IETF 118 – Side Meeting – Collective Communication Optimizations

Signaling In-Network Computing operations (SINC)

draft-lou-rtgwg-sinc-00

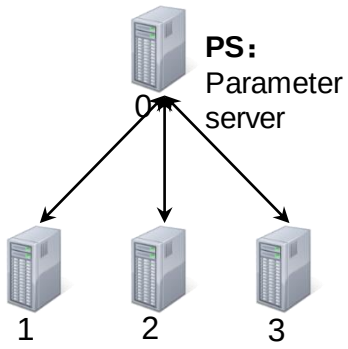
David Lou, Luigi Iannone, Yizhou Li, Cuimin Zhang, Kehan Yao

Motivation

- Recent research has shown that **offloading some computing tasks to the network** can greatly improve the overall application and system performance in some scenarios
- Their implementation is mainly based on the **programmable** network devices by using P4 or other similar languages.
- For large networks with complex network topologies, **traffic steering** is required to forward/route packets to the programmable network devices.
- We argue that it is necessary to **provide an explicit and general way** in the data/control planes **to signal the in network computation.**

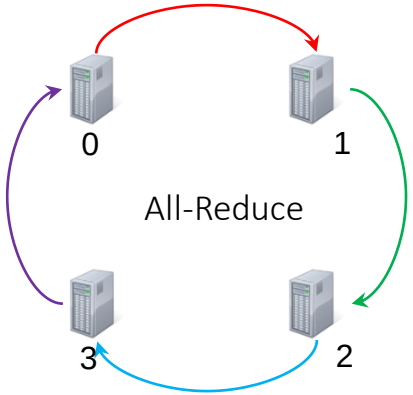
SINC Scenario – Collective Communications

Traditional way:



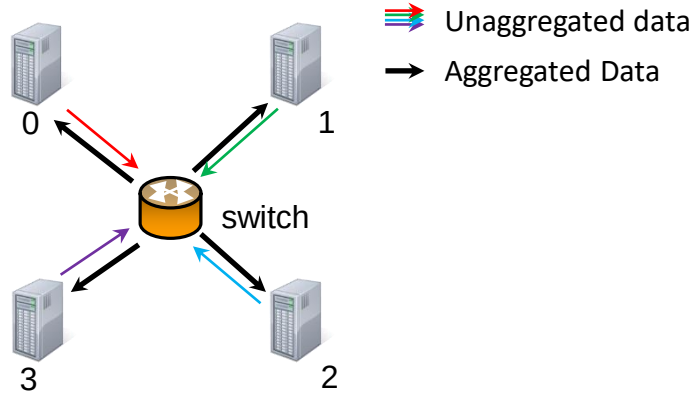
- ❖ PS aggregate the gradients, thus PS will suffer from in-cast issue.
- ❖ PS can easily become a bottleneck

Reduce Ring:



- ❖ The overall data transfer is increasing;
- ❖ The communication pattern involved may lead to higher network latency

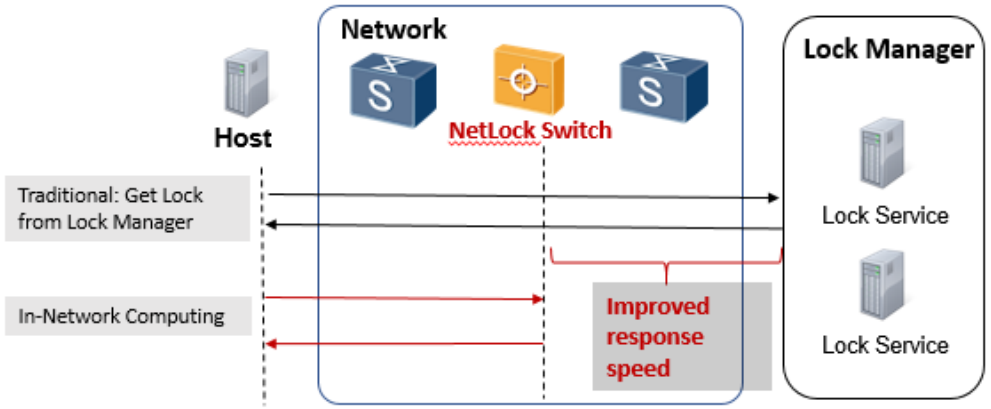
NetReduce:



- ❖ Comparing with the host oriented solutions, in-network aggregation could potentially reduce nearly half the aggregation data
- ❖ NetReduce is >1.5x faster, and has better scalability than ring all-reduce.

