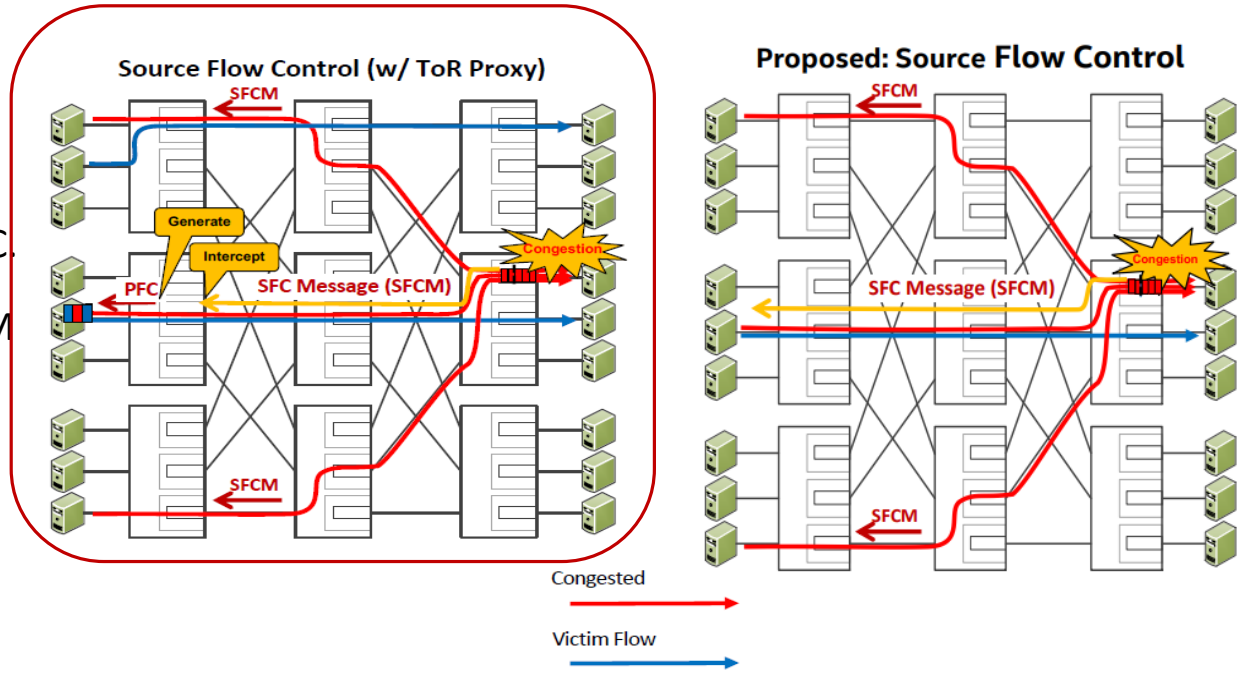
Improvements on SFC Proxy Mode

Lily Lv, Fei Chen (Huawei)

Jeremias Blendin, Yanfang Le (Intel)

Background

- SFC w/ TOR Proxy has been proposed as an optional feature for SFC.
  - When host does not support SFC, source TOR intercepts SFCM and generates PFC to host correspondingly.
- Contribution “Source Flow Control Simulation Results Fairness and Performance” shows similar performance compared with SFC.
- Proxy mode is a fast way to deploy the feature in the field.
  - Uses PFC which is widely deployed in datacenter network.
  - No changes on hosts, only changes on TOR switches.
  - Simple interaction between switches, no new protocol between switch and host.
- This presentation intends to illuminate SFC proxy mode.
  - ‘Friendly’ scenario for proxy mode
  - ‘Unfriendly’ scenario for proxy mode
- This presentation proposes to improve SFC proxy mode.



Results: Background Traffic Performance

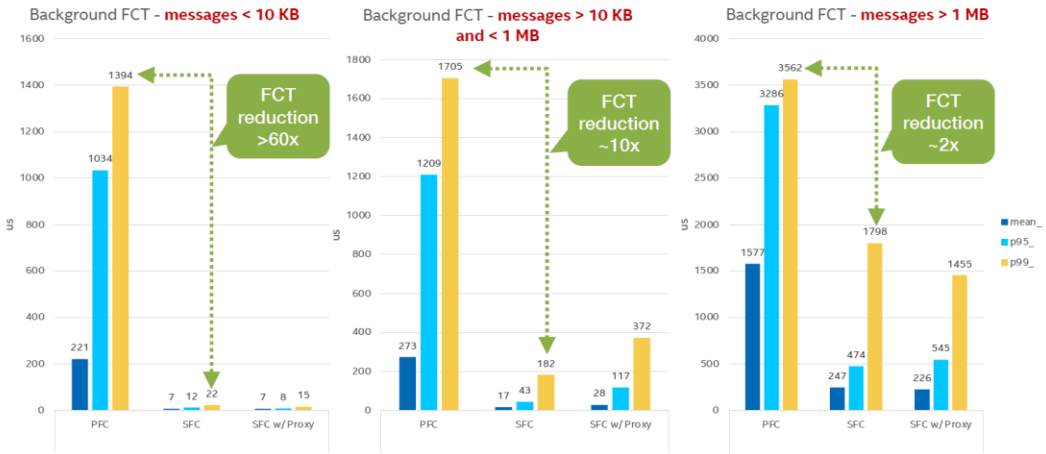


Figure source: https://www.ieee802.org/1/files/public/docs2022/new-blendin-SFC-sim-0522-v01.pdf

Analysis of Proxy Mode Performance

- SFC Proxy Mode pushes PFC to the edge.
  - + No PFC deadlock
  - + No congestion spreading
  - + Less chance for Head-of-line Blocking (HOLB)
    - Still has HOLB on hosts, depending on traffic pattern, what's impact on performance?
- Simulations on the following slides show HOLB impact on performance.
  - Generally, the simulations use the same setting as contribution "Source Flow Control Simulation Results Fairness and Performance", but tune some parameters to show factors that impact performance.
  - Basic setup for simulation
    - Simulation tool: NS3
    - Network topology: 3-tier fat-tree(100/400G), 320 host, 56 switches(20 TOR, 20 Agg, 16 Core)
    - Metrics: FCT of DCQCN+PFC vs. FCT of DCQCN+PFC+SFC Proxy

System Configuration in NS3

- Network
  - 3-tier fat-tree
    - 320 nodes to 20 ToR sw (1 link)
    - 20 ToR sw to 20 agg sw (1 link)
    - 20 agg sw to 16 cor sw (1 link)
  - Switch radix: 20
  - Link delay: 1 us
  - Link speed SW to SW: 400Gbps
  - Link speed NIC to SW: 100Gbps
  - MTU: 1KB
  - RTT: 12us

