SFC simulation

Lihao Chen (lihao.chen@huawei.com)
Paul Congdon (paul.congdon@outlook.com)
Lily Lv (lvyunping@huawei.com)
Sivakolundu, Ramesh (sramesh@cisco.com)

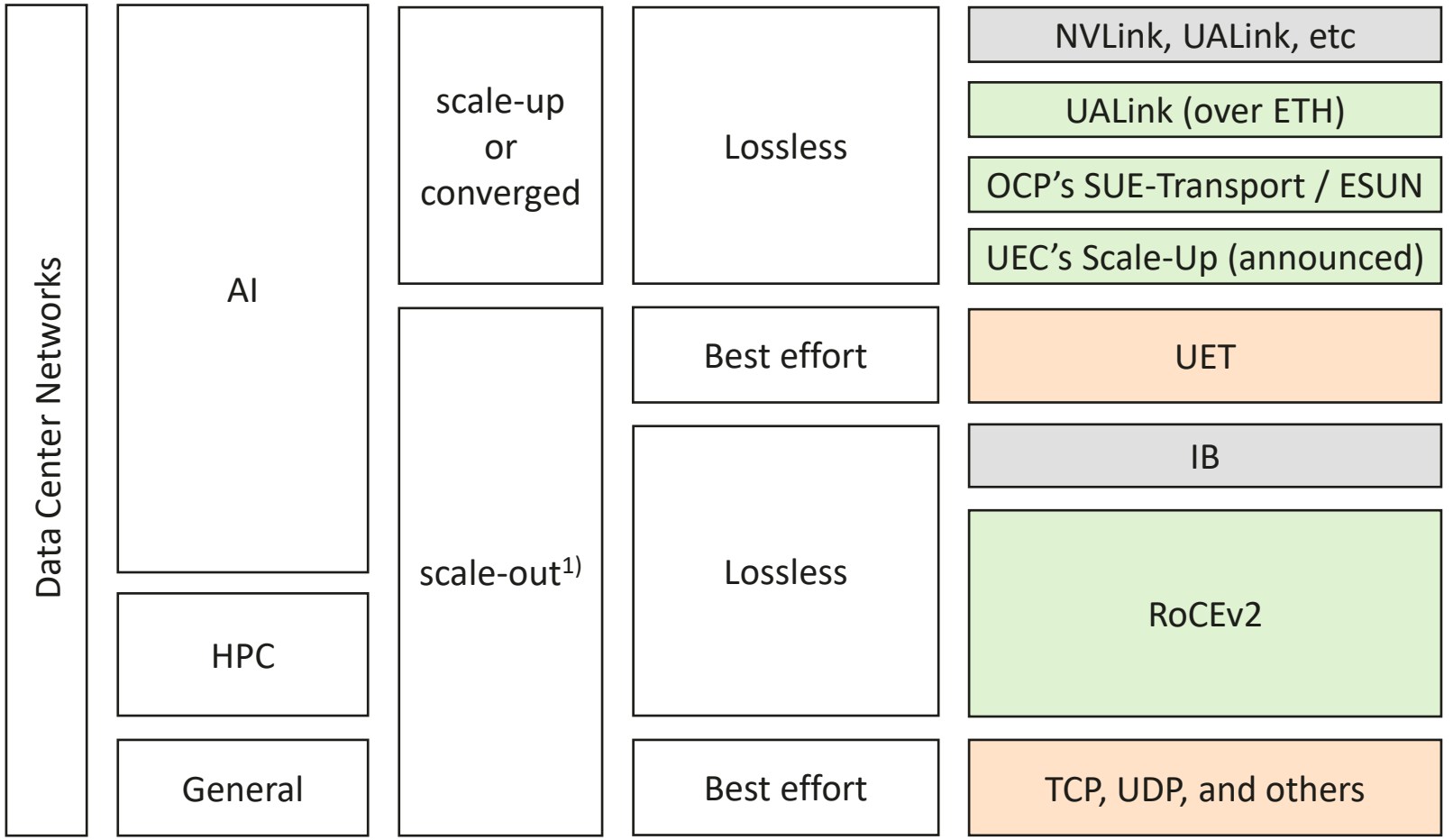
Previous works and discussion

- Sep. Interim: https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-simulation-pfc-dcqcn-0925-v01.pdf shows simulation results of SFC in AIDC backend network use cases.
  - > Part 1, prove of concept: SFC precisely pauses the source of incasts, improving the performance of the victim flow.
  - > Part 2, SFC robustness: The key is to avoid pausing too long, as bandwidth under-utilization can harm performance. All other situations are fine.
  - > Part 3, AIDC backend network: SFC improves the tail latency of prefill task (inference) as well as training.
- Disposition:
  - > Clarifying the SFC use case taxonomy, the queuing architecture.
  - > Regarding the simulation: Do a 3-layer simulation, sensitivity analysis (PFC buffer size, SFC threshold, SFC pause time, SFC RTT, incast, etc.).

- July Plenary: https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-consideration-and-design-0725-v01.pdf
  - > Disposition: Need more simulation.
  - > More previous contributions are listed in Page 2.
- May Interim: https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-computation-simulation.pdf
  - > Describe how to calculate SFC parameters, as well as SFC simulation and verification results.
- …

Taxonomy

- This slide illustrates where SFC applies within DCNs and it highlights deployment scenarios rather than defining transport protocol taxonomies or classifications.



Non-Ethernet based

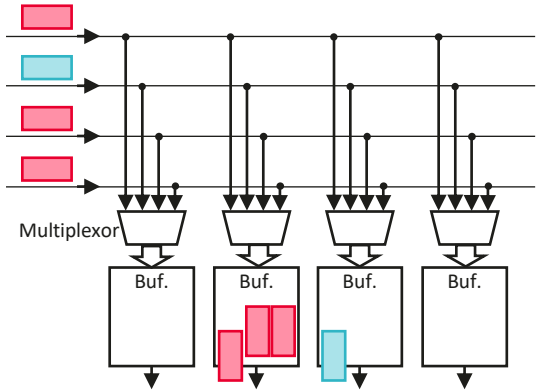
SFC can be used.

SFC may also be used.

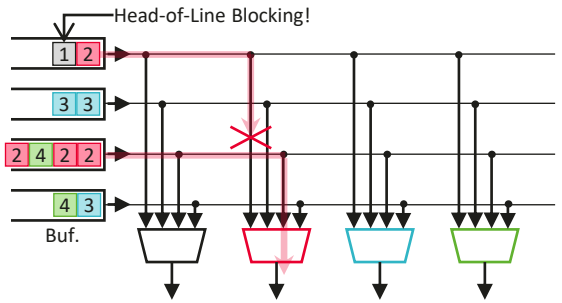
1) *Regarding DCN for HPC and general-purpose, the scale-up is mainly within the server level and thus not presented in this taxonomy.*
2) *There are other proprietary or non-proprietary protocols not listed.*

The queuing architecture

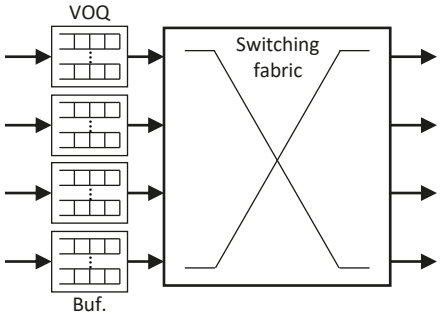
	Output Queue (OQ)	Input Queue (IQ)	Virtual Output Queue (VOQ)
Buffer position	output side	input side (a FIFO)	input side (per output port)
Pros	Extremely strong theoretical performance.	Implementation is relatively simple.	Achieving near OQ performance without HoLB.
Cons	Need N (the number of ports) times larger switching fabric bandwidth, resulting in extreme high cost.	Head-of-Line Blocking!	Need to manage N^2 queues. Implementation is relatively complex.
Summary	A theoretical model. Rarely used in actual devices.	Poor throughput performance. Low-end devices only.	Good balance between performance and price. Commonly used nowadays.