SINC Scenario – Consistency

NetLock:

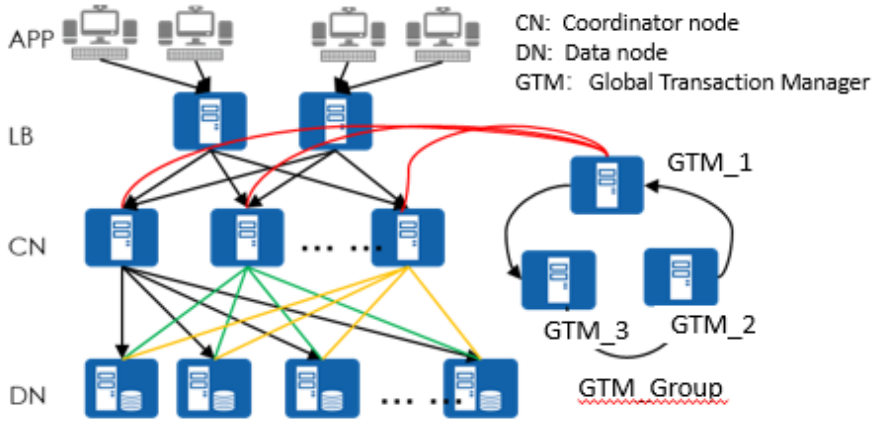


❖ For SINC :

The lock manager can be abstracted as **Compare And Swap (CAS) or Fetch Add (FA) operations**.

The test results in NetLock[1] show that the lock manager running on a switch is able to answer 100 million requests per second, **nearly 10 times more than** what a lock server can do.

NetSequencer:



❖ For SINC:

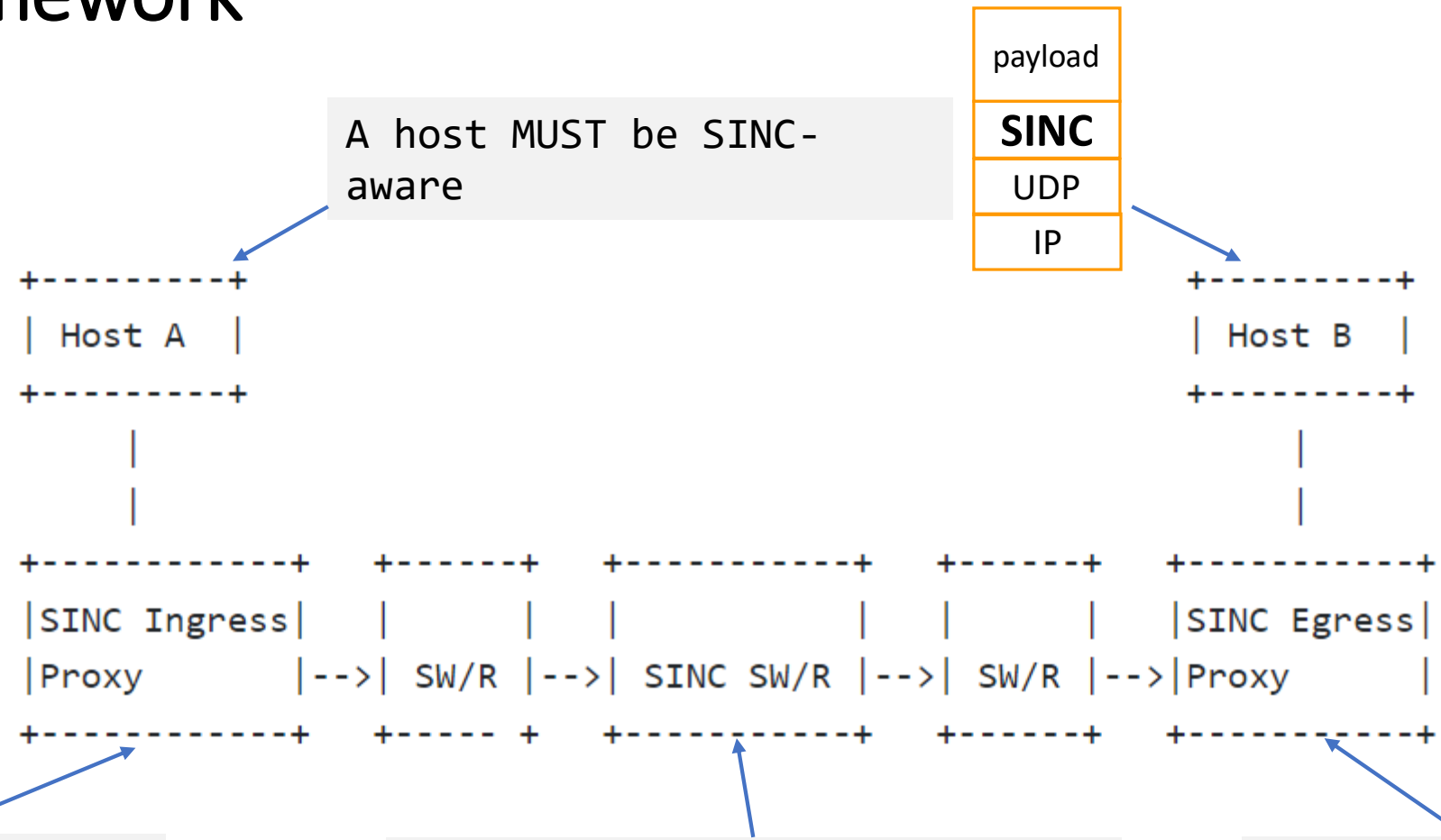
Switches could realize the sequencer[2] by using a **"Fetch-and-Add" operation**.

Compared with Gbps-level throughput of servers, network devices have **Tbps-level throughput** and **line-rate processing capabilities**

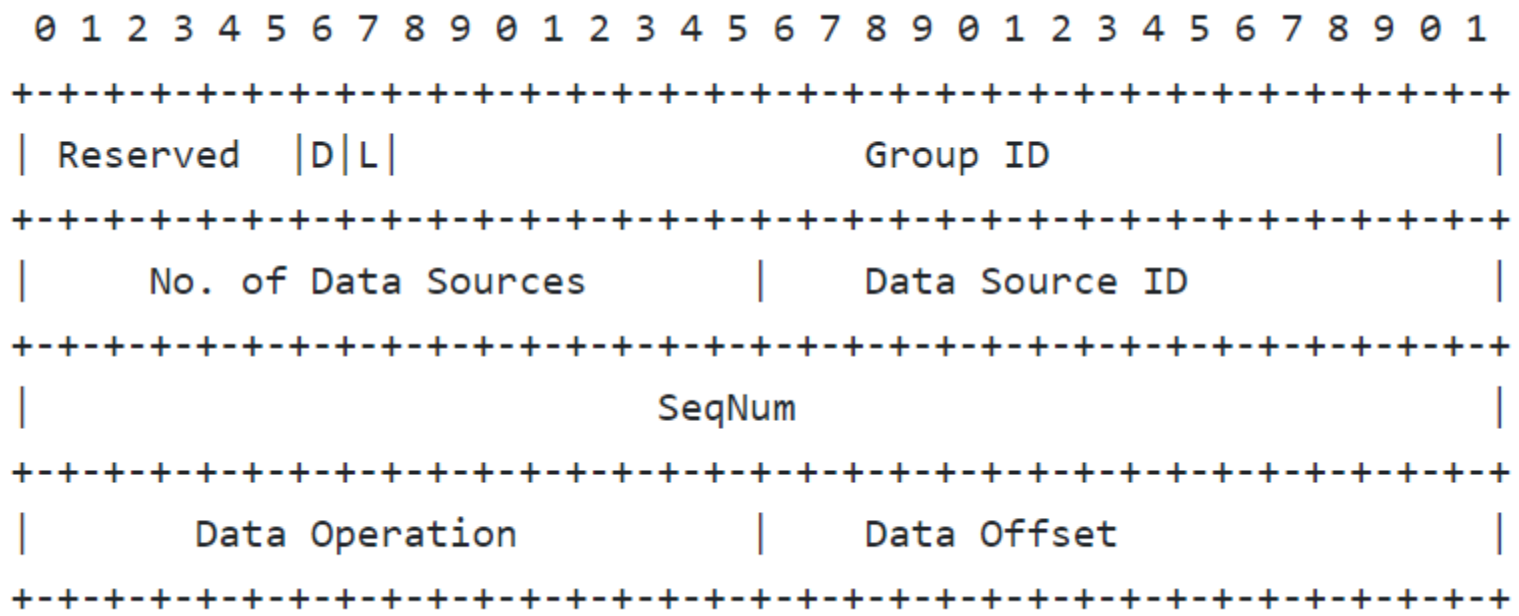
[1] Yu Z, Zhang Y, Braverman V, et al. Netlock: Fast, centralized lock management using programmable switches[C]//Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication. 2020: 126-138.