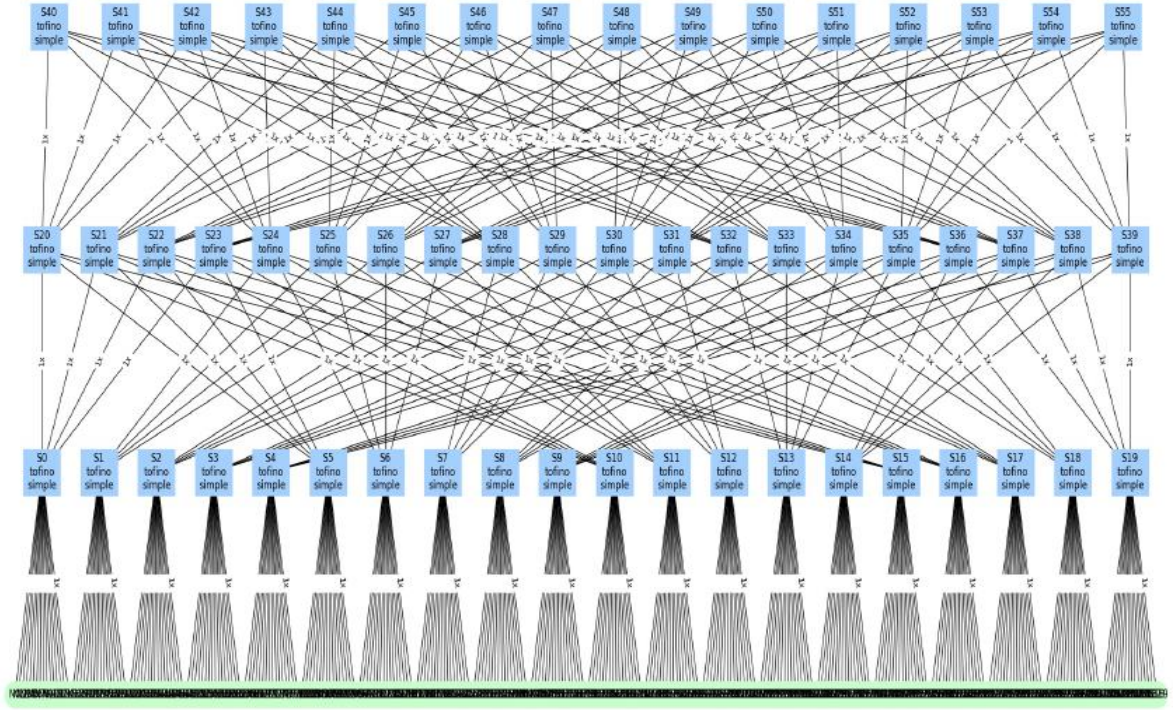


Figure source: https://www.ieee802.org/1/files/public/docs2022/new-blendin-SFC-sim-0522-v01.pdf

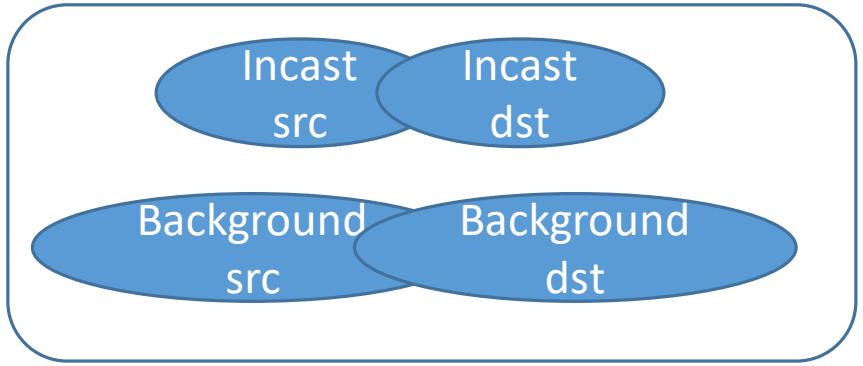
- Background traffic
  - 320 host
  - Traffic based on Google RPC workload
  - Load factor 50%

- Incast traffic
  - 120:1 incast
  - Message size: 256KB
  - The incast traffic load is 8% of the network capacity

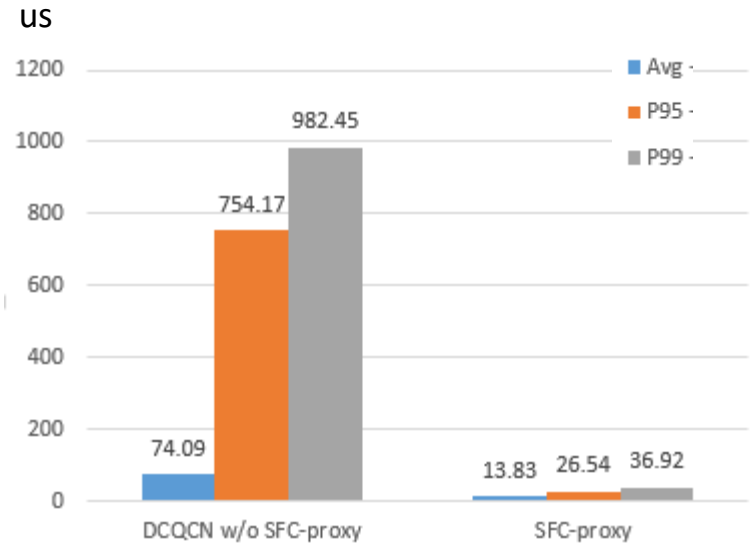
Note: Same topology and traffic as contribution “Source Flow Control Simulation Results Fairness and Performance”

Simulation 1： Traffic model 1 (good perf.)

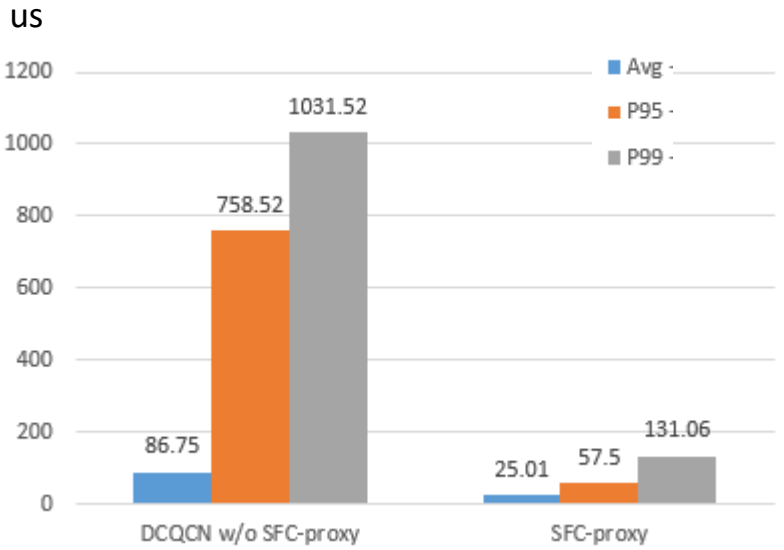
- Traffic model:
  - Incast traffic and background traffic happen on different hosts.
- Result:
  - SFC proxy has better performance compared with DCQCN.