Output Queue (OQ)



Input Queue (IQ)

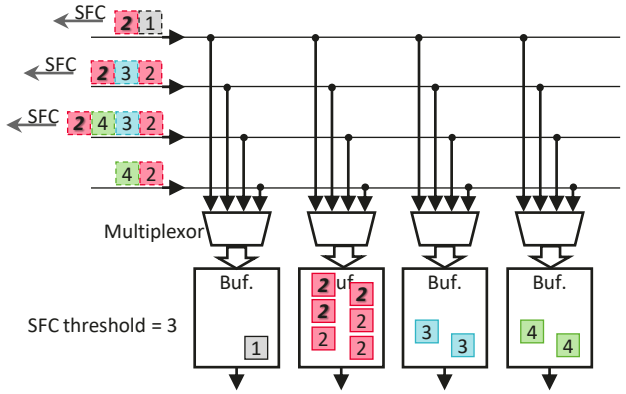


Virtual Output Queue (VOQ)

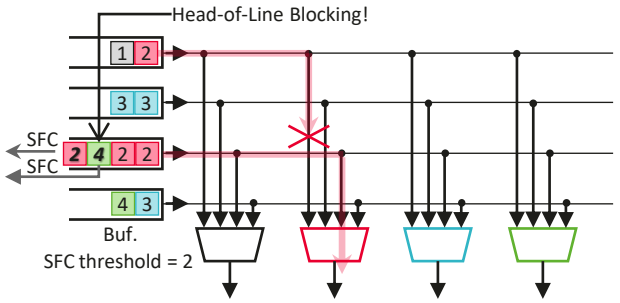
The SFC mechanism, especially in terms of threshold configuration and invocation, is related to the queuing architecture.

The queuing architecture's effect on SFC

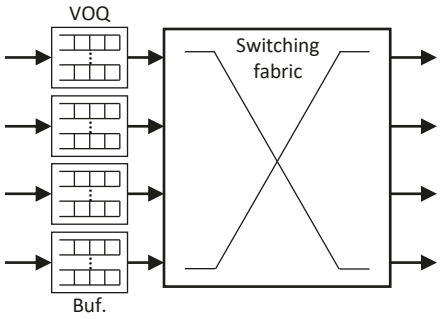
- Using SFC on the OQ architecture (to describe the externally observable behavior of SFC) is OK.
  - > Pay attention that the SFC threshold is applied at the egress buffer and the calculation should account for traffics from all ingress ports.
- Using SFC on the IQ architecture will cause the Head-of-Line Blocked victim flows to be SFCed.
  - > As shown below, packet from flow No.4 is HoL Blocked, and SFC is wrongly trigger to the source of flow No.4.
- Using SFC on the VOQ architecture is recommended for deployment.
  - > Pay attention that the SFC threshold is applied at the ingress side and the calculation should be on a per-input basis.
  - > The actual allocation method of VOQ buffers (shared and/or dedicated) also has an impact. If future SFC specifications explicitly address configuration guidelines for deploying SFC in VOQ architectures, this topic will need to be further discussed.



Output Queue (OQ)



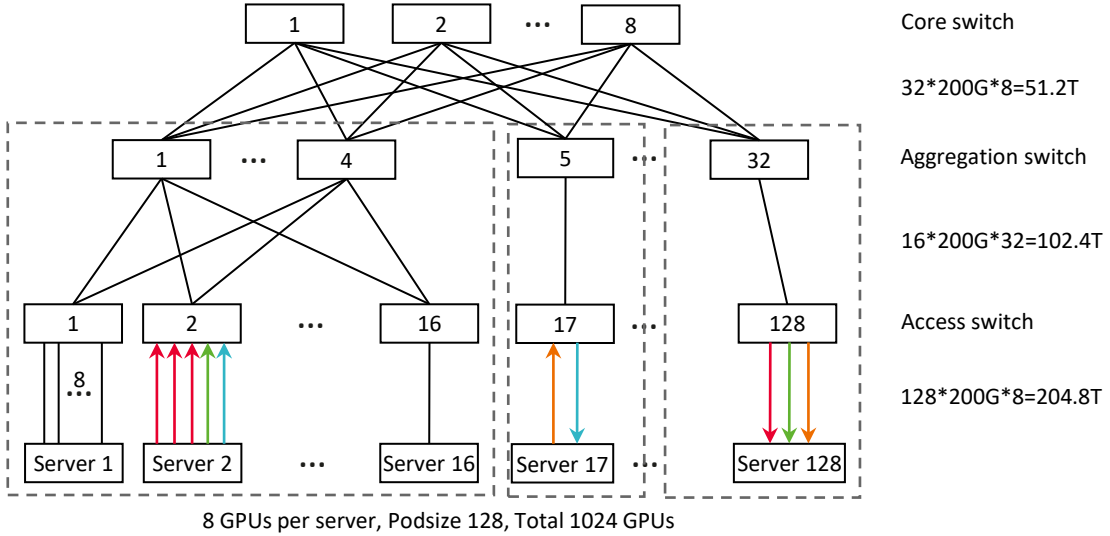
Input Queue (IQ)

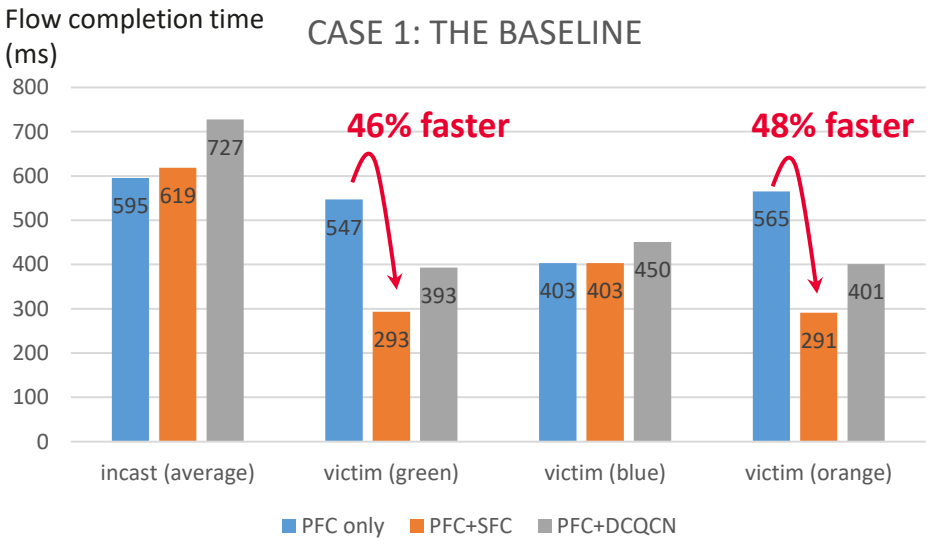


Virtual Output Queue (VOQ)

Simulation Part 4: 3-layer CLOS

- Network 3-layer CLOS
  - > 8 core switches, 32 aggregation switches, 128 access switches
  - > 1024 GPU, all 200G
- Basic settings
  - > Link delay: 150ns. Switch process (fixed) delay: 300ns
  - > Max. packet size: 4KB
  - > Buffer size: 800KB. PFC threshold: 780KB. DCQCN K: 200KB.
- Flow settings
  - > 3-to-1 incast (red arrows), and other victims (green, blue, and orange), each 5MB data
- User-defined SFC settings:
  - > SFC threshold: 200KB; SFC pause time: 20us; SFCM Min. interval: 20us.
  - > Parameter calculation process refers to https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-computation-simulation.pdf . See the next page for details.