[2] Design Guidelines for High Performance RDMA Systems, https://www.usenix.org/conference/atc16/technical-sessions/presentation/kalia

SINC Framework



SINC Header



- **Loopback flag (L):** Zero (0) -> be sent to the destination; One (1) -> be sent back to the source node.
- **Done flag (D):** Zero (0) -> the request operation is not yet performed; One (1) -> the operation has been done.
- **Group ID:** Identifies different groups
- **Number of Data Sources:** Total number of data source nodes that are part of the group.
- **Data Source ID:** Unique identifier of the data source node of the packet.
- **Sequence Number (SeqNum):** The SeqNum is used to identify different requests within one group.
- **Data Operation:** The operation to be performed, like ADD, SUM, MAX, MIN
- **Data Offset:** The in-packet offset from the SINC context header to the data required by the operation.

Control Plane Considerations

- Topology creation/deletion
  - Topology computation: When receiving the computing request from the Host, the SINC control plane needs to compute a set of feasible paths with SINC capable nodes to support individual or batch computations.
  - Topology establishment: The topology has to be sent/configured/signaled to the network device, so SINC packets could pass through the right SINC capable nodes to perform the required data computing in the network.
  - Topology deletion: Once the application finishes the action, it will inform the control plane to delete the topology and release the reserved resources for other applications and purposes.
- Resource (e.g. computing resource, buffer resource, and bandwidth resource) reservation to avoid traffic and computing congestion
- Performance monitoring to detect any potential issues during the data operation
- Service protection
  - The in-network computing service must be protected. If a SINC node of an in-network operation fails, the impact should be minimized by guaranteeing as much as possible that the packets are at least delivered to the end node, which will perform the requested operation. The control plane will take care to recover the failure, possibly using a different SINC node and re-routing the traffic.
  - The network service must be able to deliver packet to the designated SINC nodes even in case of partial network failures (e.g. link failures).

Control Plane Requirements

- The ability to exchange computing requirements (e.g. computing tasks, performance, bandwidth, etc.) and execution status with the application (e.g., via a User Network Interface). SINC tasks should be carefully coordinated with (other) host tasks.
- The ability to gather the resources available on SINC-capable devices, which demands regular advertisement of node capabilities and link resources to other network nodes or to network controller(s).
- The ability to dynamically create, modify and delete computing network topologies based on application requests and according to defined constraints.
- The ability to monitor the performance of SINC nodes and link status to ensure that they meet the requirements.
- The ability to provide failover mechanism in order to handle errors and failures, and improves the resilience of the system. A fallback mechanism is required in case that in-network resources are not sufficient for processing SINC tasks, in which case, end host might provide some complementary computing capabilities.