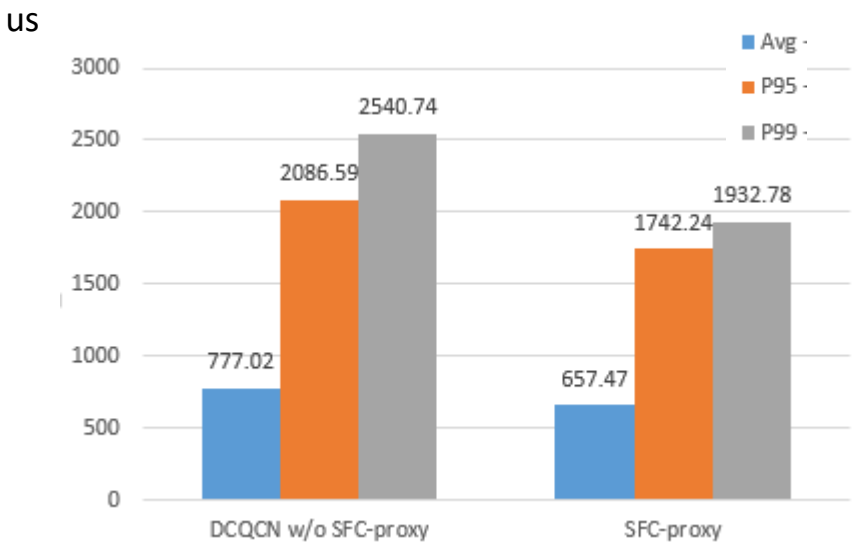
Background FCT - **messages < 10 KB**



Background FCT - **messages > 10 KB and < 1 MB**

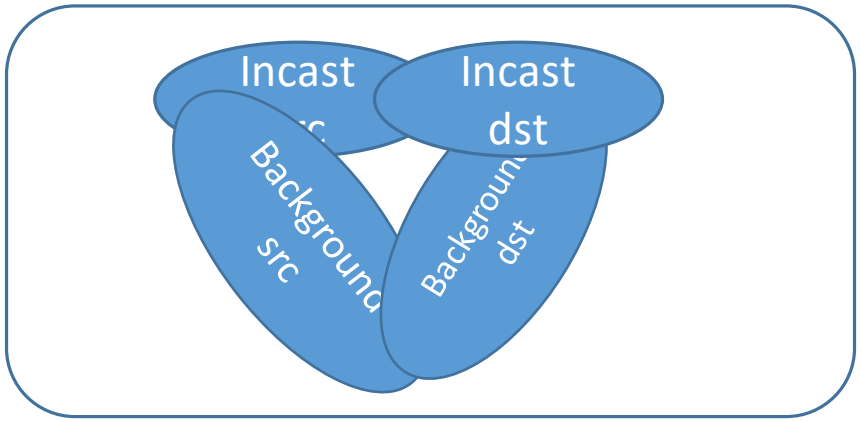


Background FCT - **messages > 1 MB**

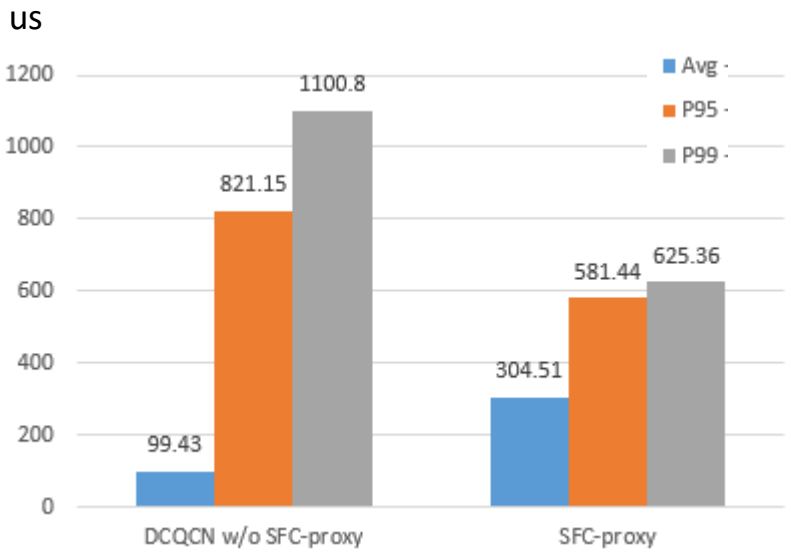


Simulation 2： Traffic model 2 (bad perf.)

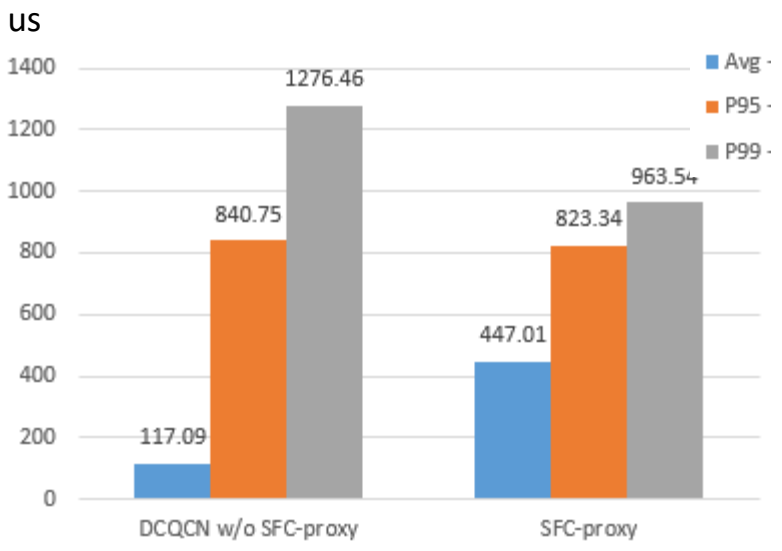
- Traffic model:
  - Incast traffic and background traffic happen on same hosts. SFC proxy may pause the queue which has both congesting flows and non-congesting flows.
- Result:
  - SFC proxy performance is degraded.