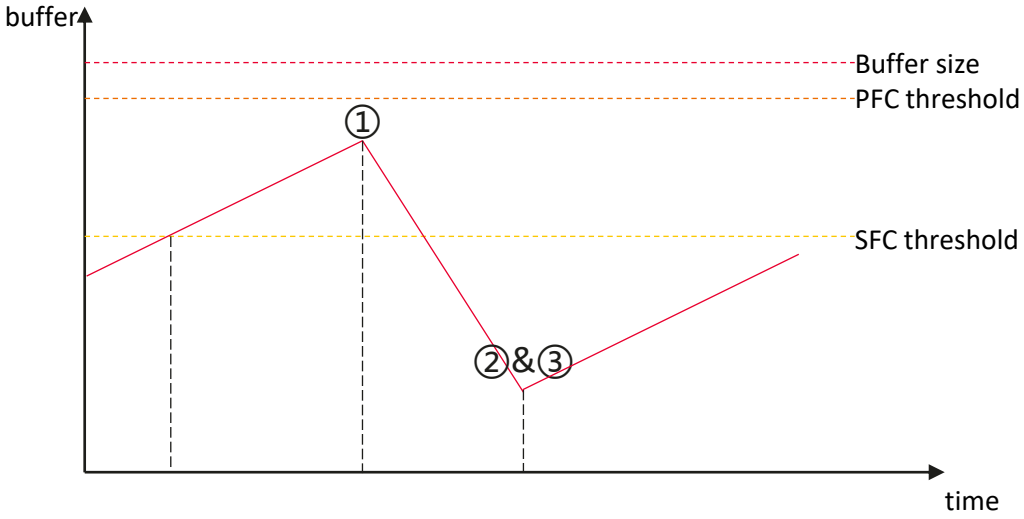


SFC precisely pauses the source of incasts, improving the performance of victim flows.

Parameter calculation process

- Basic settings
  - > Link delay: 150ns. Switch process (fixed) delay: 300ns
  - > Buffer size: 800KB.

- PFC headroom > (150ns*2+300ns)*200Gbps/8 = 15KB
  - > Set PFC threshold = 780KB
- ①SFC headroom (i.e., from SFC threshold to PFC threshold) > (N-1)*(150ns*5*2+300ns*9)*200Gbps/8 = (N-1)*105KB
  - > if N=5 (i.e., 5-to-1 incast), then SFC headroom > 420KB
  - > if N=3,7, then SFC headroom > 210, 630KB accordingly
  - > Set SFC threshold = 200KB
- ②t_{SFC_pause} * 200Gbps > 420KB, then t_{SFC_pause} > 16.8us
- ③t_{SFC_pause} * 200Gbps < (420+200)KB, then t_{SFC_pause} < 24.8us

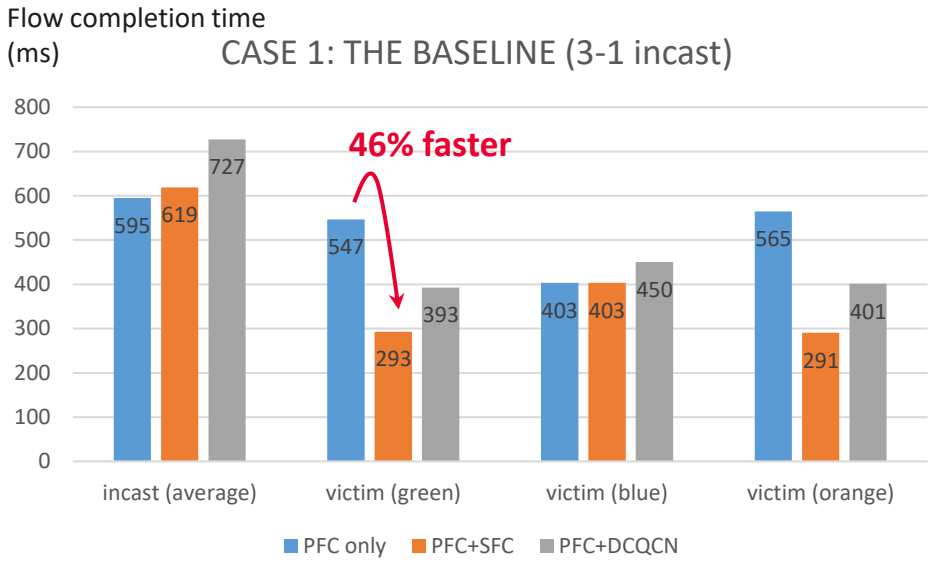


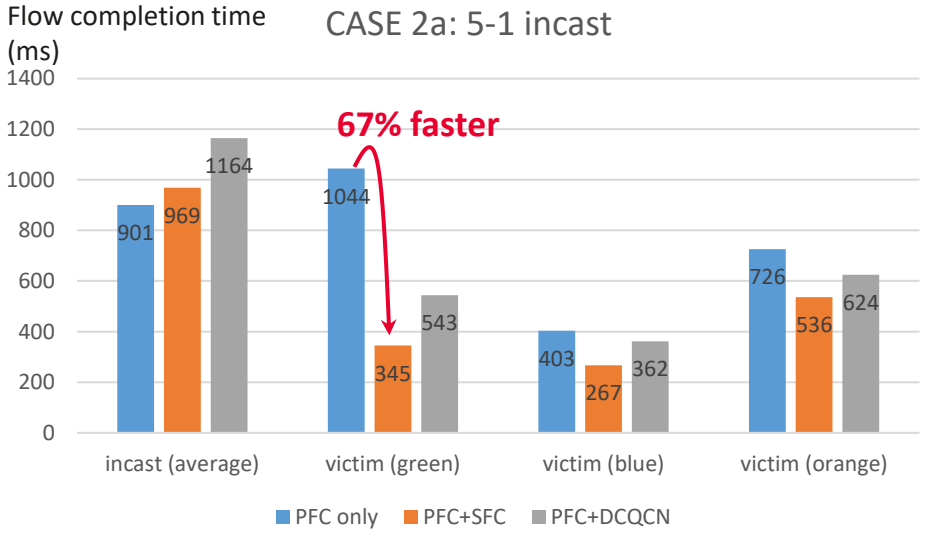
Try to avoid:
①Buffer usage exceeding the PFC threshold. (SFC reaction too slow!)
②SFC threshold still exceeded after the Pause. (SFC under-reacted! Pause too short.)
③Under-utilization of bandwidth. (SFC over-reacted! Pause too long.)

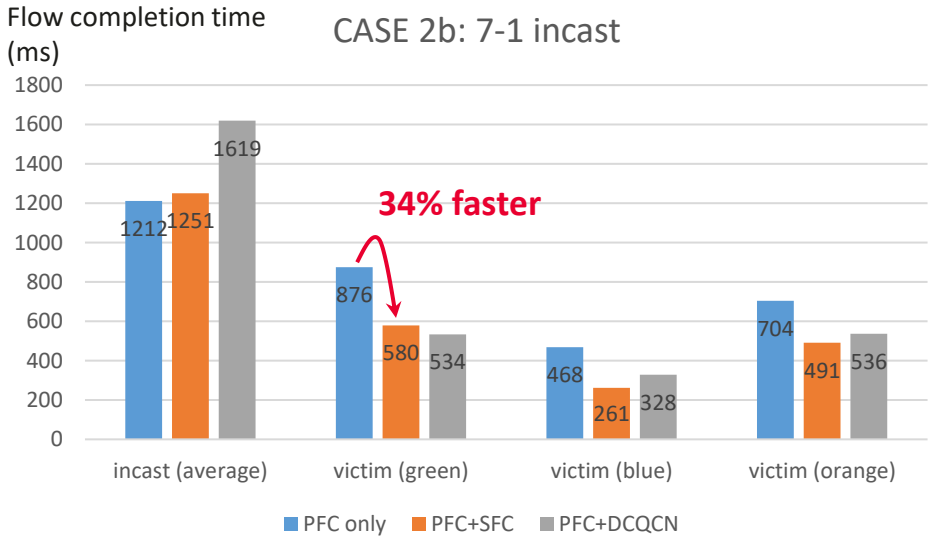
Among all three, only ③ is critical, according to the SFC robustness simulation analysis in https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-simulation-pfc-dcqcn-0925-v01.pdf

Impact of Incast

- Only add more incast flows, all other settings remain unchanged.
- As the incast degree increases, the improvement effect of SFC first increases and then decreases. It is possible that when the incast is too severe, there is insufficient space between the SFC and PFC thresholds, thus still triggering PFC.



