Application-aware Data Center Network (APDN) Use Cases and Requirements

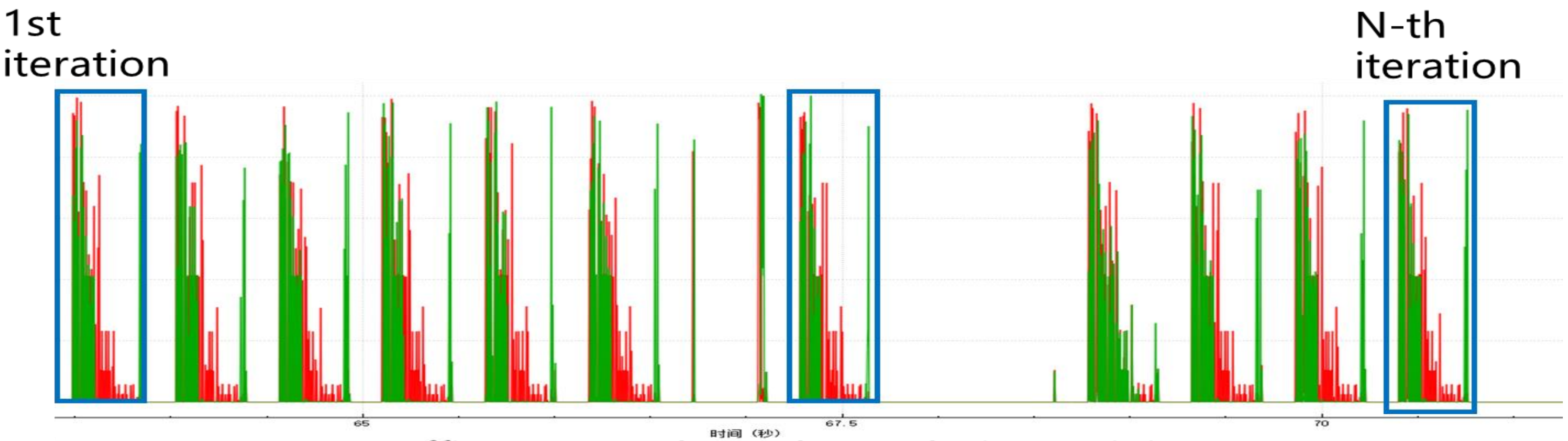
https://datatracker.ietf.org/doc/draft-wh-rtgwg-application-aware-dc-network/

Hiabo Wang(Huawei), Kehan Yao(China Mobile), Wei Pan(Huawei), **Hongyi Huang (Huawei)**

RTGWG, 119, Brisbane

Background: Distributed AI Model Training in Data Center

- Large AI models(e.g., Large Language Models) require distributed training among thousands of accelerators(e.g., GPUs), which generates **synchronized, periodic and large (>>X GB) flows** to exchange intermediate results(i.e., update gradients) between accelerators before a next iteration.



Traffic pattern(Throughput) during training

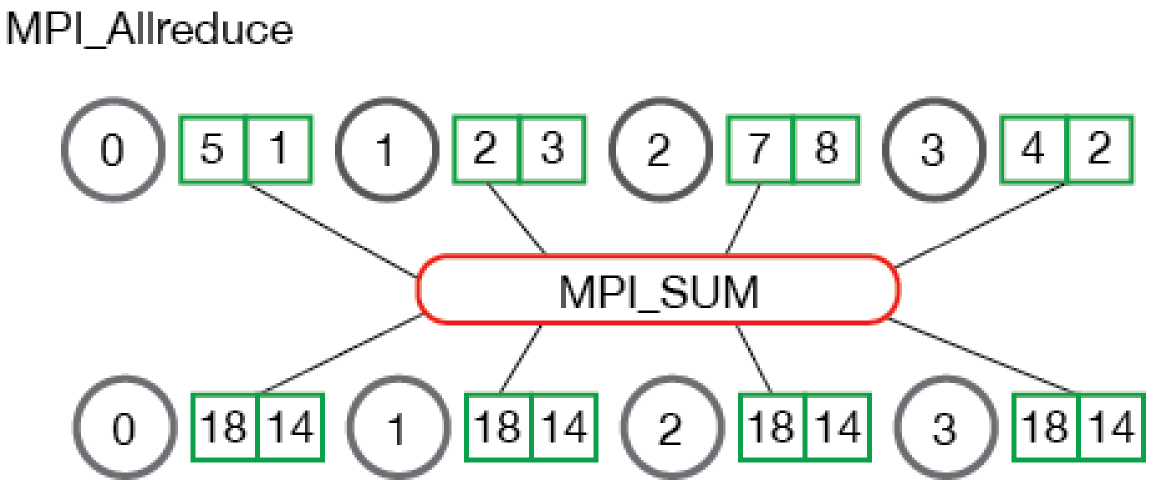
== > **New mechanisms are proposed** to serve AI training applications to meet network bottlenecks

Insights:

1. Better **collaboration between the network and the host side** is required to improve application efficiency

2. **Information** for collaboration **needs to be carried** in packets sent from the end to the network or in the reverse direction.