In-Network Operations and Data

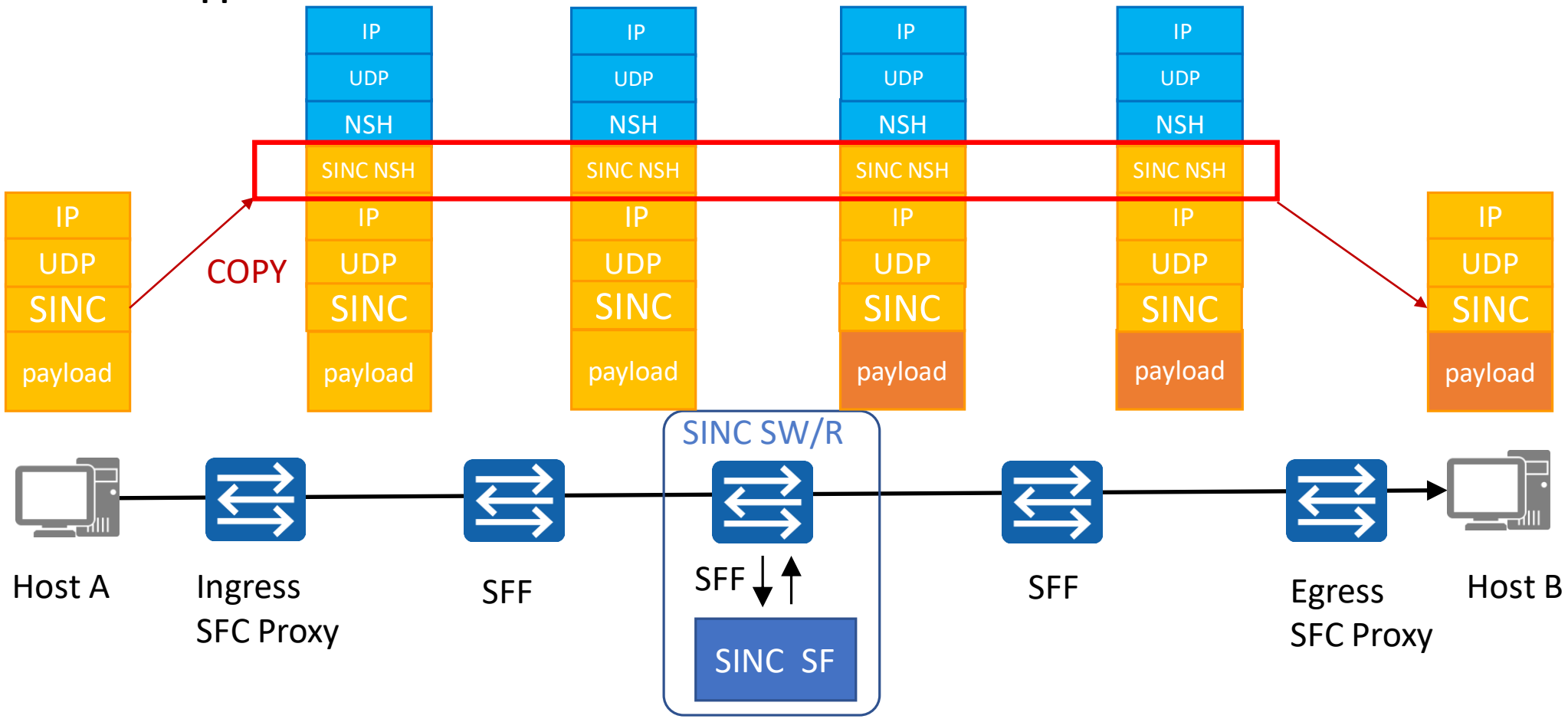
- The core idea of SINC is to offload “bottleneck” computing operations to the network devices in order to improve the system performance
- The network devices executing computing operations should not affect the forwarding performance of data plane
- Generic "simple and basic" operators are desired to support different scenarios
- An explicit and general mechanism is required to tell the switch what, where and how

Use Case	Operation	Description
NetReduce	Sum value (SUM)	The network device sums the collected parameters together and outputs the result
NetLock	Compare And Swap or Fetch-and-Add (CAS or FA)	By comparing the request value with the status of its own lock, the network device sends out whether the host has the acquired lock. Through the CAS and FA, host can implement shared and exclusive locks.
NetSequencer	Fetch-and-Add (FA)	The network device offers a counter service and provides a monotonically increasing sequence number for the host.

