SFC consideration and design

Bottorff, Paul (paul.bottorff@hpe.com)
Chen, Lihao (lihao.chen@huawei.com)
Congdon, Paul (paul.congdon@outlook.com)
Li, Jieyu (lijieyu@chinamobile.com)
Lv, Lily (lvyunping@huawei.com)
Sivakolundu, Ramesh (sramesh@cisco.com)

Data center backend networks are the primary use case for SFC

- A data center backend network is a crucial part of the data center infrastructure, specifically designed **to handle high-bandwidth, low-latency communication between high-performance computing resources like GPUs, storage, and other** critical components within the data center. **It's optimized for AI training and inference (AI) workloads, as well as other high-performance computing (HPC) tasks,** ensuring efficient data transfer and processing.

  > SFC basic concepts and comparisons to other tools (PFC, QCN, CI, …):

  - *https://www.ieee802.org/1/files/public/docs2022/new-congdon-SFC-proposal-0322-v01.pdf summarizes background, SFC concepts, and comparisons to QCN and CI.*
  - *https://www.ieee802.org/1/files/public/docs2022/new-bottorff-sfc-0322-v6.pdf describes the use cases of SFC and its comparison with DCQCN and PFC.*
  - *https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-computation-simulation.pdf explains SFC remains highly useful with AI workloads.*

  > SFC design, computation and simulations:

  - *https://www.ieee802.org/1/files/public/docs2022/new-blendin-SFC-sim-0522-v01.pdf and https://www.ieee802.org/1/files/public/docs2022/new-blendin-SFC-Simulation-Results-0722-v01.pdf show the benefits of SFC and SFC w/ Proxy compared to PFC based on HPC workloads by simulation.*
  - *https://www.ieee802.org/1/files/public/docs2022/new-congdon-SFC-design-topics-0622-v01.pdf gives 11 design topics.*
  - *https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-proposal-0125-v01.pdf provides proposals on drafting the text and updated the individual text https://www.ieee802.org/1/files/public/docs2025/dw-chen-individual-text-0325-v03.pdf based on previous contributions.*
  - *https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-computation-simulation.pdf describes how to compute SFC parameters and simulations based on AI workloads.*

  > Note: not all previous contributions are listed here.

3 types of congestions that may trigger flow control

- Incast congestion
  - > PFC: congestion spreading.
  - > SFC: preferred solution.
- In-network congestion
  - > Load balancing (packet spray, adaptive routing, etc.): preferred solution.
  - > ECN: no lossless guarantee.
  - > SFC: an option.
- Outcast congestion
  - > PFC: preferred solution.
  - > SFC: equal to PFC.

- Packet trimming BTS (back-to-sender):
  - > No lossless guarantee, triggered when packets are dropped due to buffer overflow.

SFC goals and Clarifications

- For lossless networks
  > First SFC to avoid PFC spreading
  > Then PFC for lossless
- For best effort networks
  > First SFC to avoid PFC spreading
  > Then packet trimming for fast retransmission

Notes:

1. PFC spreading can be a severe issue as the radix (number of ports per switch) is very high in data center backend networks.
2. AIDC workloads are highly dynamic, characterized by ON/OFF traffic patterns and pervasive incast.
3. Packet trimming BTS (back-to-sender) is a drop notification that the dropped packet is trimmed and directly sent back to the sender.