UC1: Enhancing Distributed Machine Learning Training with In-Network Computing

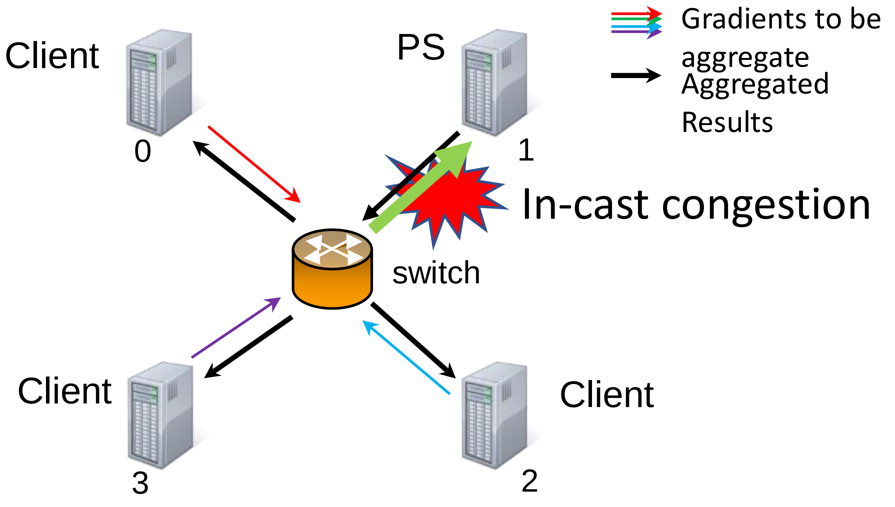


- As one kind of MPI(Message Passing Interface), AllReduce is widely used in each iteration during the process of distributed AI model training.

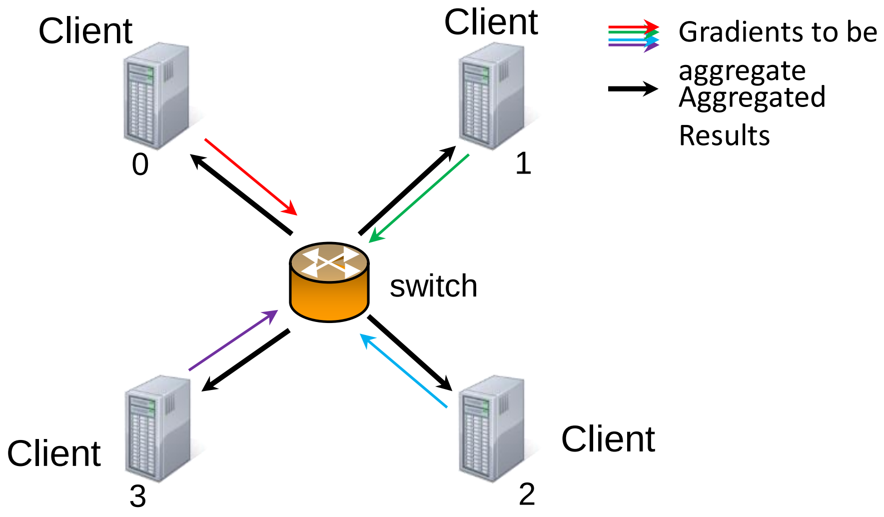
- AllReduce will reduce the values from multi-processors (MPI_SUM) and distribute the results to all processors finally.

UC1: Enhancing Distributed Machine Learning Training with In-Network Computing

AS-IS



TO-BE



Under the **Parameter Server(PS)** architecture, when multiple clients send a large amount of gradient data to the same server simultaneously for reducing, it is prone to induce in-cast (many-to-one) congestion from the perspective of server.

In-network computing (INC) offloads the behavior of the server (MPI_SUM) to the switch capable of line-rate processing, eliminating the in-cast congestion.