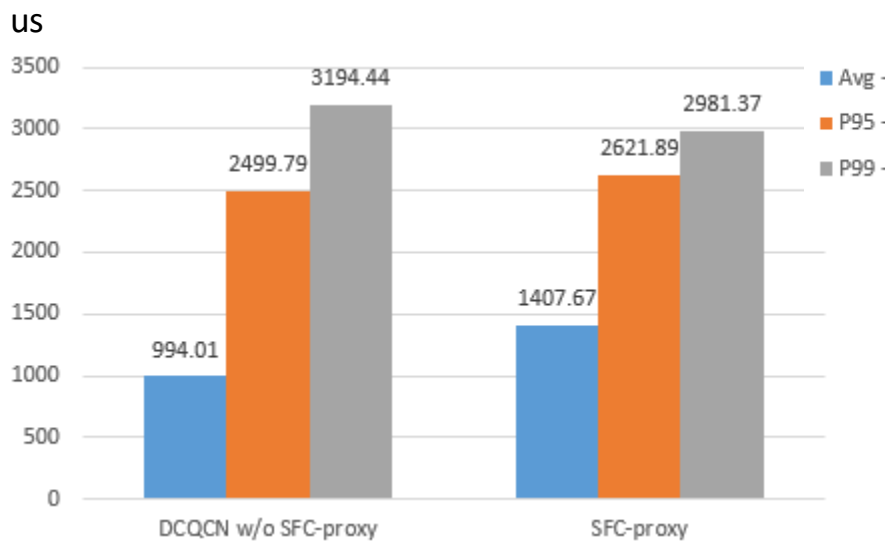
Background FCT - **messages < 10 KB**



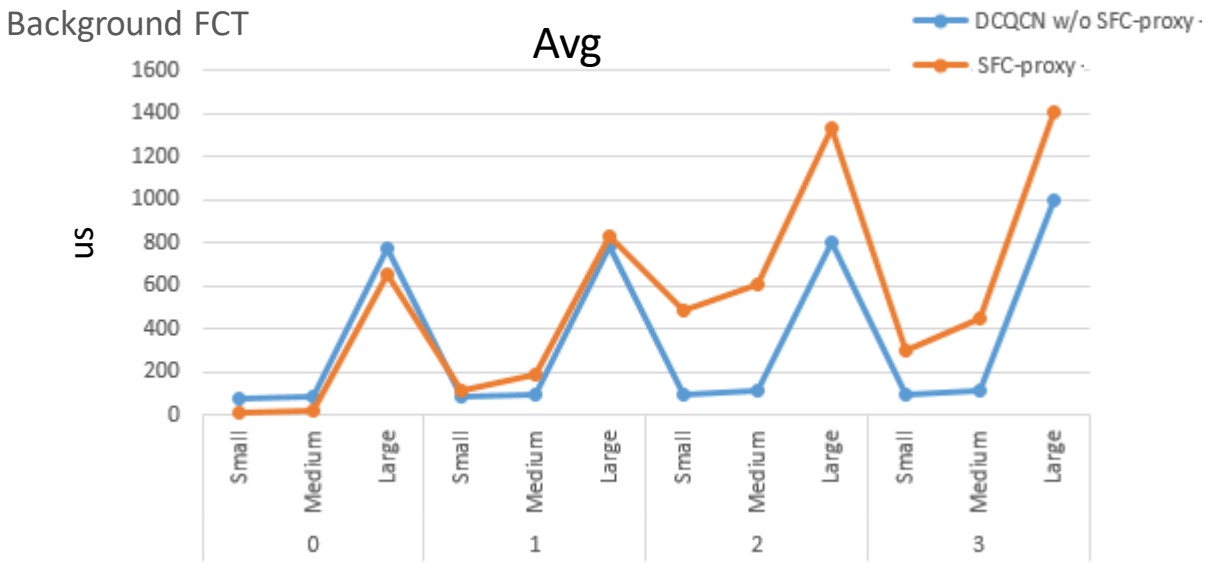
Background FCT - **messages > 10 KB and < 1 MB**

