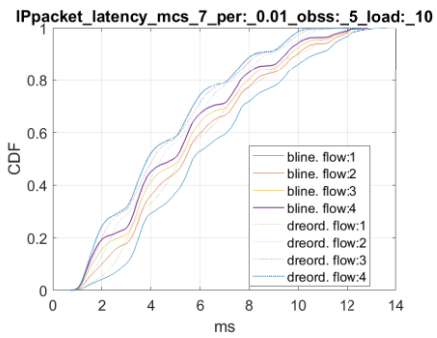


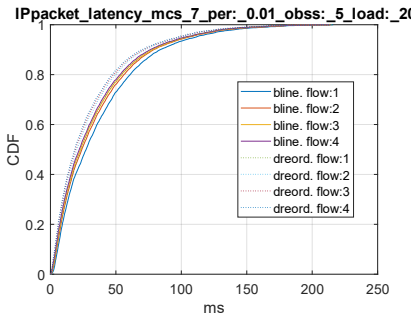
Clear gains until the network below saturation, then the DUT has difficulty delivering all traffic.

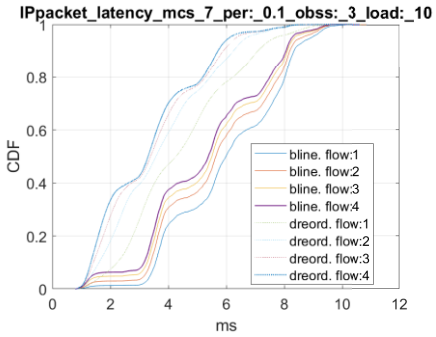
Results (scenario 2 contd. With MCS-7)

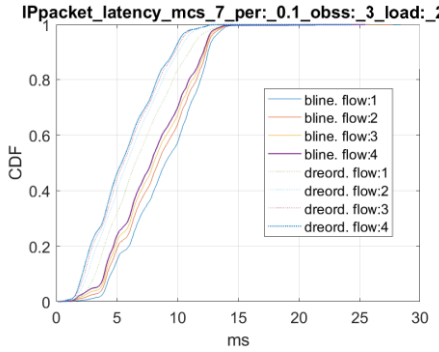


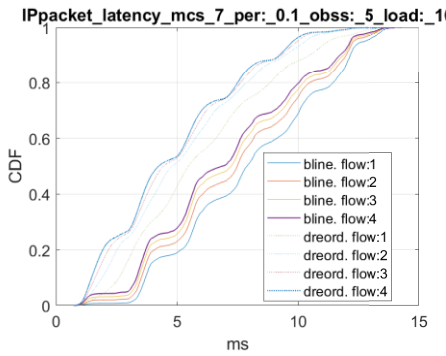


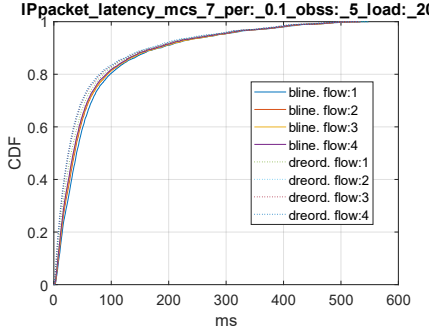












Clear gains until the network is below saturation then the DUT has difficulty delivering all traffic.