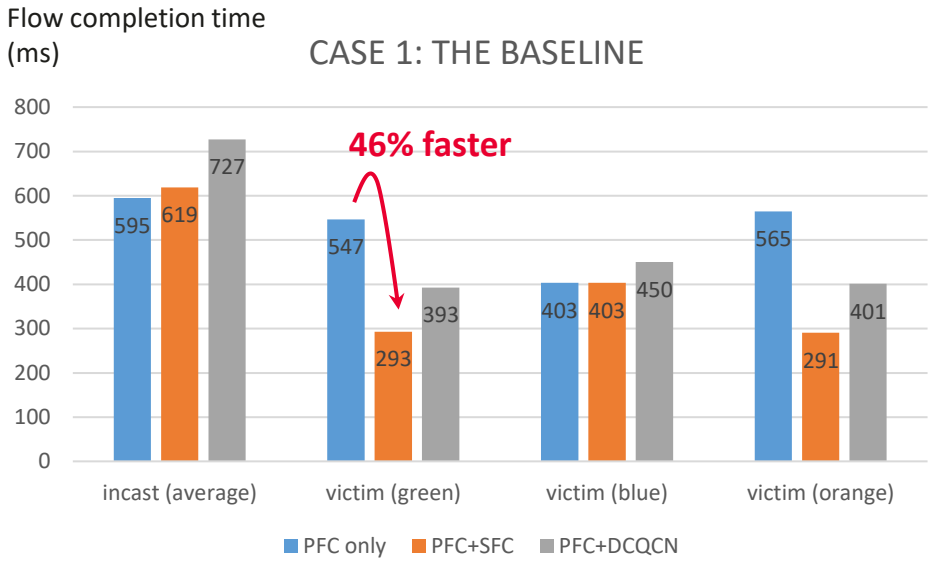


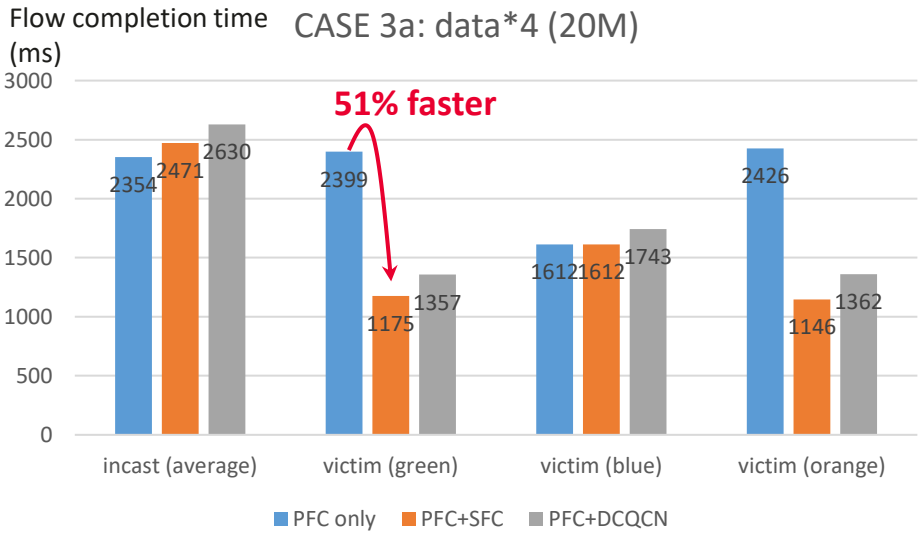


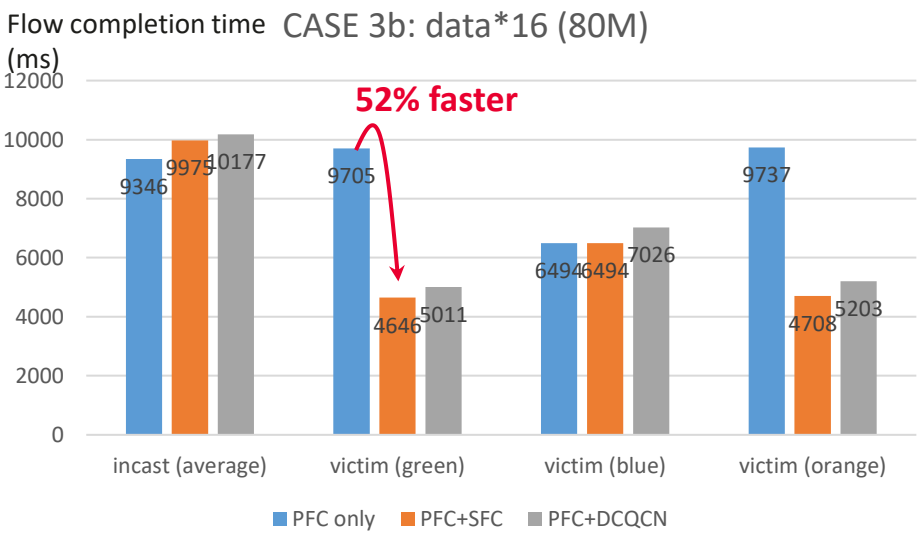
SFC can improve the performance of victim flows under different incast conditions.

Impact of data size

- Only add data size per flow, all other settings remain unchanged.
- Performance improvements when using SFC remain relatively stable.
- DCQCN performs relatively better when the data size increases (longer flow completion times), but are still not as good as SFC.



