Congestion Signaling (CSIG)

Simple and Effective In-band Network Signals for Efficient Traffic Management in Datacenter Networks

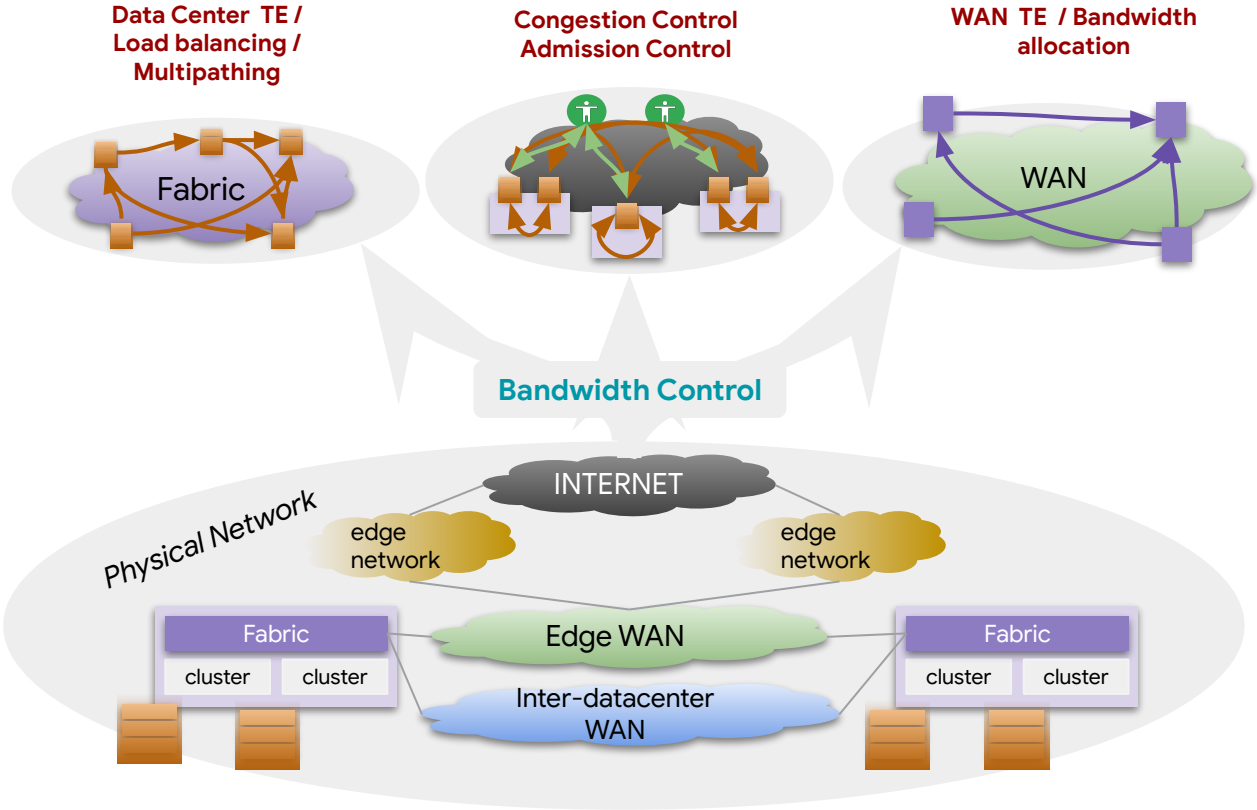
https://datatracker.ietf.org/doc/draft-ravi-ippm-csig/