UC1: Enhancing Distributed Machine Learning Training with In-Network Computing

- **Observation:** To enable INC in the switches, the host needs to expose **the intent** of the application to be calculated and **the content** to be calculated

- Current implementations (e.g., ATP[atp], NetReduce[netreduce]) require the switches
  - to parse upper-layer protocol
  - and understand application-specific logic that is dedicated to certain application

- **Gap analysis:**
  - Different transports/applications
  - Difficult to fetch INC info with encryption of app content applied

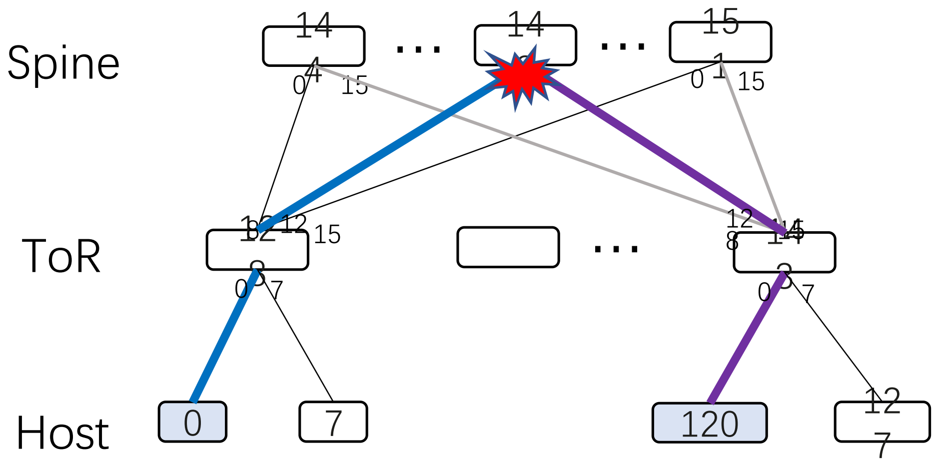
UC1: Enhancing Distributed Machine Learning Training with In-Network Computing

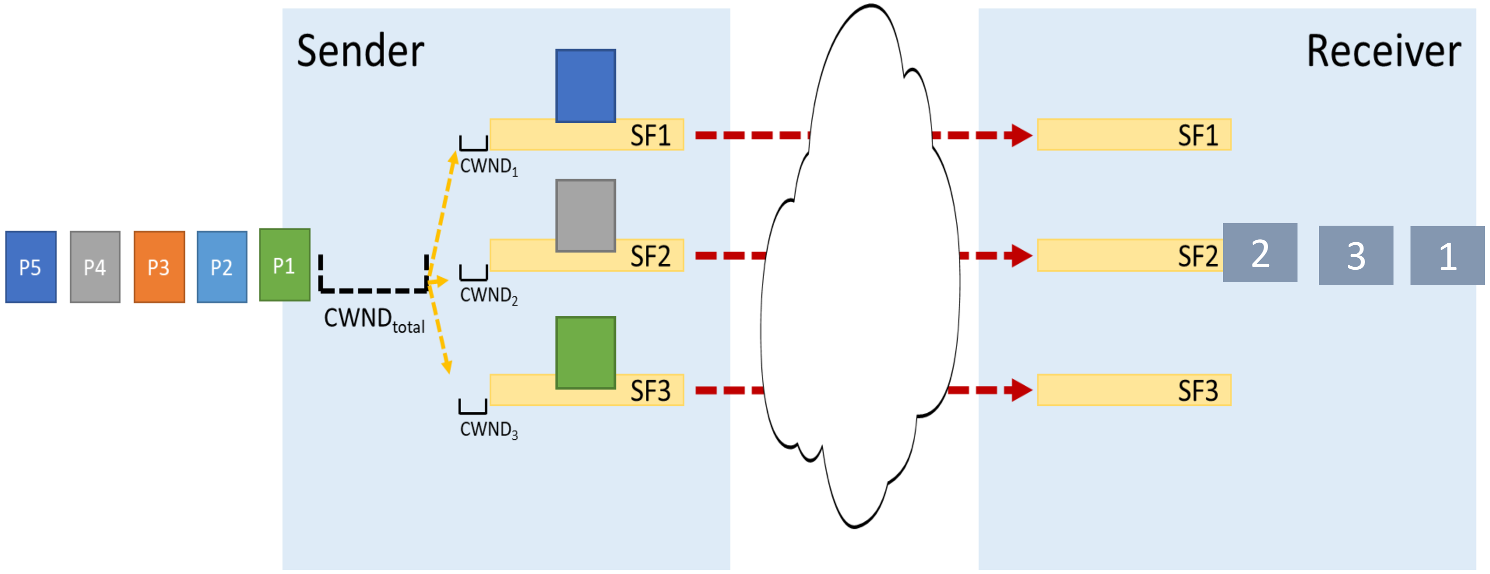
- **Potential Solution:**
  - APN can be explored to transmit information about the requested INC operations and information about the corresponding data segments to perform INC.