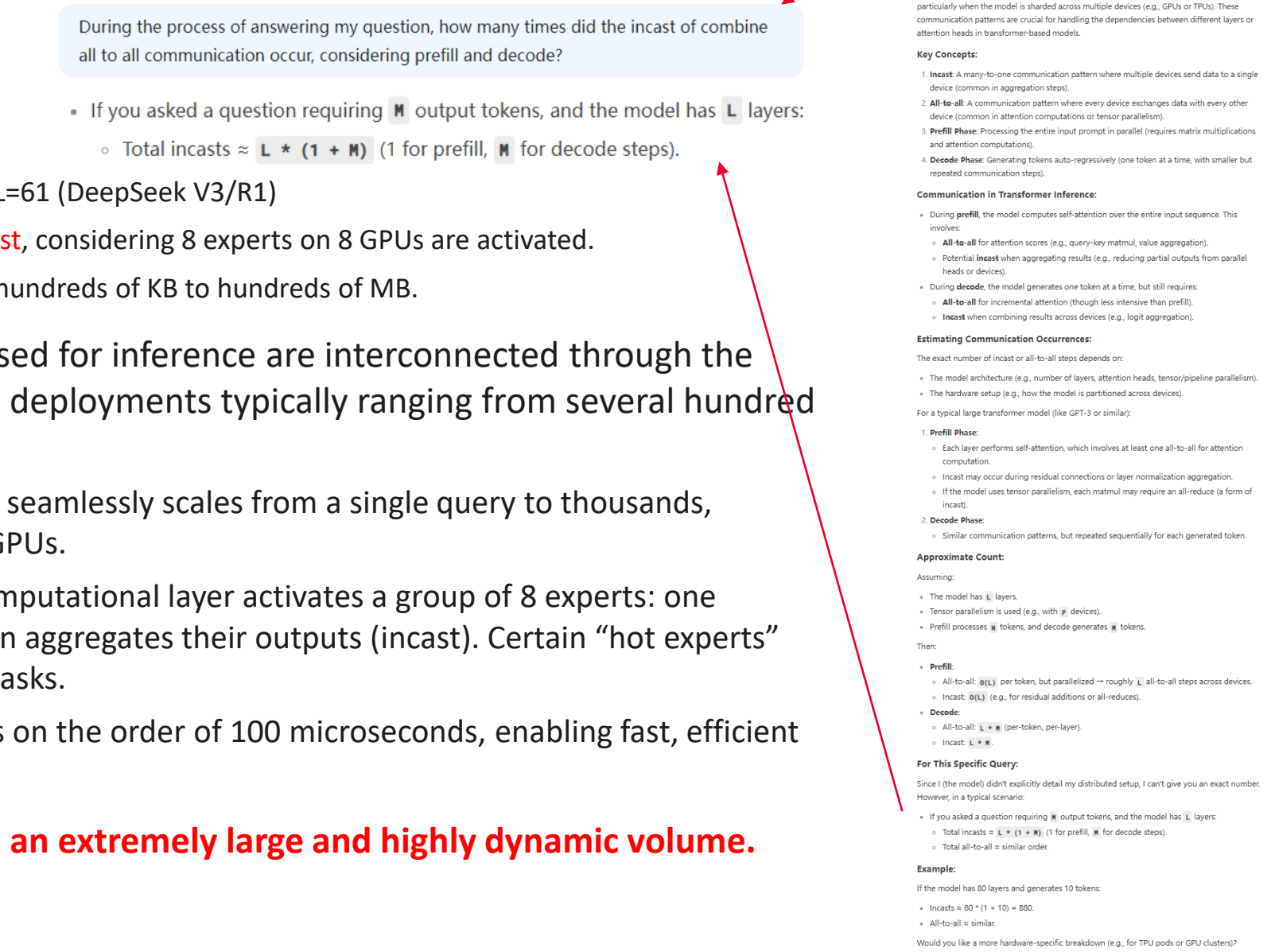
Draft status and Proposal for the next step

- 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
- Then, **take the draft to a Task Group ballot**. (Sep.2025?)

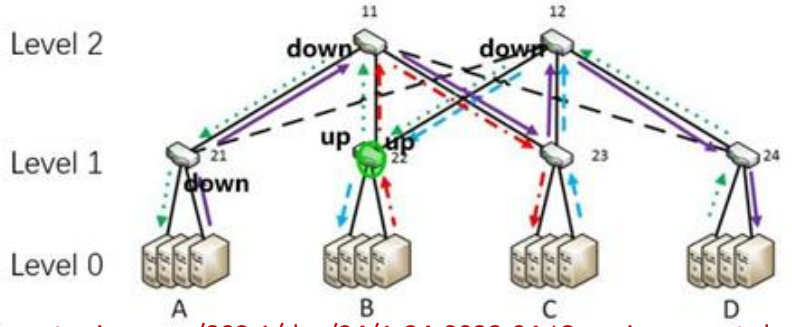
Discussion

An example to feel the incast in AIDC

- DeepSeek answering my question:
  > M=492 (492 output tokens for this answer), L=61 (DeepSeek V3/R1)
  > Causing more than 30,000 seven-to-one incast, considering 8 experts on 8 GPUs are activated.
  > Typical data size for one incast ranging from hundreds of KB to hundreds of MB.
- The AI computing modules (GPUs) used for inference are interconnected through the data center's backend network, with deployments typically ranging from several hundred to several thousand units.
  > - Scalable Task Distribution: The system seamlessly scales from a single query to thousands, dispatching each task across available GPUs.
  > - Layered Expert Coordination: Each computational layer activates a group of 8 experts: one dispatches data to the other seven, then aggregates their outputs (incast). Certain "hot experts" may receive a higher concentration of tasks.
  > - Rapid Execution: Each layer completes on the order of 100 microseconds, enabling fast, efficient responses.
- **Incast is constantly happening, with an extremely large and highly dynamic volume.**