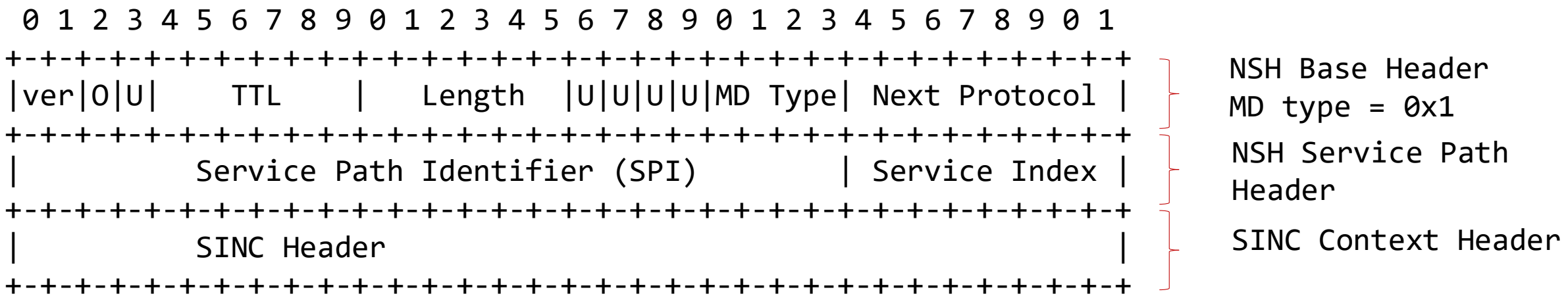
Deployment - SFC

SFC is one possible way to steer traffic to the right in-network SINC-capable switch where SINC is a Service Function (SF)