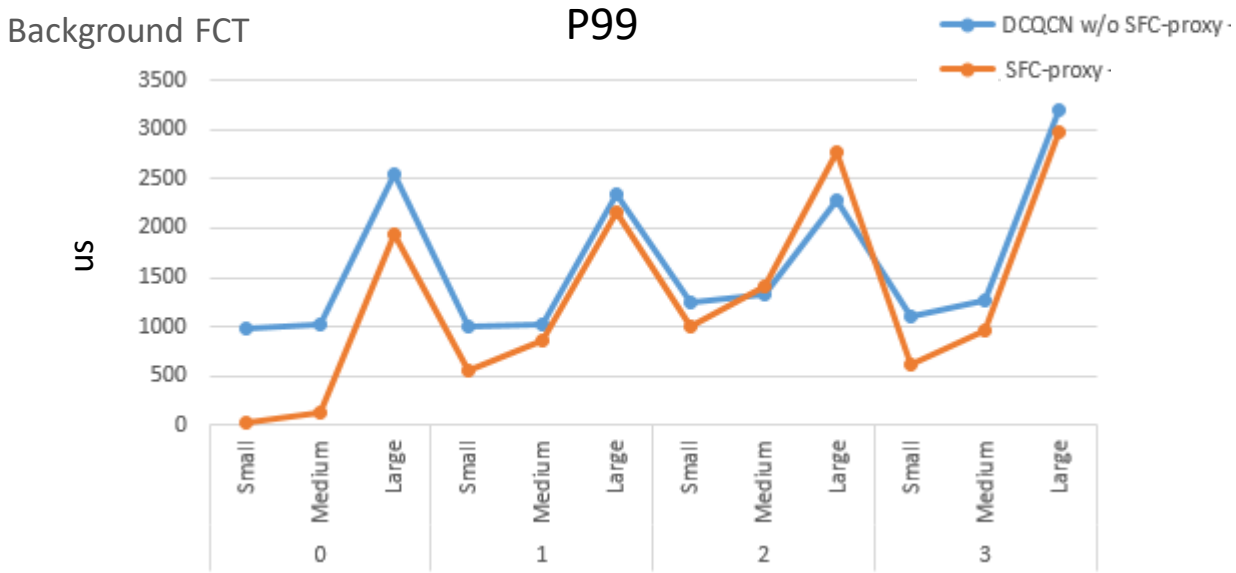


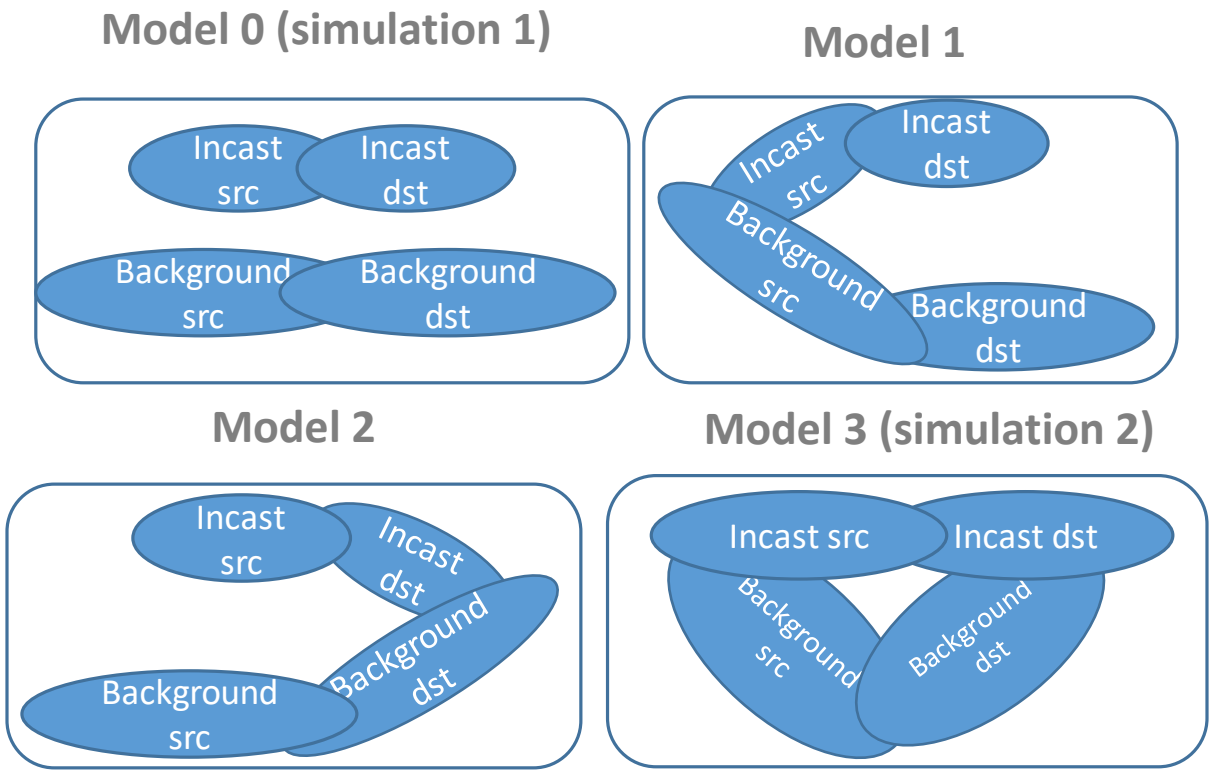
Background FCT - **messages > 1 MB**



Simulation 3： Performance Trend in Different Models







Small: **messages < 10 KB**

Medium: **messages > 10 KB and < 1 MB**

Large: **messages > 1 MB**

Simulation Results Observations

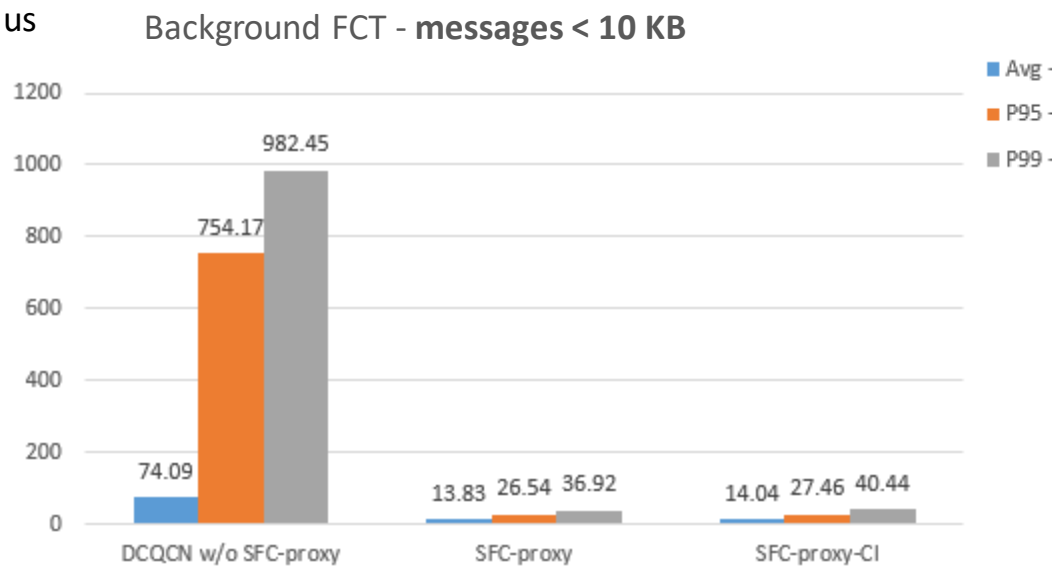
- The only difference between simulation 1 and simulation 2 is the degree of mixing of background traffic and congesting traffic.
  - Simulation 1 has no HOLB at STOR.
  - Simulation 2 has HOLB at STOR.
- HOLB severely degrades SFC proxy performance. In simulation 2, SFC proxy has worse average than DCQCN latency in all cases.
- Degree of mixing of background traffic and congested traffic (different traffic models) has more impact on SFC proxy than DCQCN. In simulation 3, DCQCN performance is more stable (predictable) in different models.
- HOLB needs to be considered when using SFC proxy in the network.

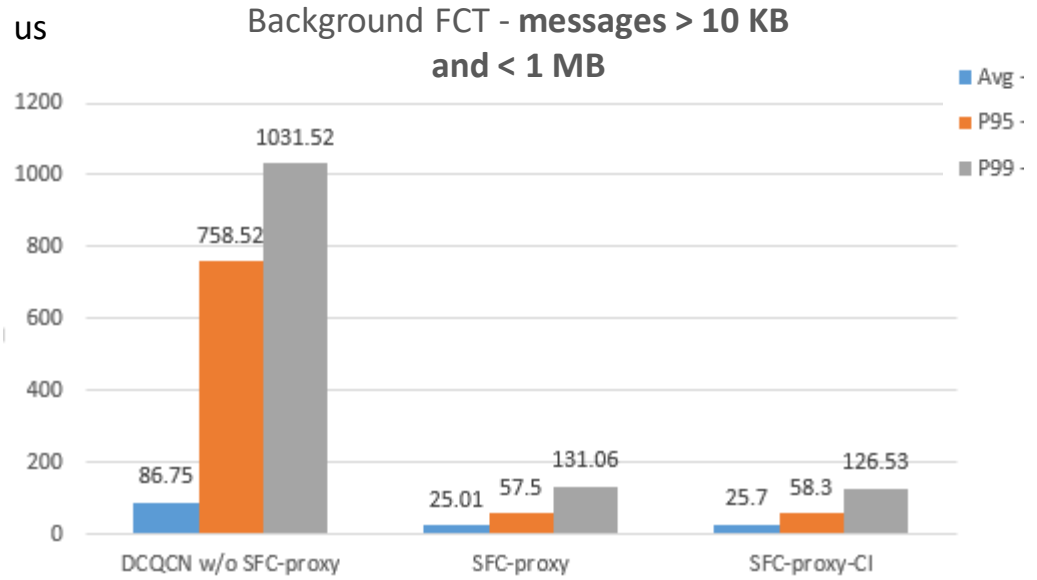
Proposal Inspired by CI (on STOR)

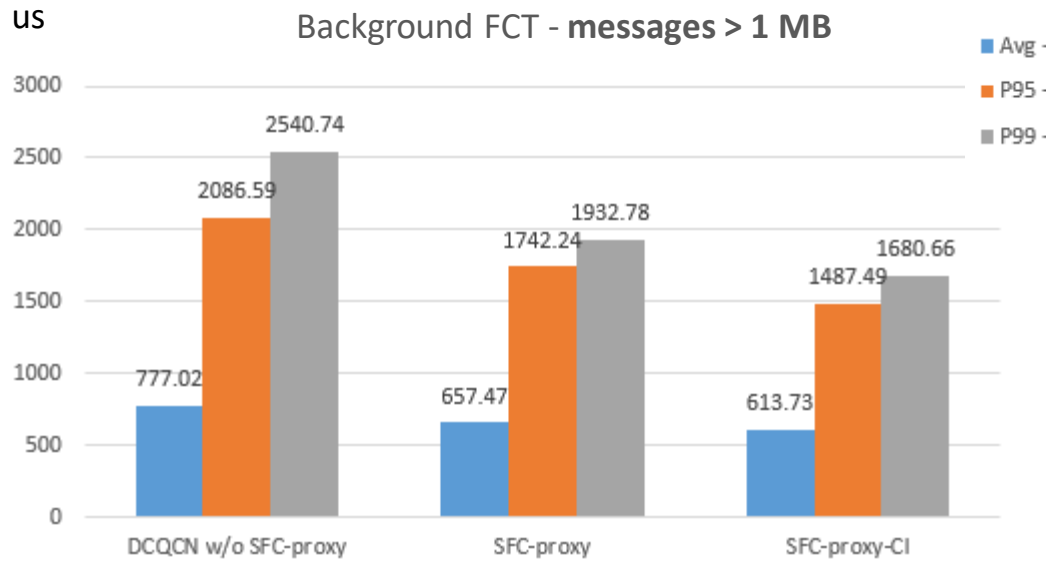
- Idea of congestion isolation
  - DTOR is congested. DTOR isolates the congested flow to its local congestion queue.
  - Once DTOR congestion queue is above certain threshold, DTOR signals upstream switch to isolate the flow.
- ‘Isolation’ concept could be leveraged in SFC proxy to deal with HOLB.
  - Idea: Isolate the congested flow to congestion queue on STOR
    - The congestion point(DTOR) generates the congestion control message --- SFCM.
    - SFCM includes information to identify congested flow, and is sent by congestion point to the source of the congested flow.
    - Source TOR receives SFCM and is indicated to isolate the congested flow to its local congestion queue.
  - Feasibility :
    - Congestion happens on DTOR while STOR is typically not congested. Therefore, STOR has free buffer to use as the congestion queue.
    - In most cases, incast on DTOR is caused by flows from multiple hosts/STORs. When distributing the congested flows on each STOR, it will not need large buffer to queue the flows. Making use of free buffer on STOR switches could absorb the burst and mitigate HOLB to a large extent.
  - Notes：
    - Host is unware of the isolation.
    - The congestion queue on STOR is paused until the pause time is up or receiving SFCM-resume.