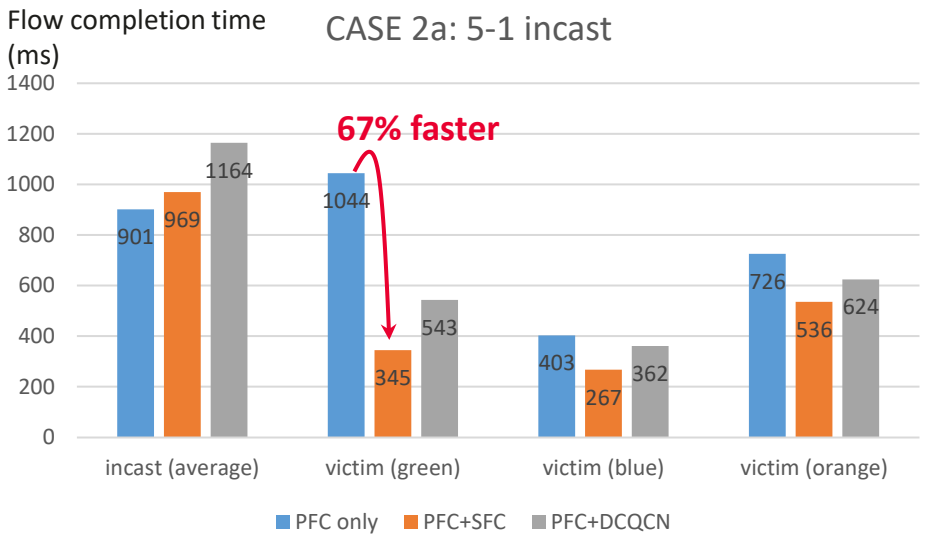


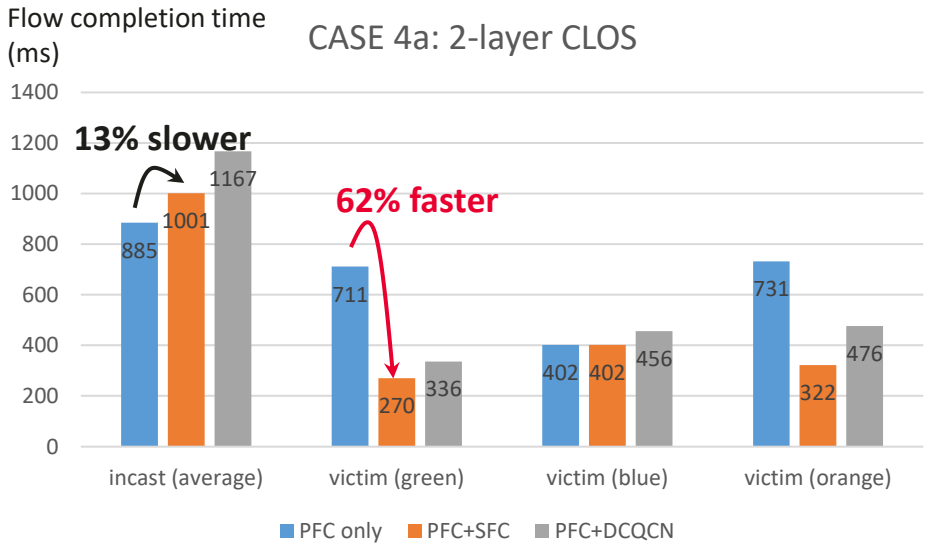


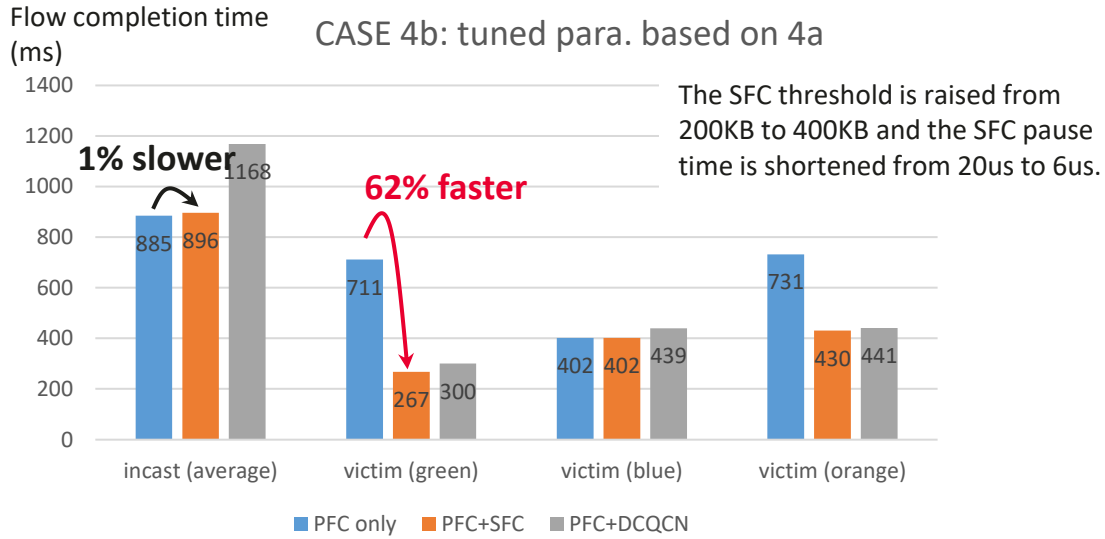
SFC can improve the performance of victim flows with different per flow data size.

3-layer vs 2-layer

- In CASE 4a, the topology is changed to a 2-layer CLOS, all other settings remain unchanged comparing to CASE 2a.
- It's hard to say whether SFC performs better in a 2-layer or 3-layer CLOS, as a set of parameter configurations might favor one over the other, making it difficult to construct a completely fair comparison.





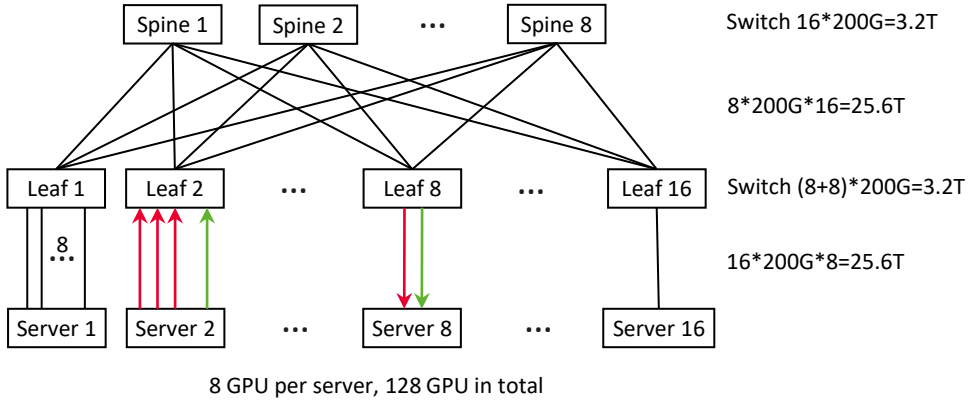


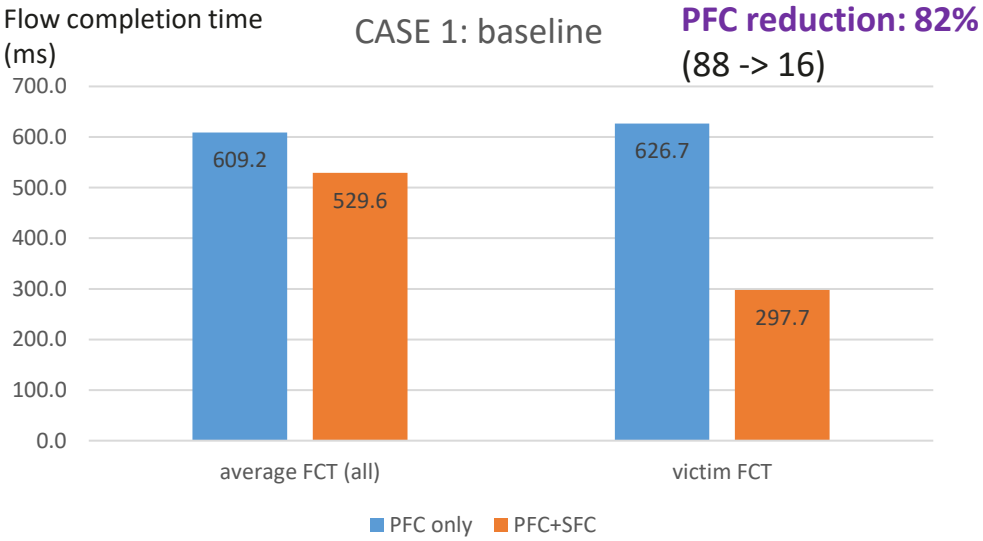
SFC can improve the performance of victim flows in both 2-layer and 3-layer CLOS topology.

Simulation Part 5: Sensitivity analysis in Layer-2 CLOS

- Basic settings (using the same topology as Simulation Part 1)
  - > Link delay: 150ns. Switch process (fixed) delay: 300ns.
  - > Buffer size: 400KB. PFC threshold: 380KB.
  - > SFC threshold: 200KB; SFC pause time: 6us; SFCM Min. interval: 6us.
  - > Flow: 3-to-1 incast and a victim, 5MB data per flow, packet size 4KB.

- PFC headroom > (150ns*2+300ns)*200Gbps/8 = 15KB
  - > Set PFC threshold = 380KB
- ①SFC headroom (i.e., from SFC threshold to PFC threshold) > (N-1)*(150ns*3*2+300ns*5)*200Gbps/8 = (N-1)*60KB
  - > if N=3 (i.e., 3-to-1 incast), then SFC headroom > 120KB
  - > if N=5,7, then SFC headroom > 240, 360KB accordingly
  - > Set SFC threshold = 200KB
- ②t_{SFC_pause} * 200Gbps > 120KB, then t_{SFC_pause} > 4.8us
- ③t_{SFC_pause} * 200Gbps < (120+200)KB, then t_{SFC_pause} < 12.8us
  - > Set t_{SFC_pause} = 10us, SFCM Min. interval: 10us