Case 1: hosts do support SINC

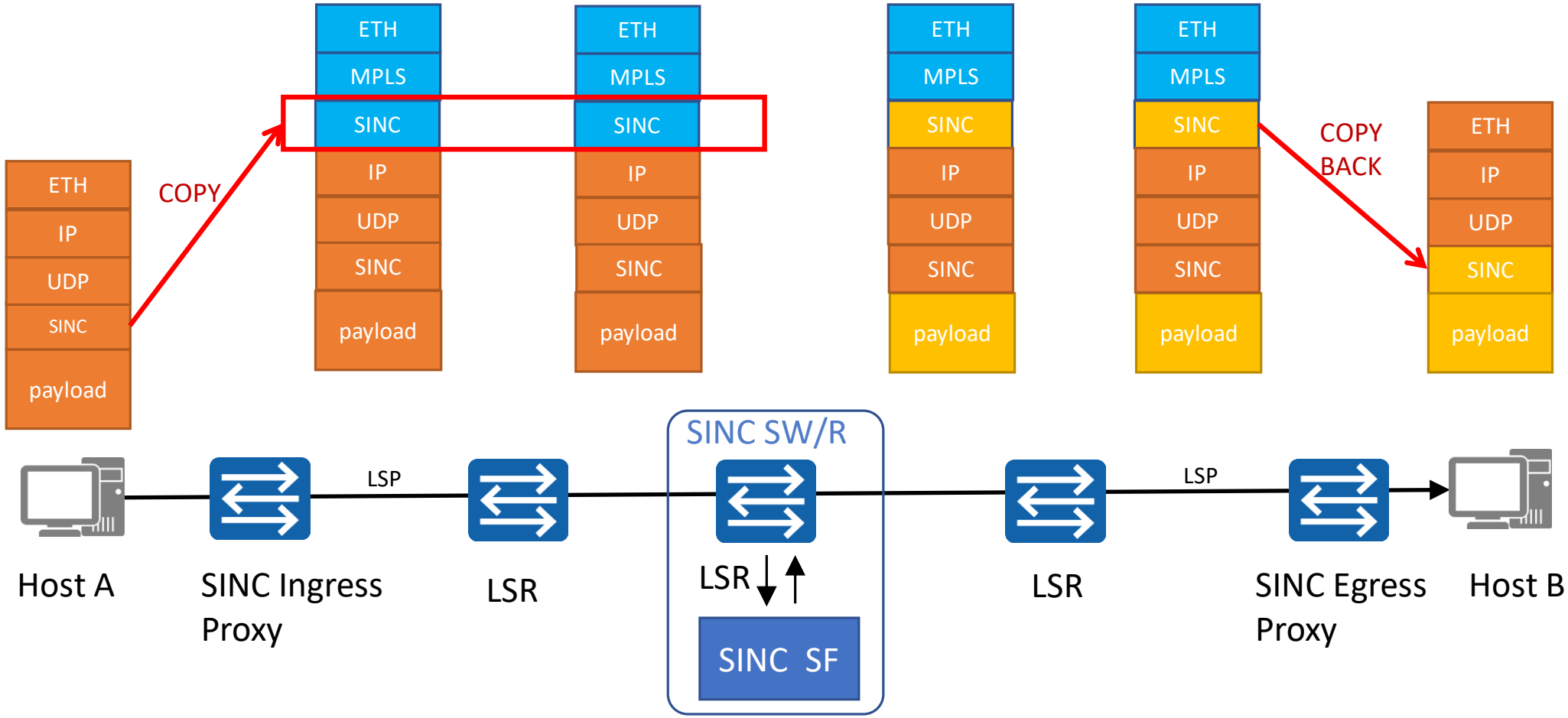


SINC NSH Encapsulation



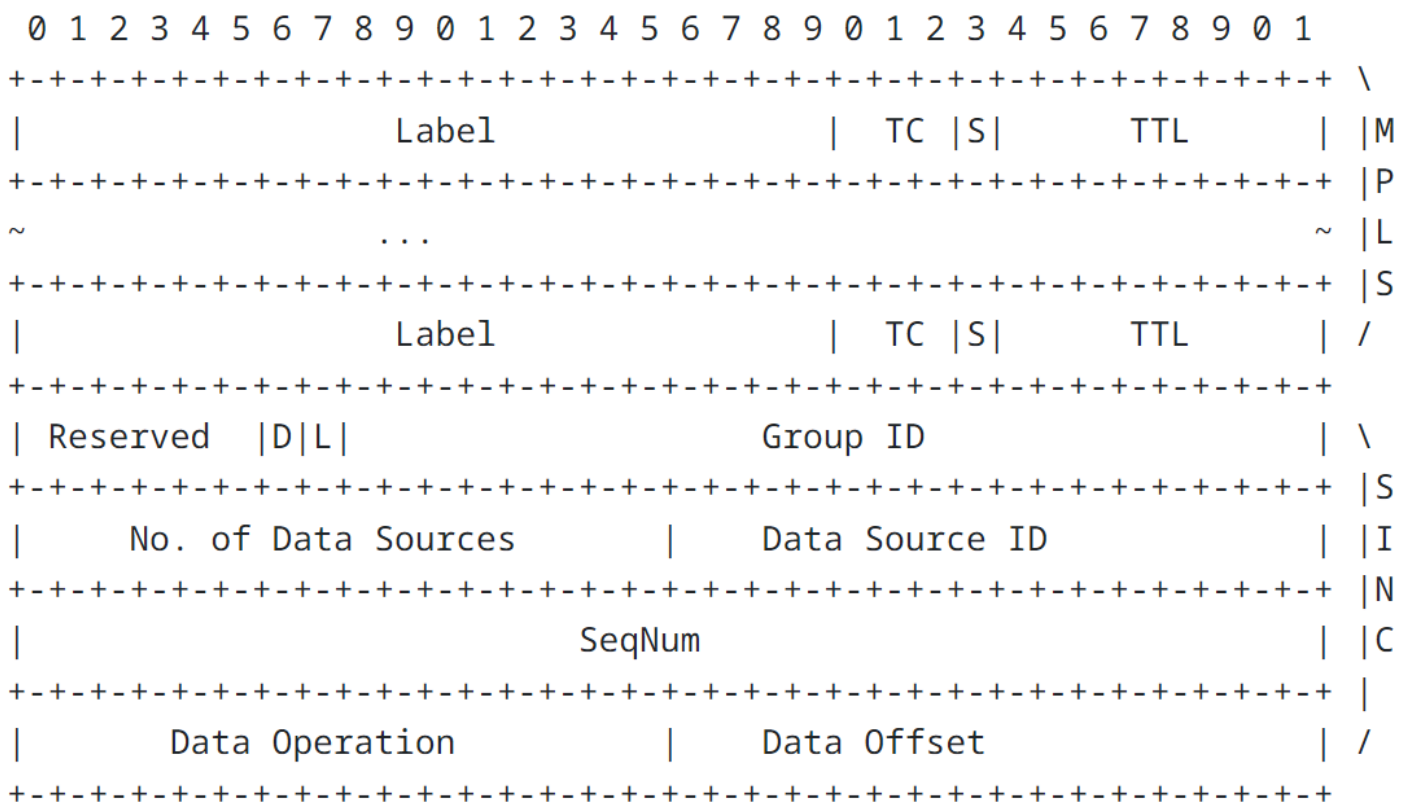
- **NSH Base Header:**
  - Use the NSH Meta Data (MD) fixed-length context headers to carry the data operation information
  - MD type = 0x4 was used in the draft because the size of the original design of the SINC header is not 16 bytes. It will be updated in the next version of the draft.
- **NSH Service Path Header:** as defined in RFC 8300.
- **SINC Context Header:** as defined SINC Header. SFC Proxy copy these information in SFC header.

