IETF 116 – RTG WG

Signaling In-Network Computing operations (SINC)

~~draft-li-zhou-sfc-sinc-00~~

draft-zhou-rtgwg-sinc-00

David Lou – Huawei Technologies

IETF 116 – Yokohama

Since IETF 115

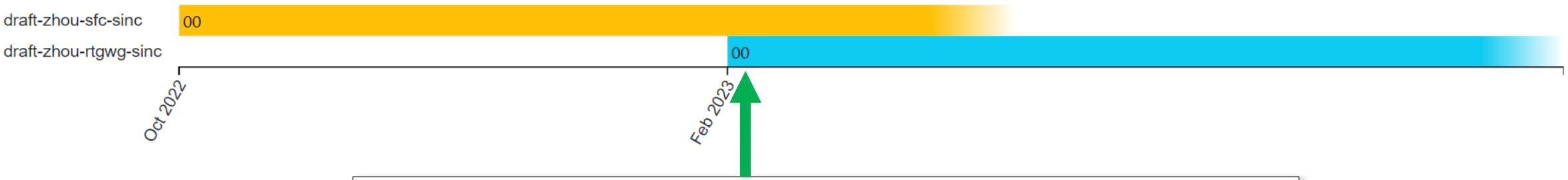
Signaling In-Network Computing operations (SINC)

draft-zhou-rtgwg-sinc-00

Status | IESG evaluation record | IESG writeups | Email expansions | History

Versions:

00



- 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

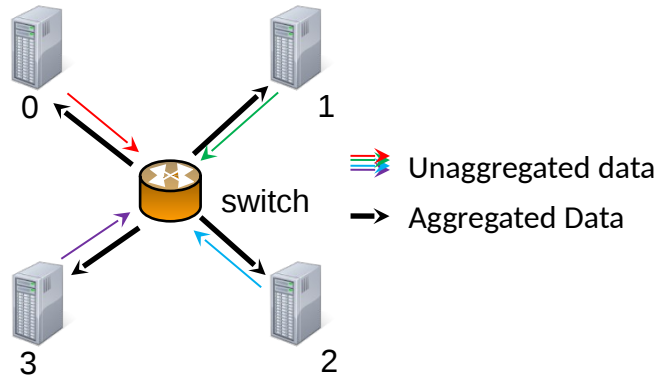
Updates since IETF 115

Table of Contents

- Moved the SFC part to a new “deployment consideration” draft
- Provided a SINC framework overview
- Discussed about the data operation mode
  - Individual computing mode
  - Batch computing mode
- Updated the SINC header format
- Added control plane requirements
- Overall text refinement

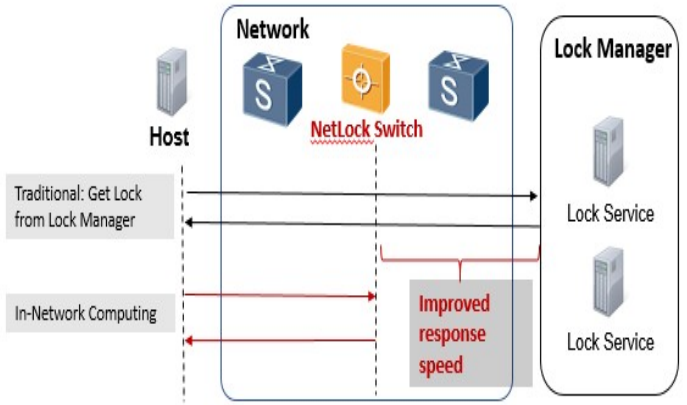
SINC Use Cases (recap)

NetReduce



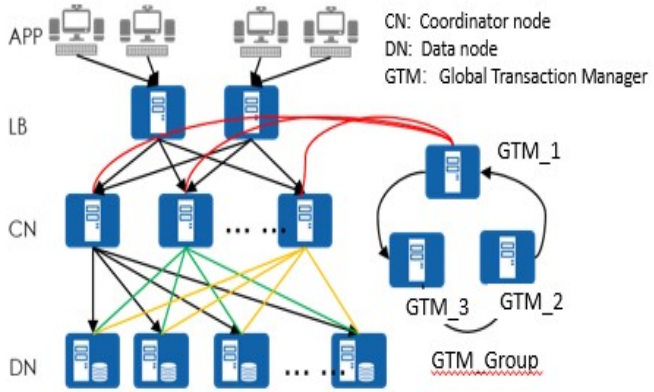
- Comparing with the host oriented solutions, in-network aggregation could potentially reduce nearly half the aggregation data
- NetReduce is >1.5x faster, and has better scalability than ring all-reduce.