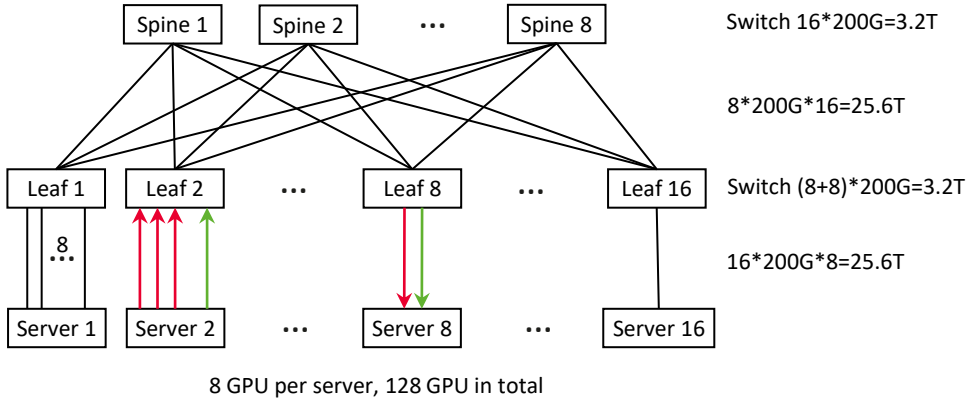


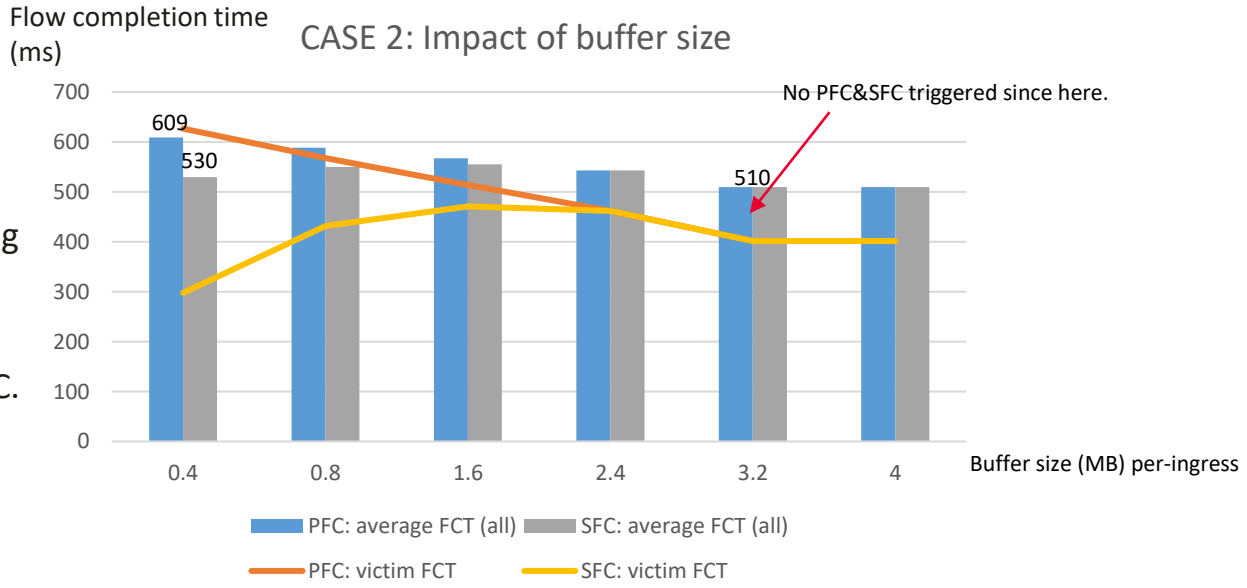


Impact of buffer size

- New case 2: same as case 1 in the previous page.

	Buffer (KB)	PFC threshold	need SFC headroom	SFC threshold	SFC pause range (us)	SFC pause time (us)
case 2	400	380	120	200	(4.8, 12.8)	10
case 2.1	800	780	120	600	(4.8, 28.8)	10
case 2.2	1600	1580	120	1400	(4.8, 60.8)	10
case 2.3	2400	2380	120	2200	(4.8, 92.8)	10
case 2.4	3200	3180	120	3000	(4.8, 124.8)	10
case 2.5	4000	3980	120	3800	(4.8, 156.8)	10





- Results and analysis:
  - > Increasing buffer size leads to shorter FCT for victim flows when using only PFC, until PFC is no longer triggered at all.
  - > If focusing on average performance, when using only PFC, increasing the buffer size by 8x yields similar improvements to just enabling SFC.

NOTE: AI-oriented data center switches generally provision tens to hundreds of KB of ingress buffer per 100G port bandwidth.

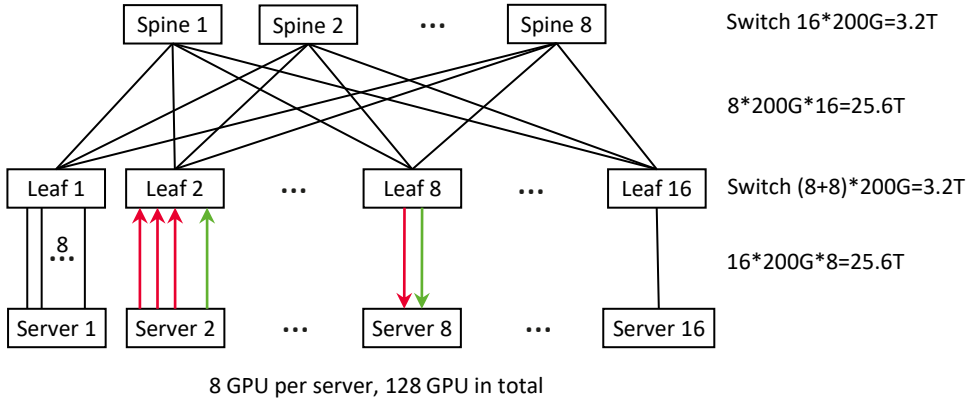
The increase in PFC thresholds (i.e., larger buffers) slightly improves overall performance.

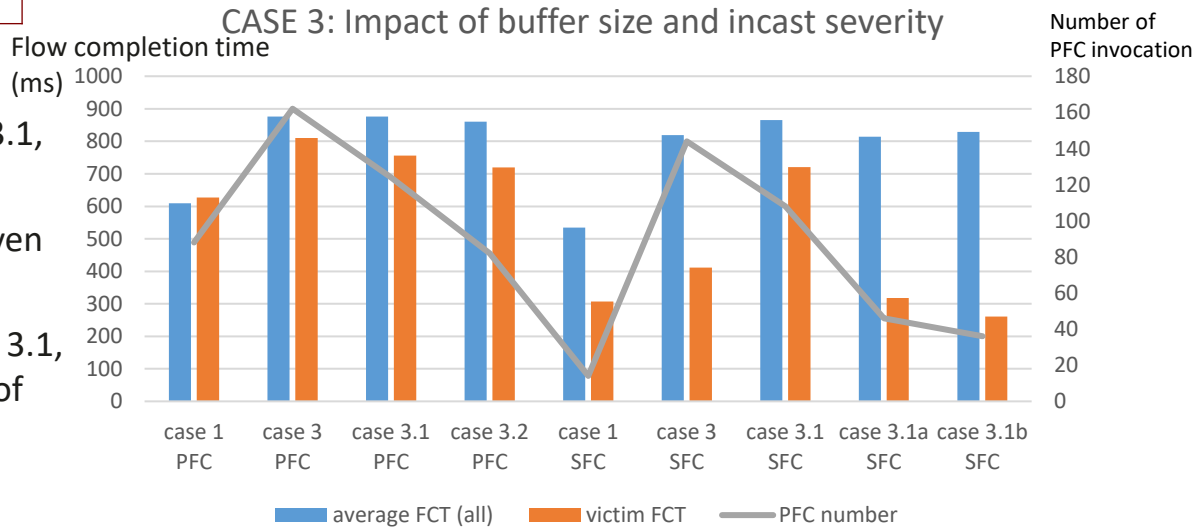
Impact of buffer size and incast severity

- Changes in case 3: 3-to-1 incast -> 5-to-1 incast

	incast	Buffer (KB)	PFC threshold	need SFC headroom	SFC threshold	SFC pause range (us)	SFC pause time (us)
case 1	3-to-1	400	380	120	200	(4.8, 12.8)	10
case 3	5-to-1	400	380	240	200	(9.6, 17.6)	10
case 3.1	5-to-1	800	780	240	600	(9.6, 33.6)	20
case 3.1a	5-to-1	800	780	240	400	(9.6, 25.6)	15
case 3.1b	5-to-1	800	780	240	200	(9.6, 17.6)	10
case 3.2	5-to-1	1200	1180	240	800	(9.6, 41.6)	20

- Results and analysis:
  - > Under a more severe incast scenario, as the buffer grows (case 3, 3.1, 3.2), using only PFC results in a slight improvement for victim FCT.
  - > Under a more severe incast scenario, the improvement of SFC is even more significant.
  - > With larger buffers, SFC configuration has more flexibility. In Cases 3.1, 3.1a, and 3.1b, as the SFC headroom becomes larger, the number of PFC invocation gradually decreases and the FCT improves.