AIDC-IETF118
November 7, 2023

Networking for ML Workloads

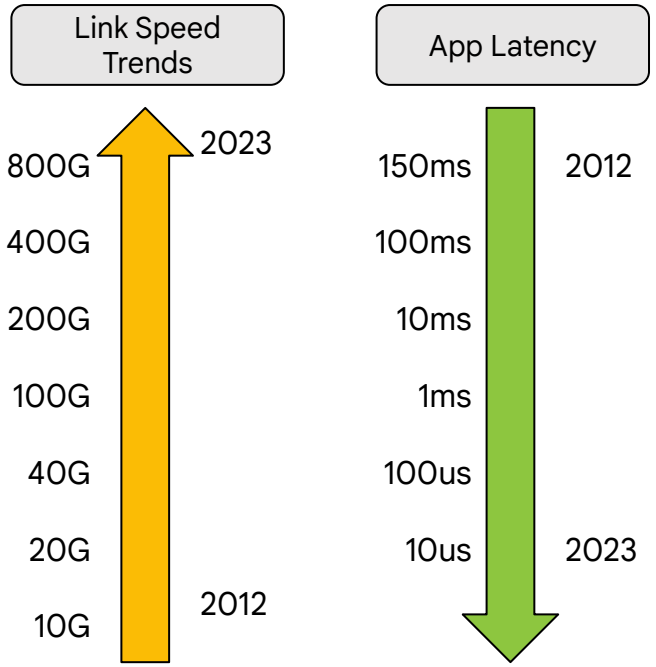
- Trend: More accelerators, more bandwidth per accelerator
  - Horizontal scaling → exercises shared data center network fabric
- Network efficiency is critical for AI/ML
  - Bursty traffic, large # of connections, flow collisions
  - Large-scale synchronized bursts stress the network unlike traditional compute / storage applications
  - Training → throughput *and* latency sensitive,
  - Inference / serving → latency sensitive
- Network performance directly impacts Training step time
  - Barrier collectives in training sensitive to 100p network latency
  - Throughput sensitive where compute/communication overlap.

Ecosystem of Traffic Control Loops



Collection of control loops work in conjunction to achieve network efficiency and performance

Requirements for Network Efficiency



To sustain network efficiency, early and accurate visibility of **bottlenecks** is important for

- Robust congestion control
  - Requires real-time network telemetry signals from the fabric to address blind spots in end-to-end algorithms
- Traffic management
  - Requires visibility into transient congestion due to (milliseconds) bursts to guide traffic engineering, multipathing and load balancing techniques and accommodate bursts

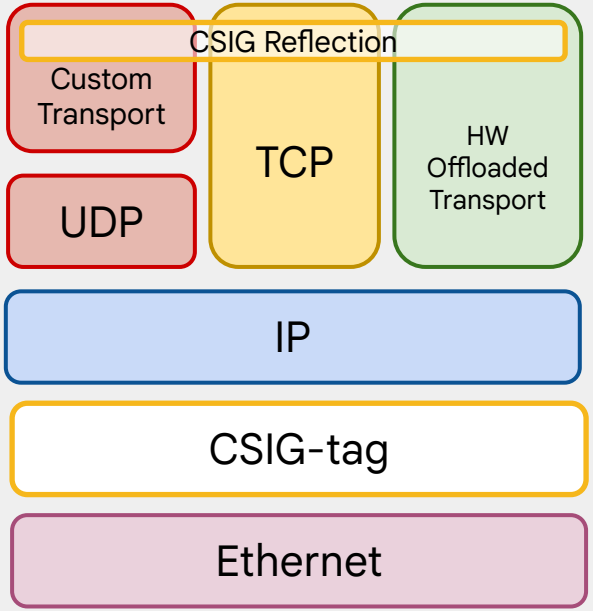
In-band signaling over application data packets is required to enable fast response and flow attribution

In-band Network Signals for Detecting Path Bottlenecks

Minimum Available Bandwidth: min(ABW)	The minimum available bandwidth in bps across all links on the packet path
Maximum Link Utilization: max(U/C) or min(ABW/C)	The maximum link utilization in percentage of link speed across all links on the packet path
Maximum Per-Hop Delay: max(PD)	The maximum per-hop delay across all hops in the packet path