- **APDN Requirements:**

1. APN MUST carry application identifier to distinguish different INC tasks.
2. APN MUST support to carry various formats and length of application data (such as gradients in this use case) to apply INC and the expected INC operations.
3. APN SHOULD be able to carry other application-aware information, not compromising the reliability of end-to-end transport.
4. APN MUST be able to carry complete INC results and record the computation status in the data packets.

UC2: Fine-grained packet scheduling for load balancing





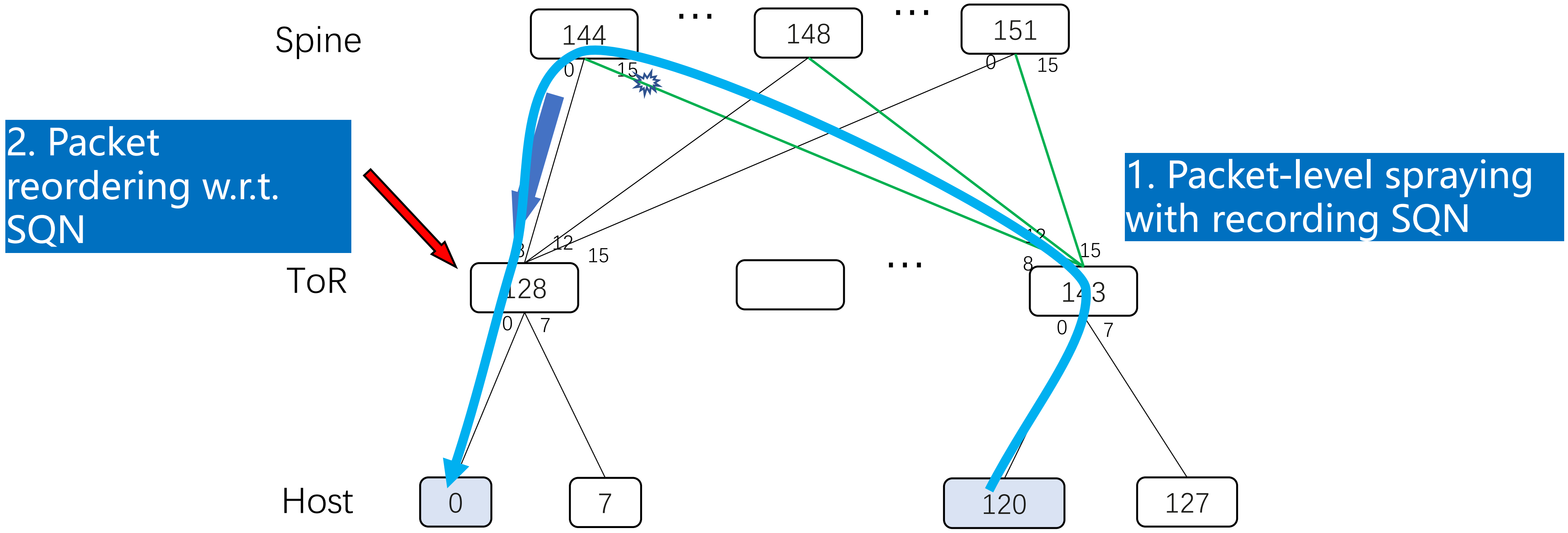
Traditional per-flow ECMP: Synchronized large flows may be distributed to the same path/switches and incur congestion.

➔**Fine-grained per-packet ECMP**: leading to packet disorder due to multi-pathing.

➔Experiments[1] show that disorder can greatly decrease the performance of applications.

[1] https://www.linkedin.com/pulse/spray-validation-dmitry-shokarev/

UC2: Fine-grained packet scheduling for load balancing



- One of the solutions is to re-order the packets in order in the egress switch/NICs, without modifying and affecting the end-host transports.