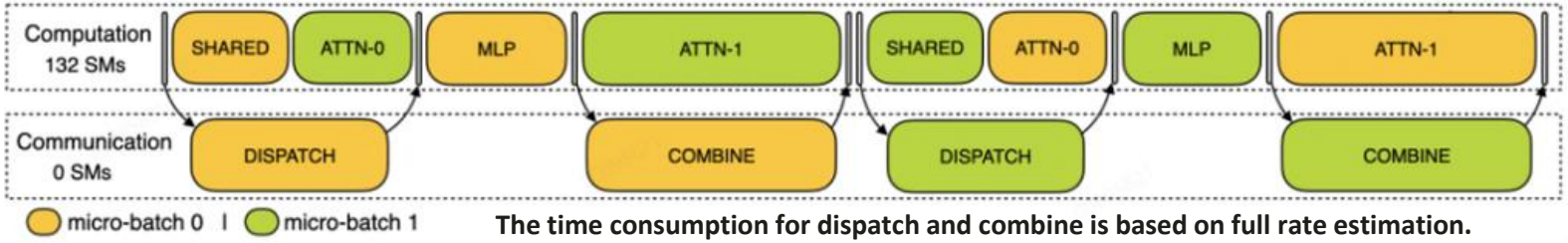


AIDC network is quite homogeneous



https://mentor.ieee.org/802.1/dcn/24/1-24-0028-04-ICne-aicn-report-draft.pdf

- **Networking (Topology, bandwidth, latency):**
  - > 2 or 3-layer CLOS/fat-tree is the main topology, interconnecting thousands or tens of thousands of GPUs.
  - > High bandwidth (e.g., 400G ports), high radix (e.g., 128 ports per switch) and ultra-low latency (down to 250ns switch delay[1]).
- **Collective communication:** Whether training or inference and regardless of parallelism method, all jobs rely on collective operations like AllReduce and AllToAll.
- **Overlap design:** Maximize the overlap between computation time and communication time to optimize Job Completion Time (JCT).



The time consumption for dispatch and combine is based on full rate estimation.

- Therefore, simulation and verification cases in https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-computation-simulation.pdf employ representative 2-layer CLOS combined with AllReduce and AlltoAll workloads.

- *__Capital intensive__: In 2024, global DC capital expenditure (CAPEX) reached $450B due to a surge in AI computing power demand, with network infrastructure (including high-speed interconnects, optical modules, switching equipment, etc.) accounting for approximately 15-20%.

1. https://investors.broadcom.com/news-releases/news-release-details/broadcom-ships-tomahawk-ultra-reimagining-ethernet-switch-hpc

SFC Design Principles: easy to use

- Ideally, SFC should be almost as simple and easy to use as PFC.

- Step 1: preconfigure SFC threshold and pause time (informative on the perspective of SFC standard)
  > Option a) Know your topology, switch capabilities, and the AI application you run (relates to the level of incast), then calculate as in https://www.ieee802.org/1/files/public/docs2025/dw-chen-sfc-computation-simulation.pdf
  > Option b) Using empirical values summarized from typical scenarios
- Step 2: enable SFC on switches. Enable proxy mode if end-stations do not support SFC.

- NOTE: SFC trigger-to-activation latency is very low with no significant jitter in data center backend networks.
  > Max. 2 hops (2-layer CLOS) or 4 hops (3-layer CLOS).
  > Switch (fixed) delay down to 250ns per hop, queuing delay for SFCM at high priority is negligible.
  > Plus the processing time related to SFC procedures.

SFCM format design

- General idea: Design a unified format that can be used in both AI and HPC data centers.
- What is needed (mandatory):
  > Source address of the congesting frame as SFCM's destination address
  > Indication of the class to be paused as well as the pause time.
- Encapsulation options: SFCM PDU in Ethernet (EtherType 0x89A2) or UDP IPv4 or UDP IPv6.
  > Including the address.
- SFCM PDU: Version + Reserved + TLV. For example,
  > Value of TLV Type A: referencing the PFC format in a standard PFC frame that includes pause time associated with the class.
  > Value of TLV Type B: some first bytes of the congestion frame's MSDU and an explicit pause time.