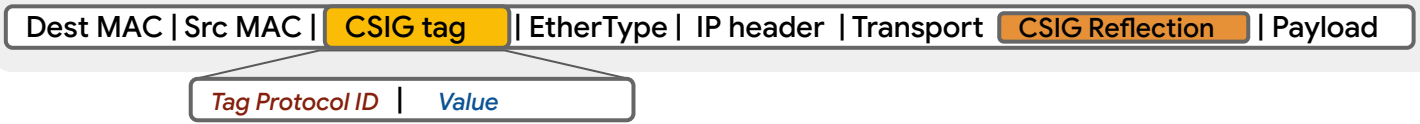
...and potentially many more

CSIG: Practical & Effective In-band Signaling protocol

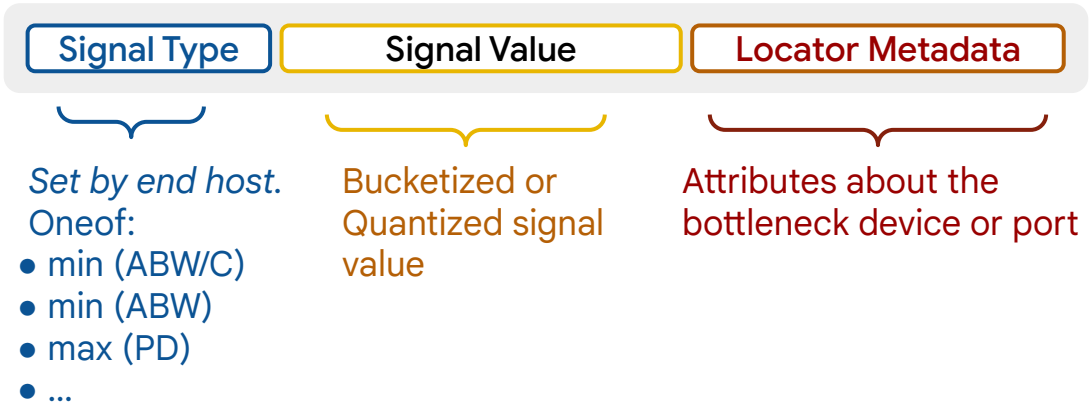


- Provides fixed-length simple summaries from the path bottlenecks
- Designed for Congestion Control, Traffic Management and Network debuggability use-cases
- Designed for brownfield deployment with backward compatibility / interoperability
- Link to IETF Draft - https://datatracker.ietf.org/doc/draft-ravi-ippm-csig/

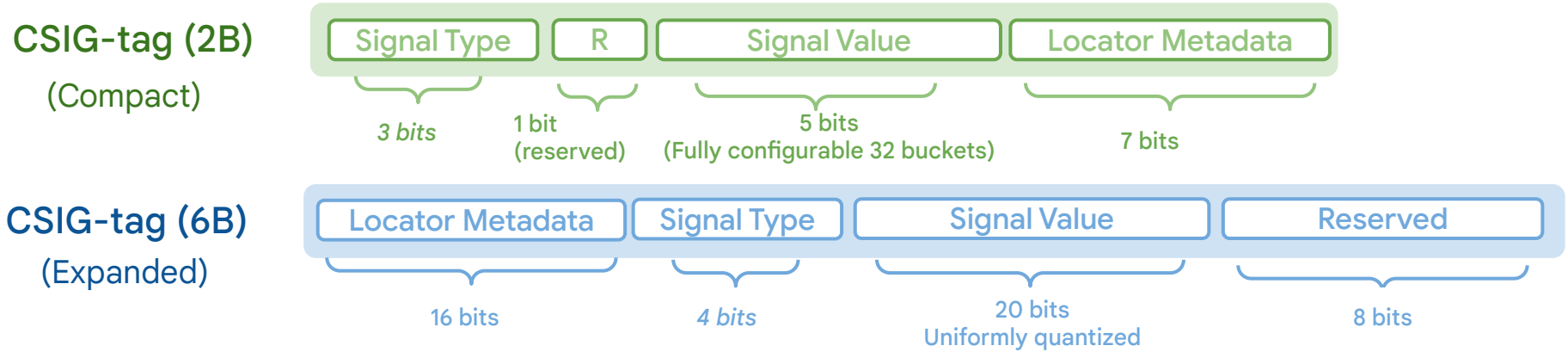
Packet format (with reflection)