Re-run Simulation 1

- Result:
  - SFC proxy with isolation has similar performance as SFC proxy

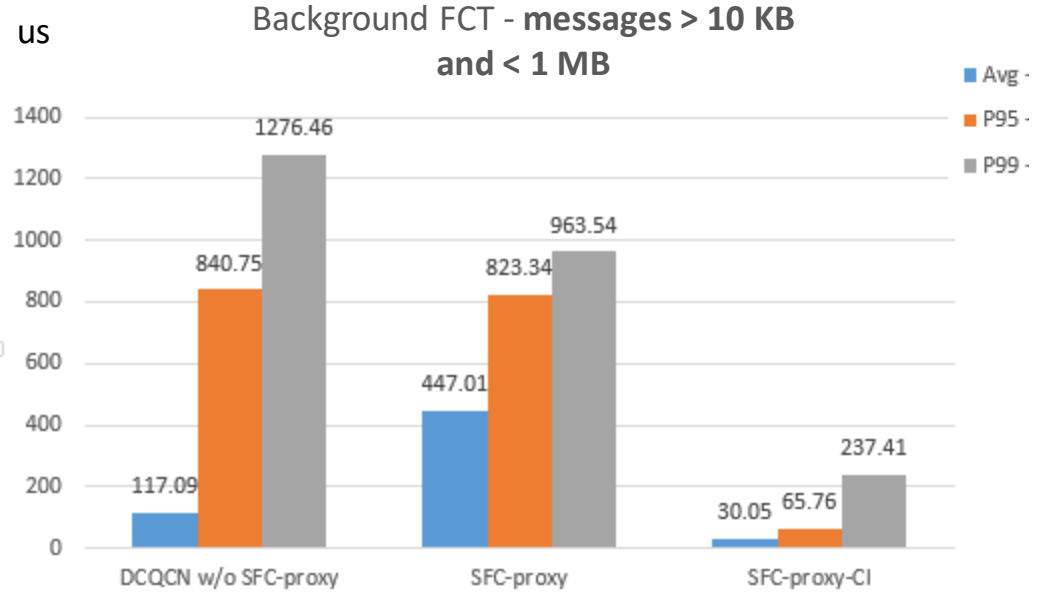


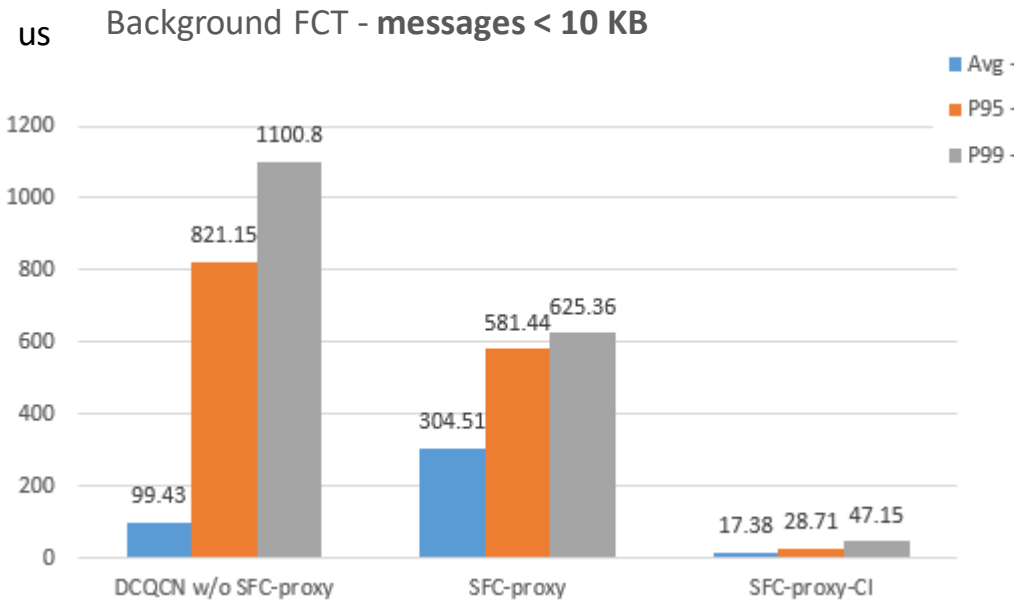


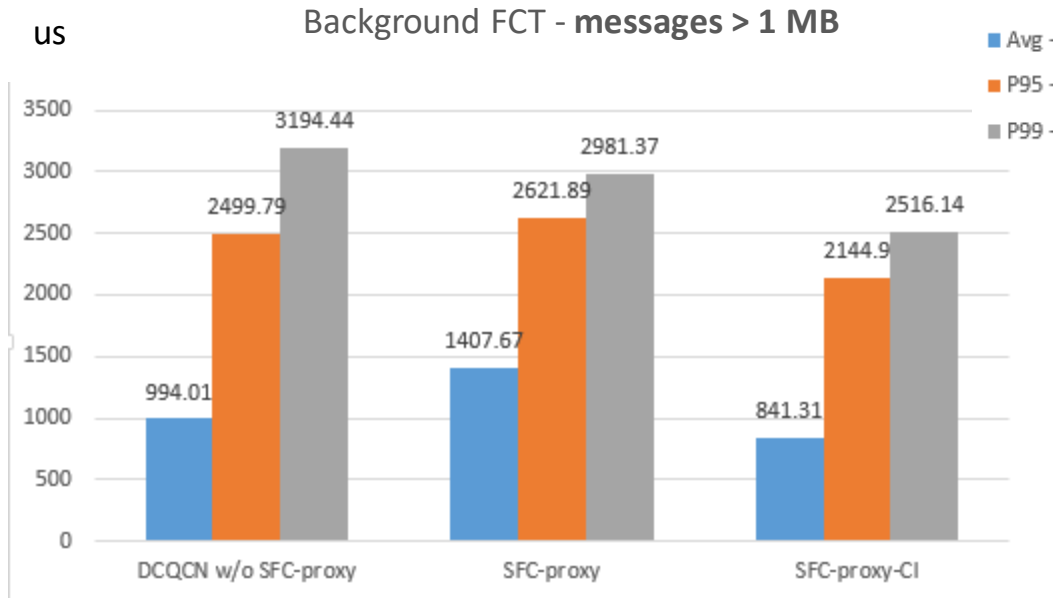


Re-run Simulation 2

- Result:
  - SFC proxy with isolation performance is better than DCQCN and SFC proxy.