NetLock



- The lock manager can be abstracted as Co mpare And Swap (CAS) or Fetch Add (FA) o perations.
- 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

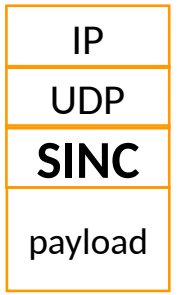


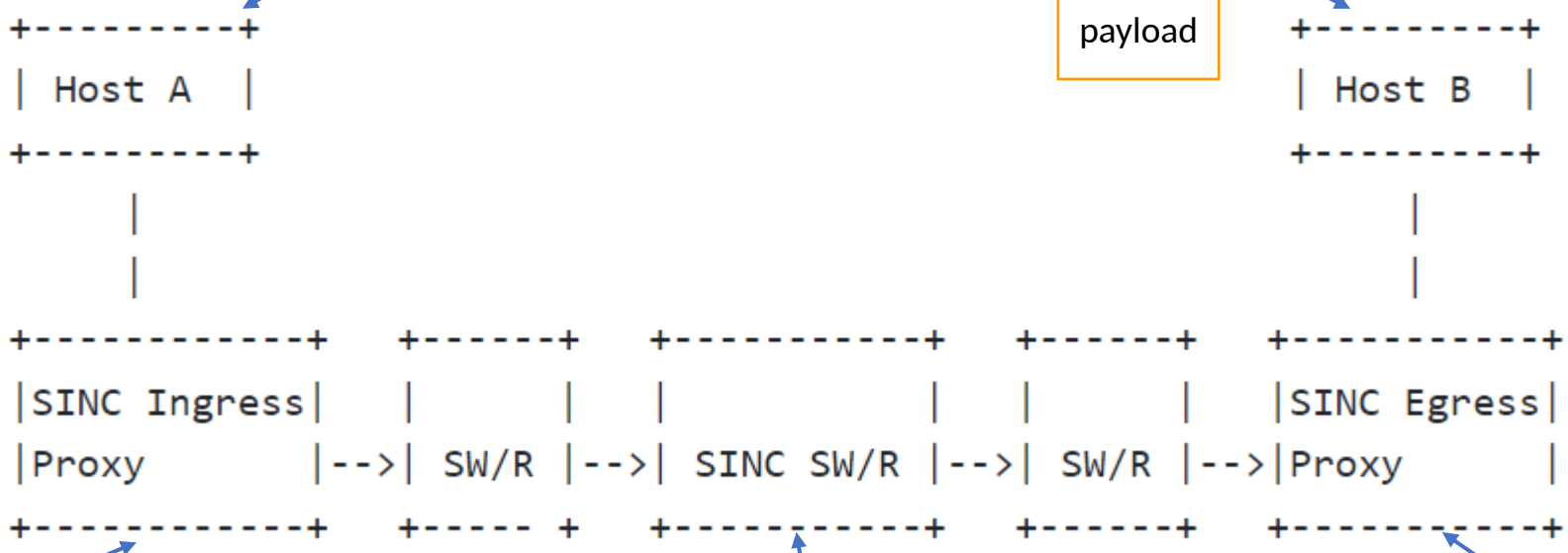
- Switches could realize the sequencer[2] by using a "Fetch-and-Add" operation.
- Compared with Gbps-level throughput of servers, network devices have Tbps-level throughput and line-rate processing capabilities

An explicit and general mechanism is required to tell the switch how to process the packet