Deployment - MPLS



The SINC SW/R will further identify the location of the SINC header by checking the Next Hop Label Forwarding Entry (NHLFE) as defined in the [RFC3031].

SINC MPLS Encapsulation



Drafts Status

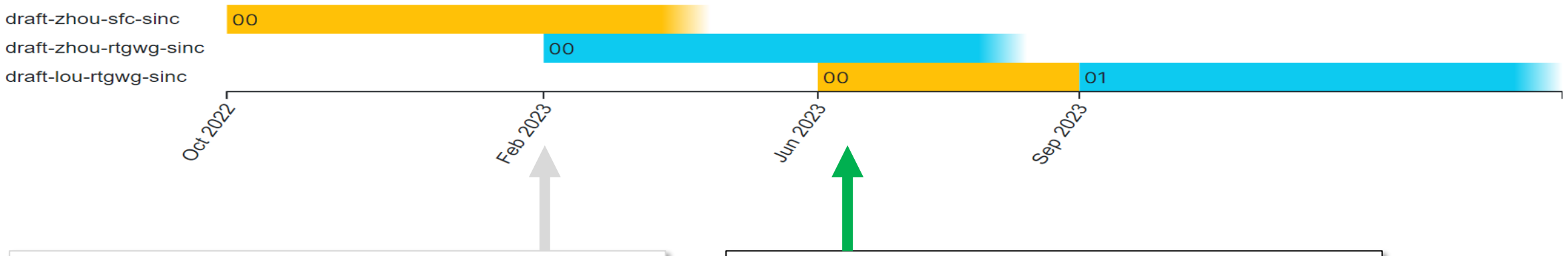
Signaling In-Network Computing operations (SINC)

draft-lou-rtgwg-sinc-01

Status | Email expansions | History

Versions:

00 | 01



- Split the original draft into 2 drafts,
  - draft-zhou-rtgwg-sinc depicts the main spec
  - draft-zhou-rtgwg-sinc-deployment-considerations covers the deployment considerations
- New co-author: Jinze Yang

- Added control plane considerations
- New co-author: Yizhou LI (Huawei), Kehan YAO (China Mobile)

Next Steps

- Encourage discussion on the mailing lists
- Update the draft based on comments and remarks
- Welcome to contributions and co-authors

THANKS!