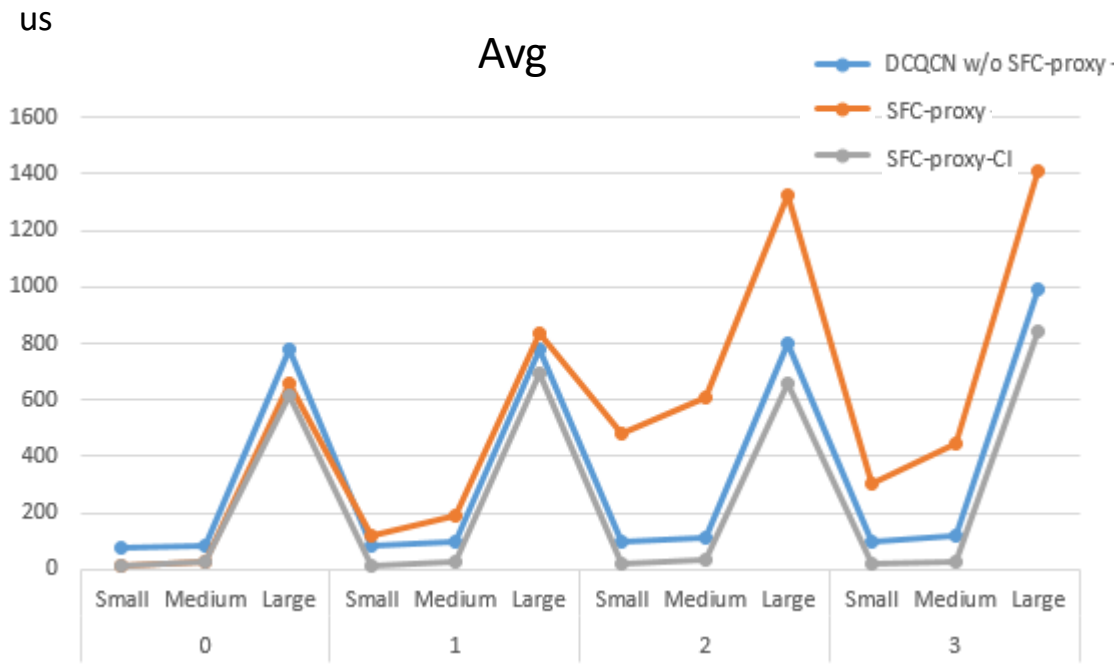


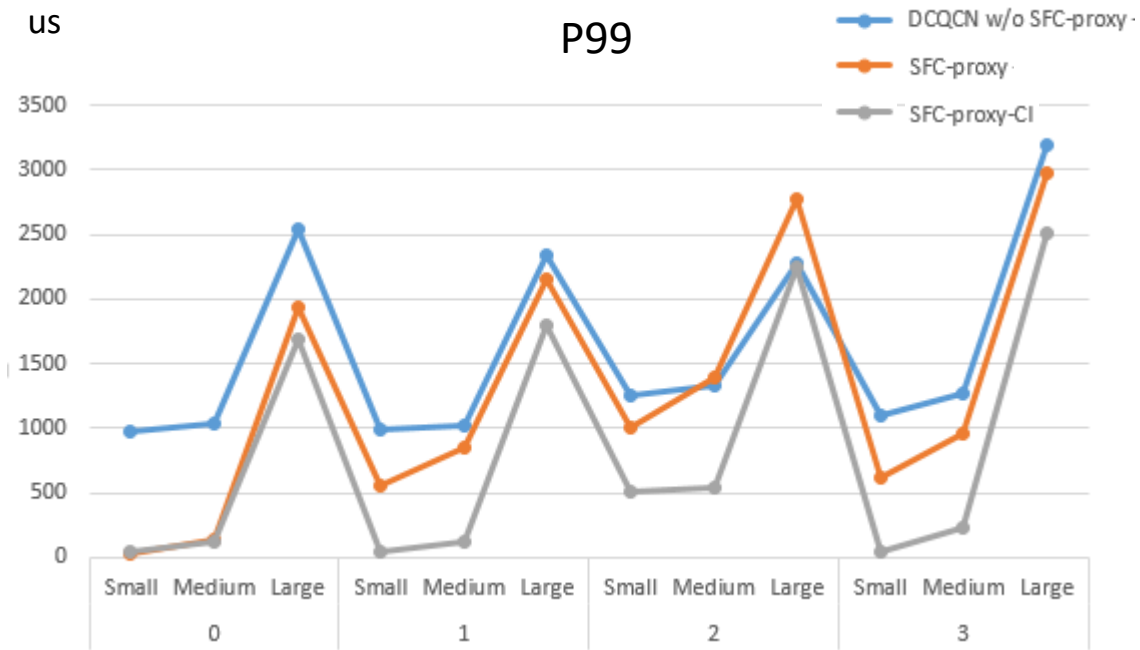




Re-run Simulation 3

- Result:
  - SFC proxy with isolation performs stable in different traffic models compared with SFC proxy.
  - SFC proxy with isolation has better performance in different traffic models.



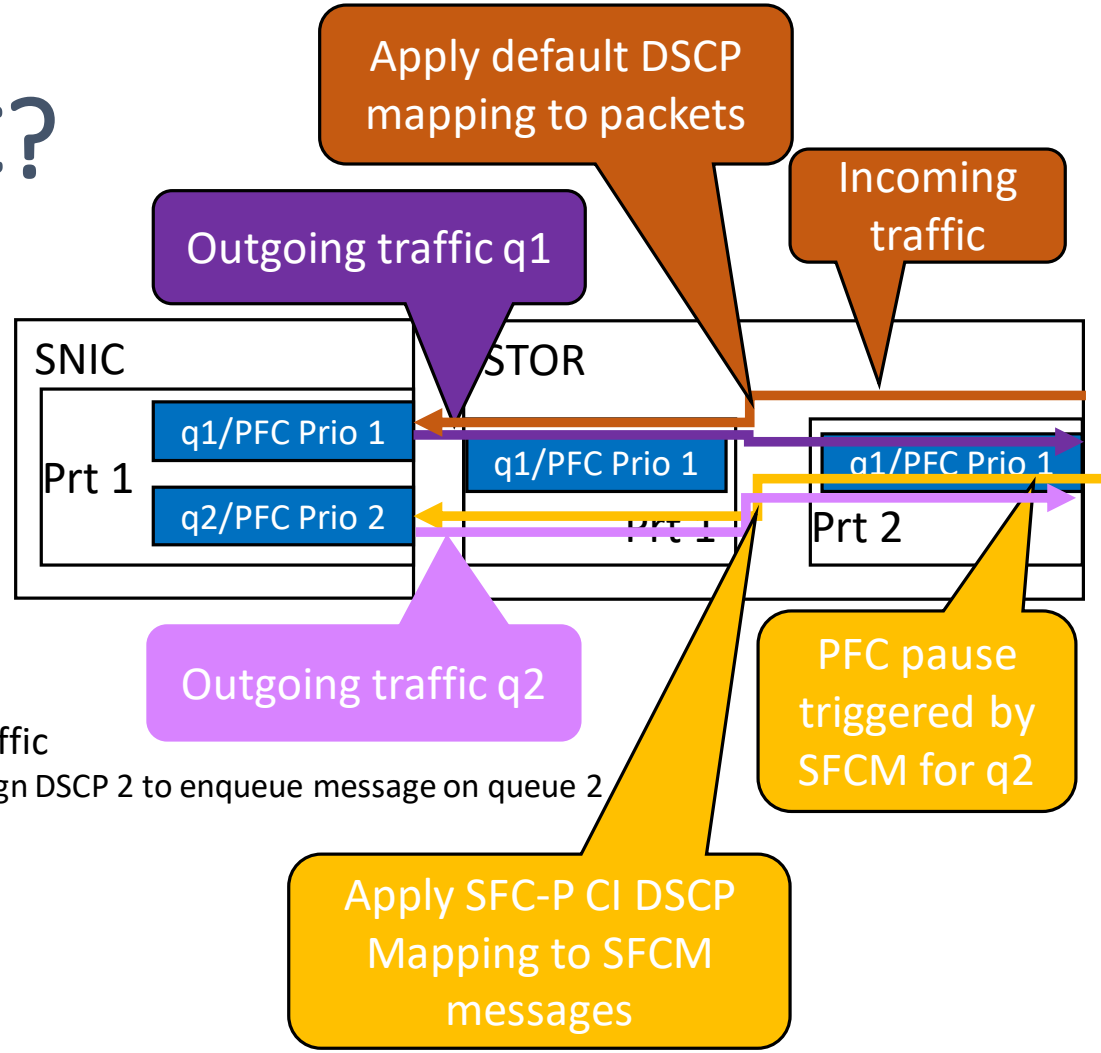


Proposal Inspired by CI (on NIC)