SINC Framework

A host MUST be SINC-aware



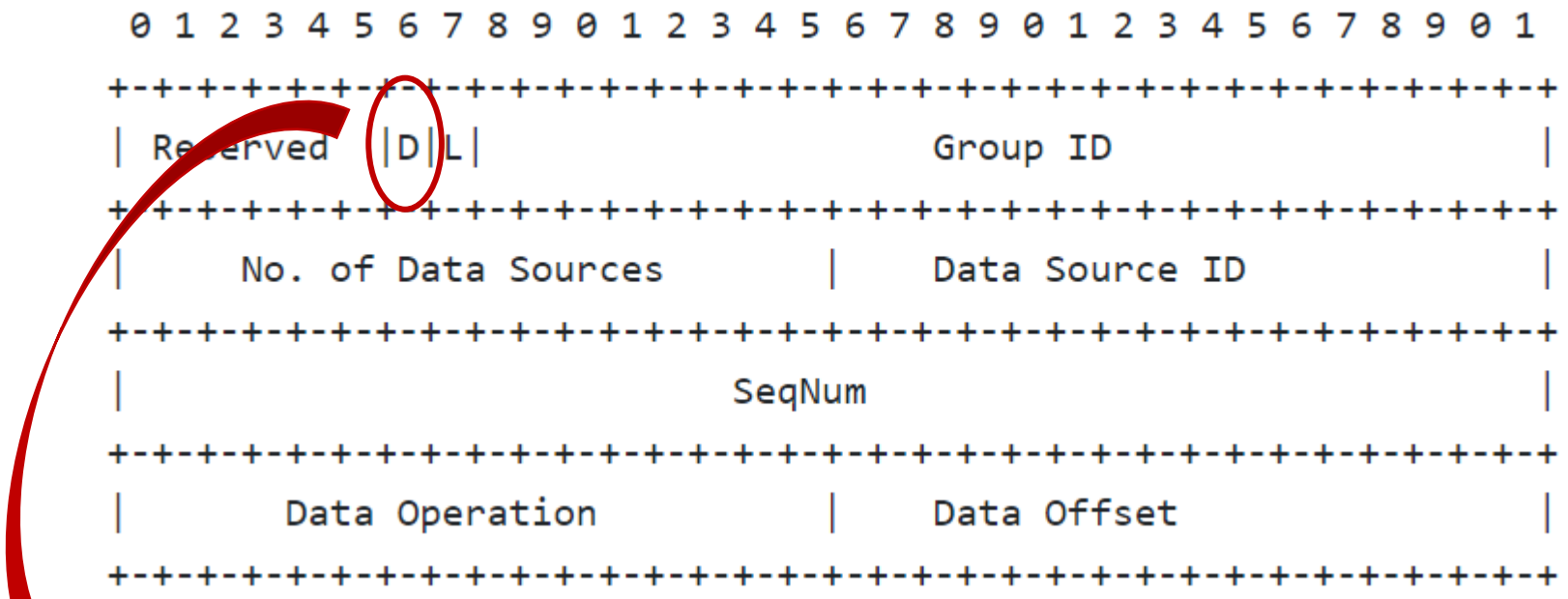


SINC Ingress Proxy encapsulates the packet

SINC-capable switches/ routers executes all or part of the data computation during the transmission

SINC Egress Proxy removes the encapsulation

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 Requirements

- The SINC control plane should **be aware of the switch resources**. This may be achieved by regularly querying the devices.
- The SINC control plane should be able to **select the switches/routers based on certain constraints**. For instance selecting switches/routers that are able to perform a specific more complex operations, or being able to distribute the load on various alternative switches/routers without increasing the transmission delay.
- The SINC control plane should be able to **provide the necessary configuration** so that packets flow to the right place and encapsulation/decapsulation operations are performed correctly.
- The SINC control plane should provide **monitoring and failover mechanism** in order to handle errors and failures.

Next Steps

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

THANKS!