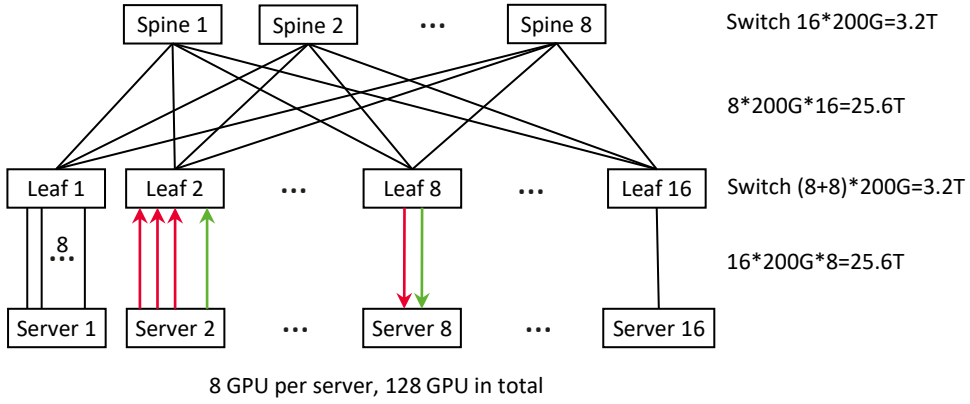
Larger buffers provide flexibility in SFC threshold settings (e.g., based on incast severity).

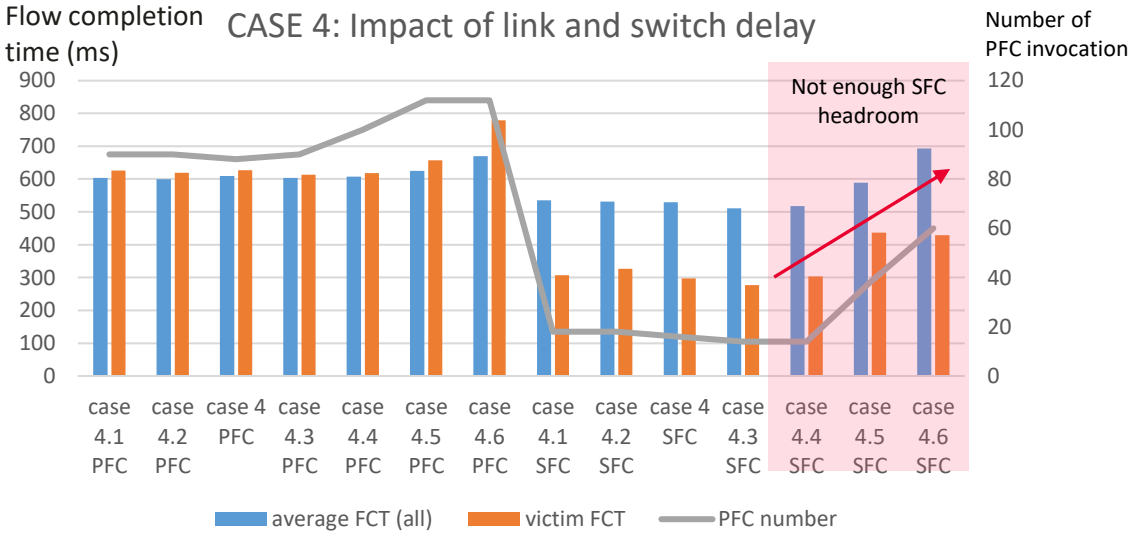
Impact of link and switch delay

- New case 4: same as case 1 in the previous page.
  - > 3-to-1 incast, total buffer: 400KB

	link delay(ns)	switch delay(ns)	PFC threshold	need SFC headroom	SFC threshold	SFC pause range (us)	SFC pause time (us)
case 4.1	50	100	380	40	200	(1.6, 13.6)	10
case 4.2	100	200	380	80	200	(3.2, 13.2)	10
case 4	150	300	380	120	200	(4.8, 12.8)	10
case 4.3	300	600	365	240	100	(9.6, 13.6)	10
case 4.4	600	1200	340	480	50	(19.2, 21.2)	20
case 4.5	1200	2400	280	960	50	(38.4, 40.4)	40
case 4.6	2400	4800	160	1920	50	(76.8, 78.8)	78

- Results and analysis:
  - > PFC threshold is tuned to maintain lossless.
  - > SFC threshold is also tuned when delay gets larger and more SFC headroom is needed.
  - > When the delay is too high and there is no enough buffer for the SFC headroom, performance degrades.





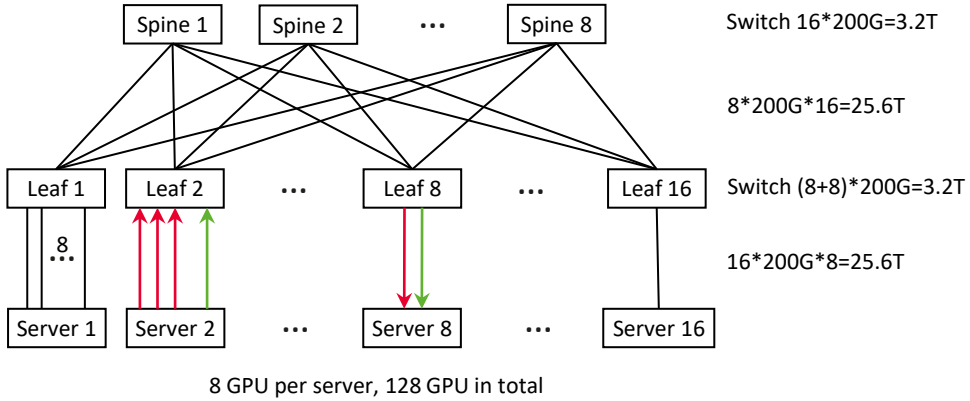
The delay has no significant impact on FCT or the improvement effects of SFC, as long as there is sufficient buffer to keep PFC and SFC configurations within reasonable ranges.

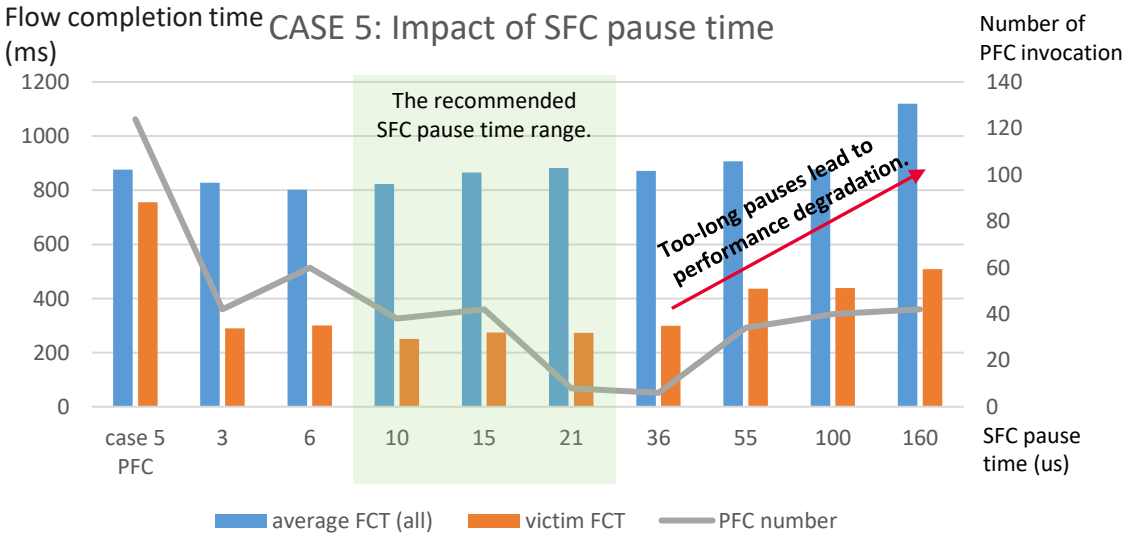
Impact of SFC pause time

- New case 5: same as case 3.1a in the previous page.

	incast	Buffer (KB)	PFC threshold	need SFC headroom	SFC threshold	SFC pause range (us)	SFC pause time (us)
case 5.1							3
case 5.2				all the same			6
case 5.3							10
case 5	5-to-1	800	780	240	400	(9.6, 25.6)	15
case 5.4							21
case 5.5				all the same			36
case 5.6							55
case 5.7							100
case 5.8							160

- Results and analysis:
  - > Too-long pauses lead to bandwidth underutilization and thus performance degradation. Too-short pauses just consume more resource without significant impact on performance.
  - > The results align with the SFC robustness simulation result in: https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-simulation-pfc-dcqcn-0925-v01.pdf





The SFC enhancement remains effective across a wide configurable range of pause durations.