- Idea: Isolate the congested flow to congestion queue on Source NIC (SNIC)
  - If a sender queue in the sender NIC (SNIC) is paused, use a second queue to send new messages
  - Like Isolation concept in STOR, but in SNIC
  - Use case: RDMA clusters
- Feasibility :
  - Requires no NIC HW changes, pure software implementation possible
  - Use special DSCP to PFC priority mapping in SNICs and STORs for incoming SFCM
    - Details on next slide
  - Before adding a new message on SNIC to NIC:
    - Select unpaused TC using matching DSCP value for message
  - Challenges:
    - Reserves multiple TCs, DSCP values for a single in-network RDMA traffic class
    - Number of CI “standby queues” limit the total number of SFC-enabled traffic classes

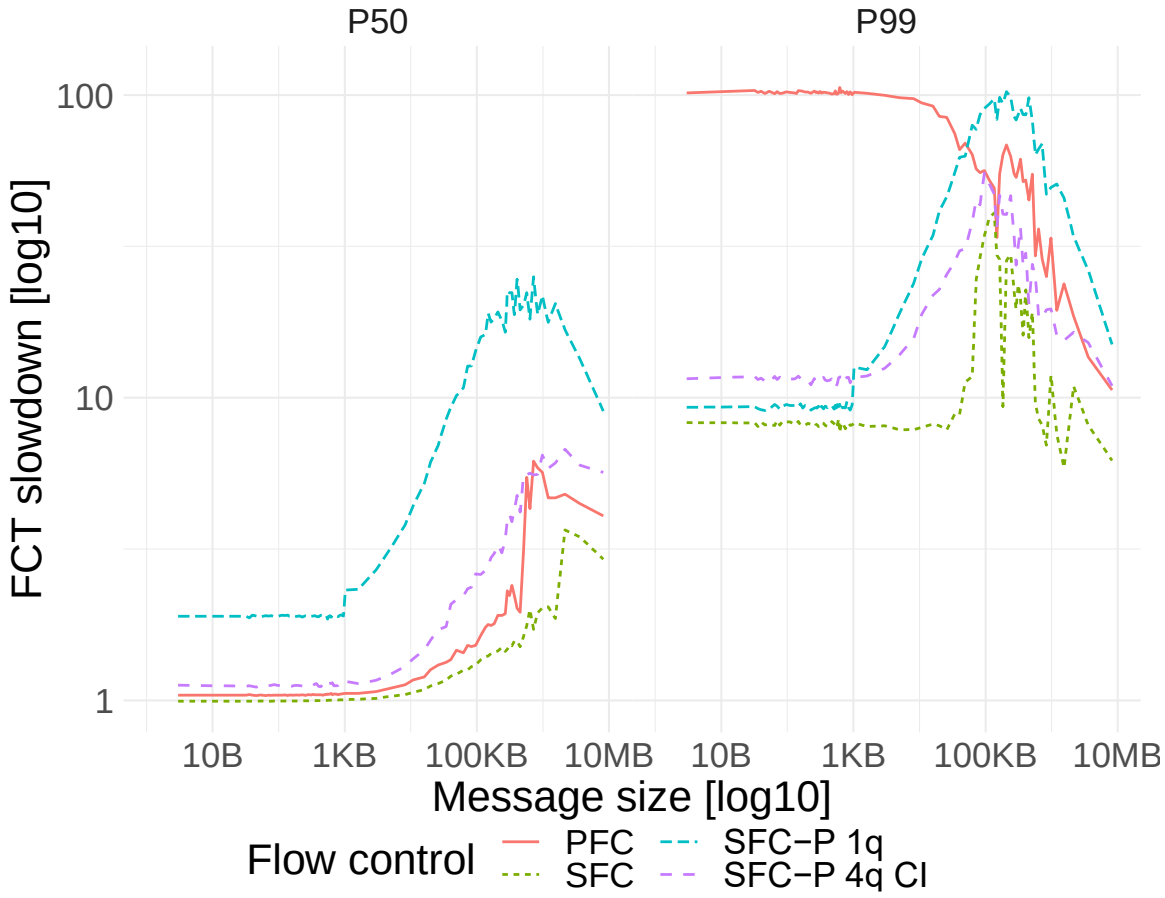
How does CI work on the NIC?

- Example DSCP/queue/PFC priority mapping:
  - Assume 1 SFC traffic class in the network
  - On SNIC/STOR-SFCM:
    DSCP 1 -> queue 1 -> PFC priority 1
    DSCP 2 -> queue 2 -> PFC priority 2
  - In network including STOR outside of SFCM handling:
    - DSCP 1,2 -> queue 1 -> PFC priority 1
- SNIC
  - Reserve extra outgoing queues and PFC priorities for SFC-enabled traffic
    - When sending new message check if PFC priority 1 is paused, if it is, assign DSCP 2 to enqueue message on queue 2
  - Allows SNIC to "sidestep" congestion for new messages
- On congestion point:
  - Send SFCM containing the DSCP of the original message
- On STOR:
  - Apply default network DSCP to PFC mapping for actual traffic
  - Receive SFCM and apply special SNIC/STOR-SFCM DSCP to PFC priority mapping to determine which PFC priority to pause on the NIC
  - Send PFC pause frame with priority based on SNIC/STOR-SFCM mapping
  - STOR pauses only the PFC priority that the original message was sent on



Simulation

- SFC-P CI
  - Use 4 outgoing queues on SNIC in total
- Environment
  - Same as described in CI
- Setup
  - Model-3 (mixed)
  - Protocol: DCQCN+W
  - Background:
    - Google RPC, 30%
  - Incast:
    - 60:1, 5%
    - 250KB message size
- Results
  - Using CI with 4 queues improves FCT significantly



SFCM Function in SFC Proxy

- SFCM is the signal to pause/resume the flow at the source TOR
  - From DTOR's view, the flow is paused at the source TOR no matter it is paused due to PFC or due to isolation.
- In standard, describe SFCM as pausing signal of congested flow at the source TOR.
    - Approach 1: PFC
    - Approach 2: Isolation
    - Implementation could choose its own approach based on its capability
- The standard specifies information carried in the SFCM (e.g. flow) and 'flow pausing' as the external behavior triggered by SFCM, but does not limit the 'pausing' approaches for an implementation.
- The standard has informative content about HOLB issue and the possible 'isolation' solution.

Summary

- SFC Proxy mode is optional mechanism/feature in SFC.
- However, only asserting PFC may encounter HOLB issue which will severely degrade the performance.
- An improvement inspired by Qcz is proposed to solve HOLB issue.
- Agree upon what is specified in the standard: what is normative vs. what is informative
  - Proposal on the normative content: SFCM contents, external behavior triggered by SFCM
  - Proposal on the informative content: implementation suggestions on STOR when receiving SFCM

BACKUP

Flow Control / Congestion Control Configurations

- PFC configuration
  - Dynamic PFC threshold
  - Total buffer 32M
- SFC configuration
  - Trigger threshold: 1000KB
  - Target threshold: 200KB
  - Suppression period: pause/resume messages

- DCQCN configuration
  - Fast recovery step: 5
  - Gain: 0.00390625
  - Byte counter: N/A
  - Timer: 900
  - Alpha timer: 1
  - AI: 50Mbps
  - Hyper AI: 100Mbps
  - CNP period: 4us
  - Window: BDP
  - ECN threshold: 800KB/200KB