IETF 116 – RTG WG

Signaling In-Network Computing operations (SINC) deployment considerations

draft-zhou-rtgwg-sinc-deployment-considerations-00

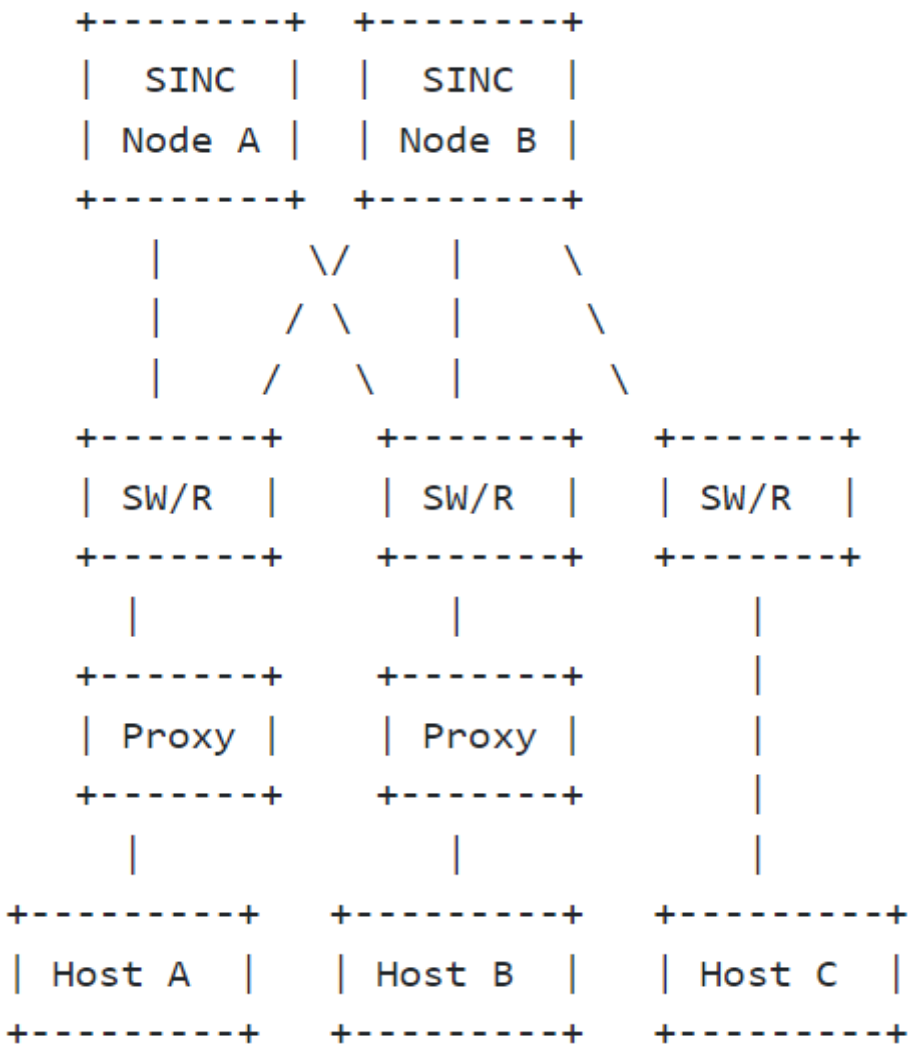
IETF 116 – Yokohama

Table of Content

Table of Contents

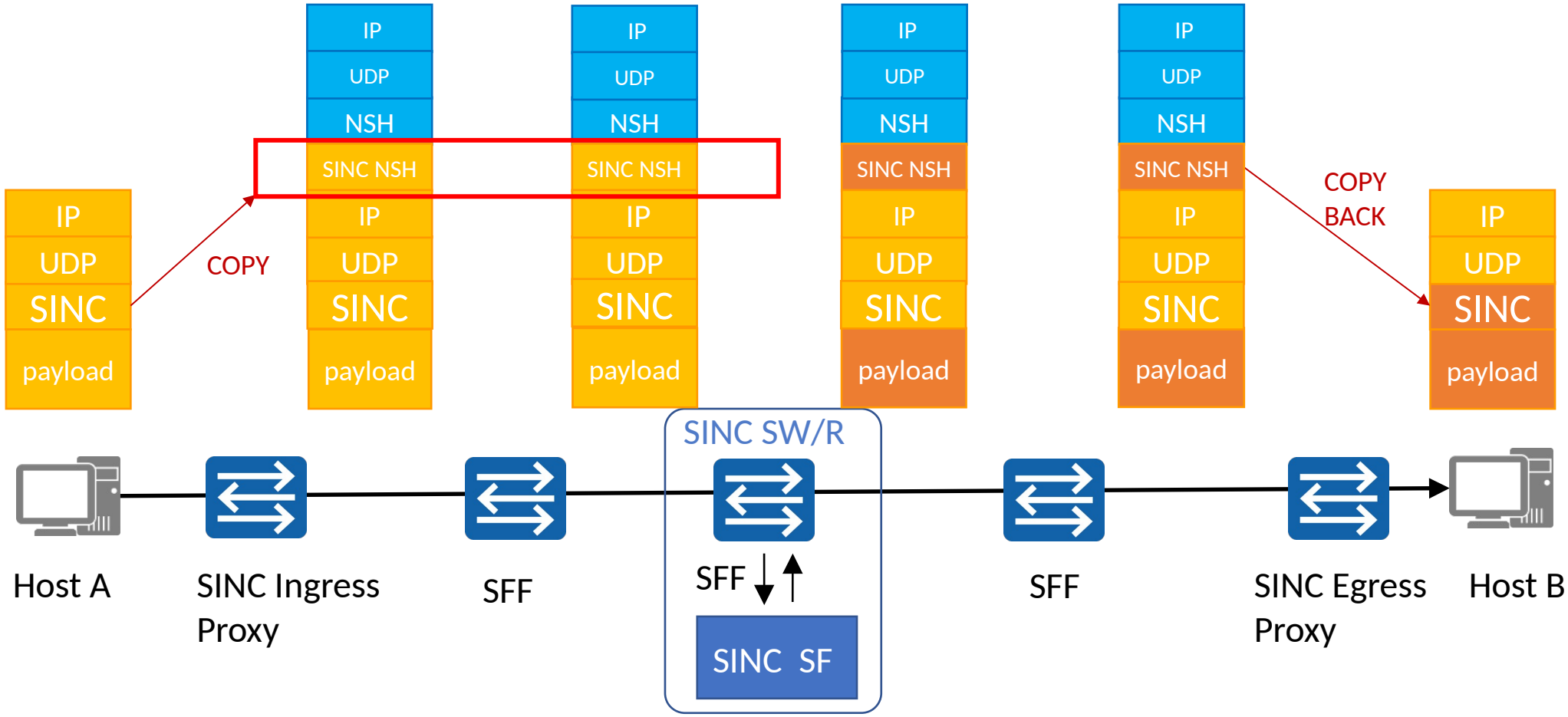
- General deployment considerations
- Example of SINC over SFC
- Example of SINC over MPLS

Deployment Considerations

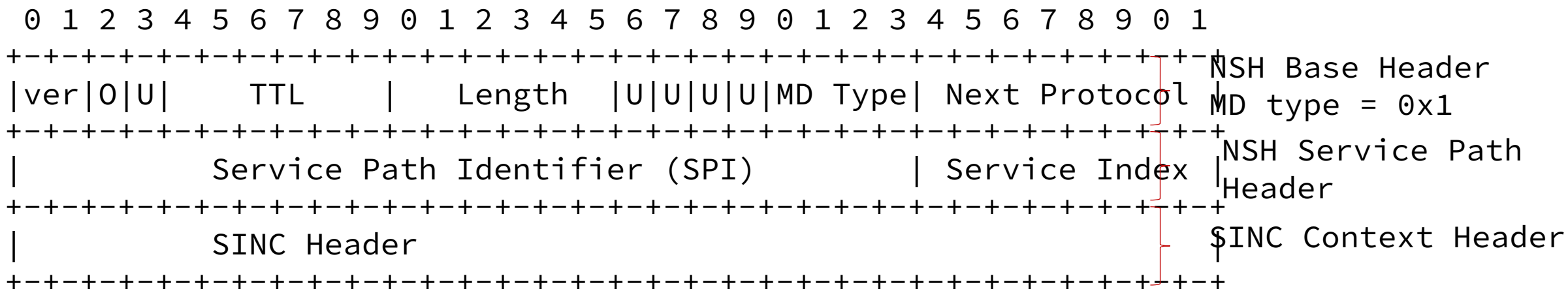


- The SINC capable SW/Rs should periodically advertise their networking and computing capacities and capabilities
- The control plane is responsible to create a proper route where the data in the packet will undertake the desired computation before arriving at the destination host.
- The encapsulation could be located at layer 2, 3 or 4 dependent of the network context and application environments
- The SINC header is usually copied right after the new encapsulation header, which makes it easier to access