UC2: Fine-grained packet scheduling for load balancing

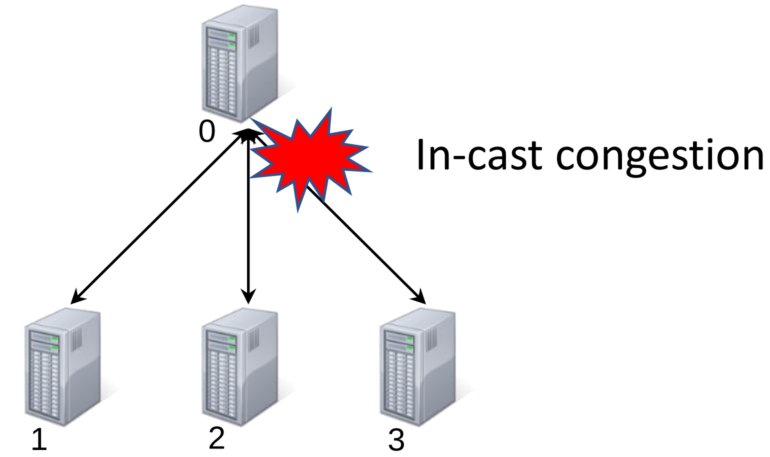
- **Observation:** The sequence of packets needs to be carried on different host &/ network devices along with the data packets.

- **Gap Analysis:** Only Transport/application carries sequence number of packets.
  - Difference transports/applications.
  - Each transport/application (flow) requires separate reordering queue.
  - Transport SQN is not supposed to be modified by the network.

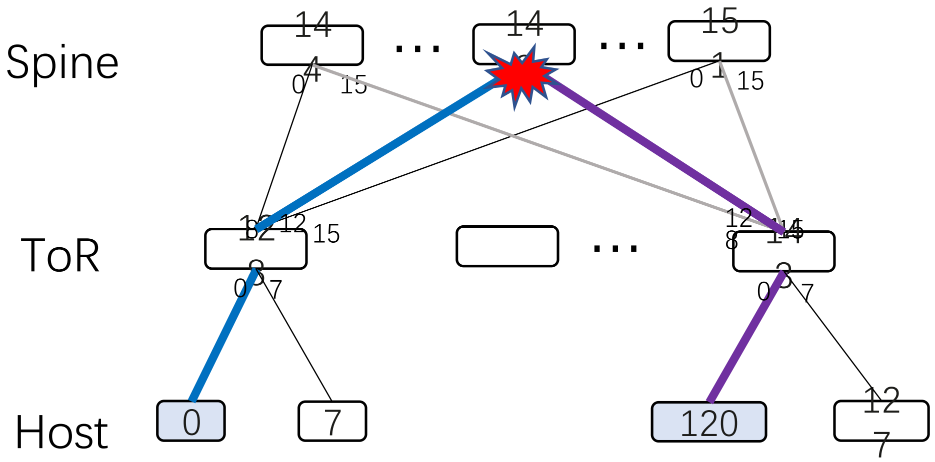
- **APDN Requirements:**
  1. APN SHOULD encapsulate each packet with SQN besides APN ID for reordering.
  2. The SQN in APN MUST NOT be modified inside the multi-pathing domain and could be cleared from APN at the egress device.
  3. APN SHOULD be able to carry necessary queue information (i.e., the sorting queue ID) usable for fine-grained reordering process.

UC3: Enhanced Congestion Control with Precise Feedback Mechanisms

- AI Data center requires congestion control, due to
  - In-cast congestion (e.g., AllReduce, All2All traffic)



  - Unbalanced load distribution among paths



UC3: Enhanced Congestion Control with Precise Feedback Mechanisms

- ECN [RFC3168] uses only 1-bit marker unable to transmit more congestion information.
- Newly CC algorithms (e.g., AECN[draft-shi-ccwg-advanced-ecn]) encourages to collect/update the congestion info hop by hop to help locate the congestion point and support fine-grained control.

```
              pkt+                     pkt+                     pkt+
         AECN Command+            AECN Command+            AECN Command+
+------+Congestion Info0+-------+Congestion Info1+-------+Congestion Info2+--------+
|Sender|===============>|Router1|===============>|Router2|===============>|Receiver|
+------+     Link-1     +-------+     Link-2     +-------+     Link-3     +--------+
  /|\                                                                          |
   |                                                                           |
   +---------------------------------------------------------------------------+
                                        ACKs
```

UC3: Enhanced Congestion Control with Precise Feedback Mechanisms

- **Observation:** The application specifies the required network information in the packet and the information is returned to the application along with the data packet

- **Potential solution:** APN for application side can be used to carry application information to determine the type of information to be collected.

- **Requirements:**
    1. APN MUST allow the data sender to express its intention about which measurement it wants to collect and the condition when it should be recorded or updated.
    2. APN MUST allow network nodes to record/update necessary measurement results, if the nodes decide to do so, and transmit these results to the data receiver.

Next Steps

- Solicit comments and refine drafts.
- Welcome cooperation.
- More details of Use Case Sharing: APN Side Meeting [Thursday]

Thank you