Thoughts

- AI Data Center Networks: A real and significant use case for SFC.
  - > Currently all IB based AI training systems are lossless and are a scale-out network.
  - > Other known AIDC backend networks deployed in the real world currently use the lossless approach.
  - > Meanwhile, UET is focusing on the best effort approach.
  - > AI DCNs are **sensitive to throughput** and primarily involve massive numbers of **short flows** (especially in inference), making simple flow/congestion control mechanisms more suitable. Heuristic congestion control algorithms tend to underperform in terms of throughput because when one flow completes its transmission, another flow can only explore and increase its rate incrementally rather than directly preempting bandwidth.
  - > Lossless seems to be a reasonable choice for AI inference as low latency and no-retransmission is critical.
- High-Performance Computing DCNs: Another potential use case for SFC.
  - > Real HPC runs lossless. The systems have several tens of thousands of nodes, using scale-out networks.
  - > Fast (line-rate) start and minimizing PFC-induced HoLB (victim flow underutilization) are also important.
- Even so, SFC can work with both lossless and lossy (best effort) networks. Every transport can use SFC.
- SFC can be designed from One or Both of the following perspectives
  - > Keep it simple and easy-to-use. -> As the SFC baseline solution.
  - > Leverage more information from the network for more accurate control (e.g., Dynamically sets the pause time and/or the SFC threshold based on detected incast severity and buffer occupancy along the path.). -> for future extension.

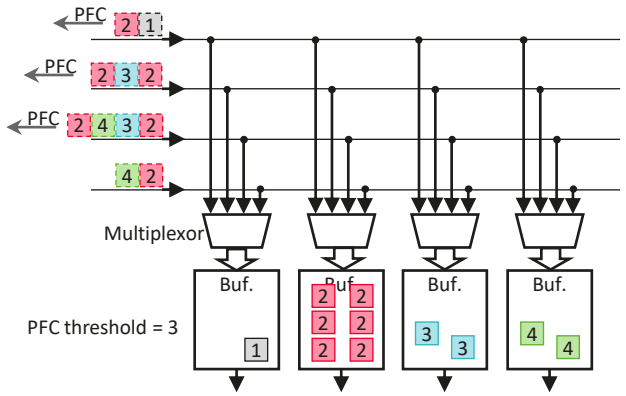
Discuss

Draft status and Proposal for the next step

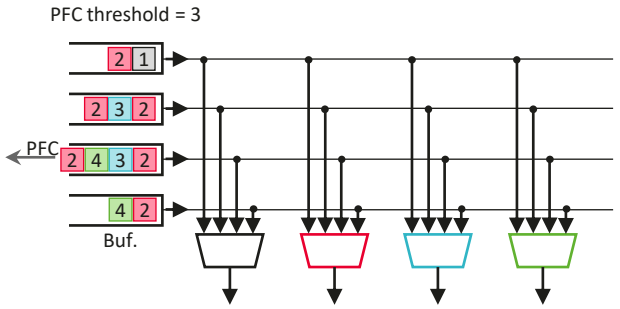
- The latest individual text and its brief can be found at https://www.ieee802.org/1/files/public/docs2025/dw-chen-individual-text-0325-v03.pdf and https://www.ieee802.org/1/files/public/docs2025/dw-chen-text-status-and-todos.pdf
- The introducing (clause 1-6) and the concept and component description (52.1-52.4) parts are almost there.
- The management part (12, 48) can be handled last.
- Clause 52.5 is the meat. For **next step**:
  - > Update and finalize a first but complete version of SFCP procedure (52.5.2).
  - > Update Encoding (52.5.3) based on what has been proposed in this contribution.
  - > Revise Variables (52.5.1) accordingly.
  - > Add Buffer requirements for SFC (Annex Y) based on the calculation given in https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-computation-simulation.pdf
- Editor appointment and take the draft to a Task Group ballot. (Nov. 2025, Mar. 2026?)

The queuing architecture's effect on PFC (Back up)

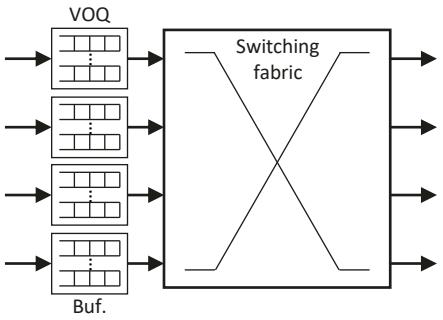
- If the reason for triggering PFC is: **multi-to-one**
  - > When OQ is used, it may trigger PFC upstream on many ingress ports (PFC spreading).
  - > When IQ is used, it may trigger less PFCs under the same PFC threshold as packets are distributed across ingress buffers.
- This might be the reason why it is said in IEEE 802.1Q 36.2.1 PFC Initiator:
  - > The PFC Initiator entity generates M_CONTROL PFC requests using the M_CONTROL.request primitive when appropriate (e.g., when **an input buffer reaches a certain threshold**).



Output Queue (OQ)



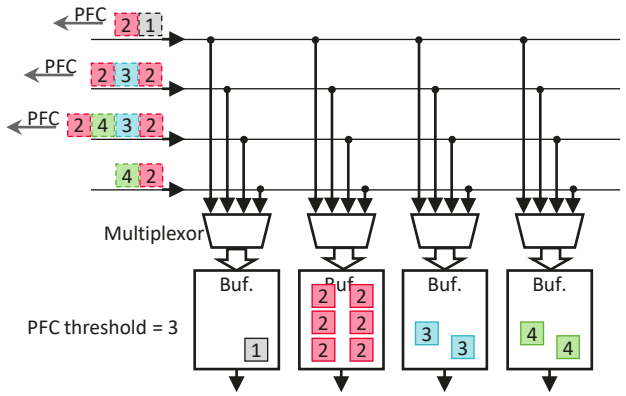
Input Queue (IQ)



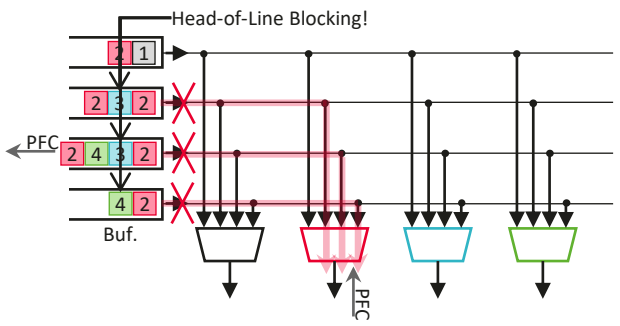
Virtual Output Queue (VOQ)

The queuing architecture's effect on PFC Cont. (Back up)

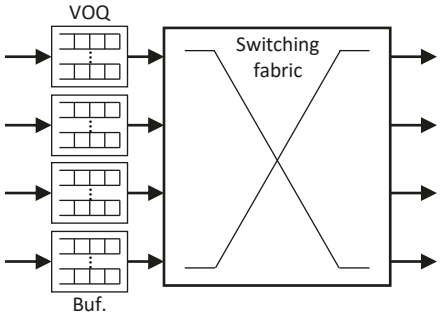
- If the reason for triggering PFC is: **receiving PFC from downstream**
  - > When OQ is used, it may trigger PFC upstream on many ingress ports (PFC spreading).
  - > When IQ is used, it may make the Head-of-Line Blocking even worse.
- If the reason for triggering PFC is: **large-to-small** (oversubscribed/bandwidth mismatch)
  - > For a one-large-ingress to one-small-egress case, OQ and IQ have the same effect. If combined with multi-to-one, see the previous page.
- In practice, PFC is preferred to be used with VOQ architecture. Although 802.1Q described a simple output queue model, there does not seem to be a strong need to modify PFC specifications.



Output Queue (OQ)



Input Queue (IQ)



Virtual Output Queue (VOQ)