Example: SINC over SFC

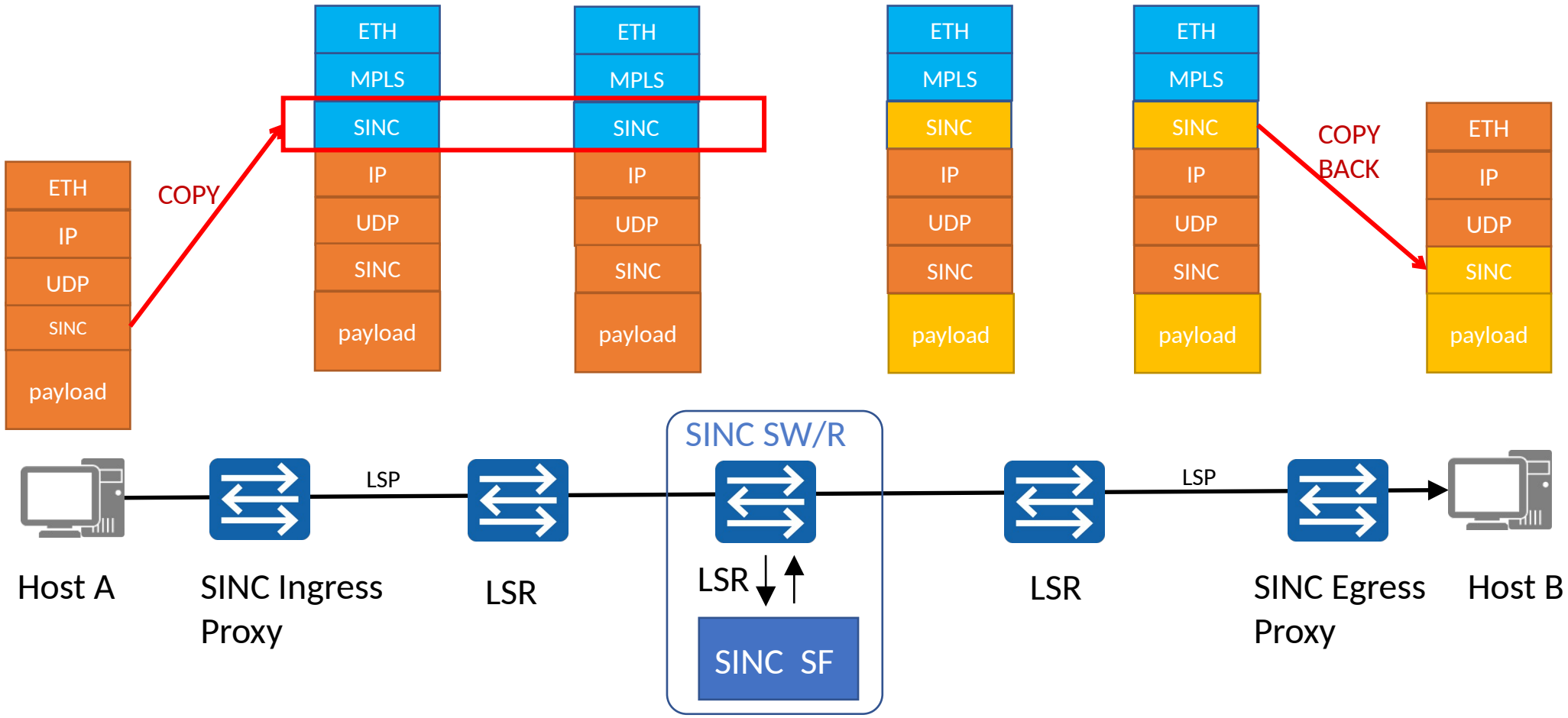


SINC NSH Encapsulation

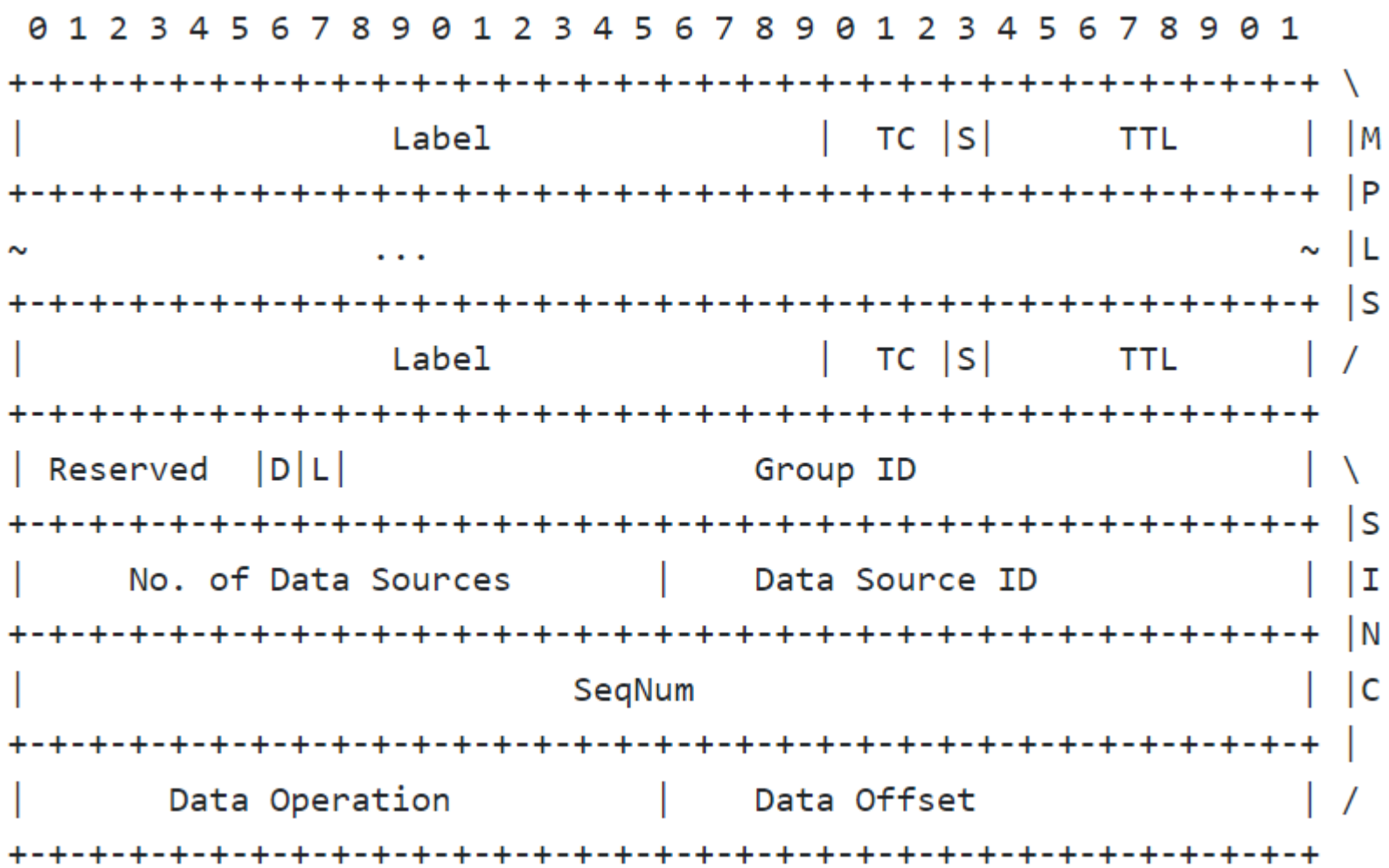


- **NSH Base Header:**
  - Use the NSH Meta Data (MD) fixed-length context headers to carry the data operation information
  - MD type = 0x1 (suggested) was used in the draft because the size of the original design of the SINC header is 16 bytes.
- **NSH Service Path Header:** as defined in RFC 8300.
- **SINC Context Header:** as defined SINC Header.

Example: SINC over MPLS



SINC MPLS Encapsulation



Next Steps

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

THANKS!