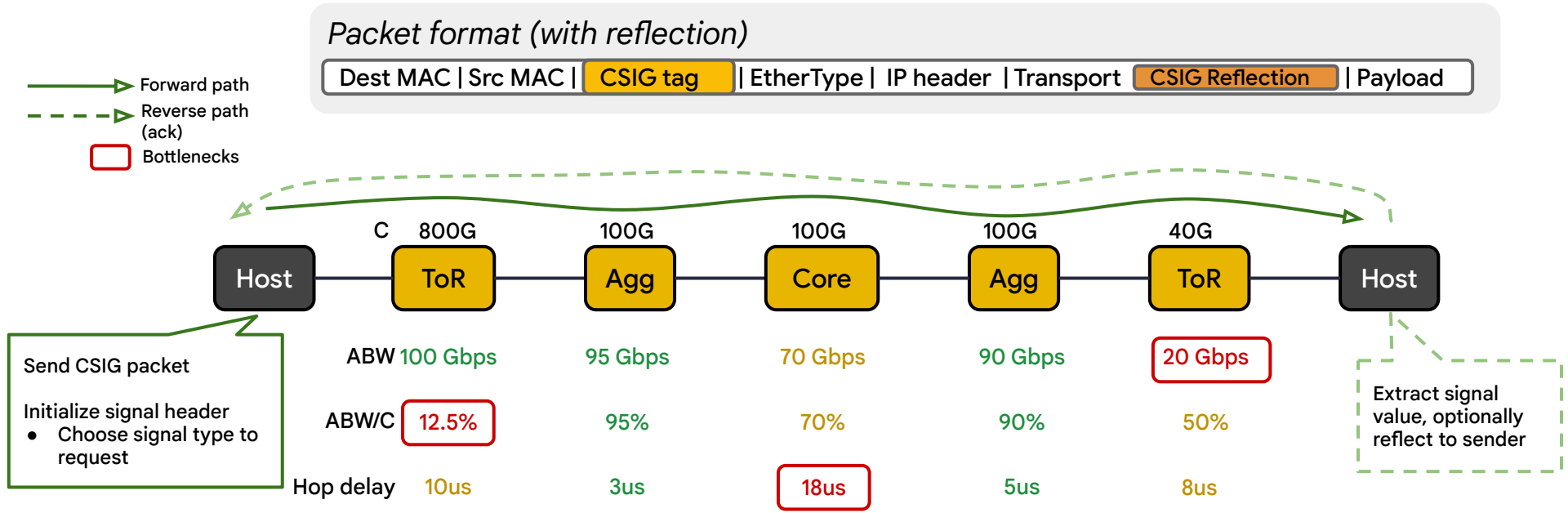
CSIG-tag Value Structure



- Can carry one congestion signal at a time on a given packet.
- Signal Type: Type of signal being carried in the header.
- Signal Value: Fully configurable bucketized or uniformly quantized signal value
- Locator Metadata of the bottleneck port or device.



Life of a CSIG Packet



- Forward path bottleneck information carried and updated over L2 CSIG-tags
  - ***Compare-and-replace*** operation at each network hop
  - Red boxes show forward path bottleneck hops for each signal type
- CSIG signals reflected back to sender over L4 CSIG Reflection headers
  - E.g., TCP Options, Falcon header, Ponyexpress transport header

Advantages of CSIG

Small/fixed header per packet

Low bandwidth overhead for small packets: 4B or 8B in 4K MTU is < 0.1% byte overhead

No MTU change

Low resources and parser complexity at NIC/Switch

Simple signals realized over live packets at line-rate

Simple port and queue signals driven by targeted use cases

Aggregate signals `min()` and `max()` implementable in ASICs at line-rate for every packet

Deployable on all live production packets without the need for dedicated probes

Compatible with in-network tunneling and encryption

CSIG-tags stay at a fixed offset, unaffected by encapsulations at L3+ headers

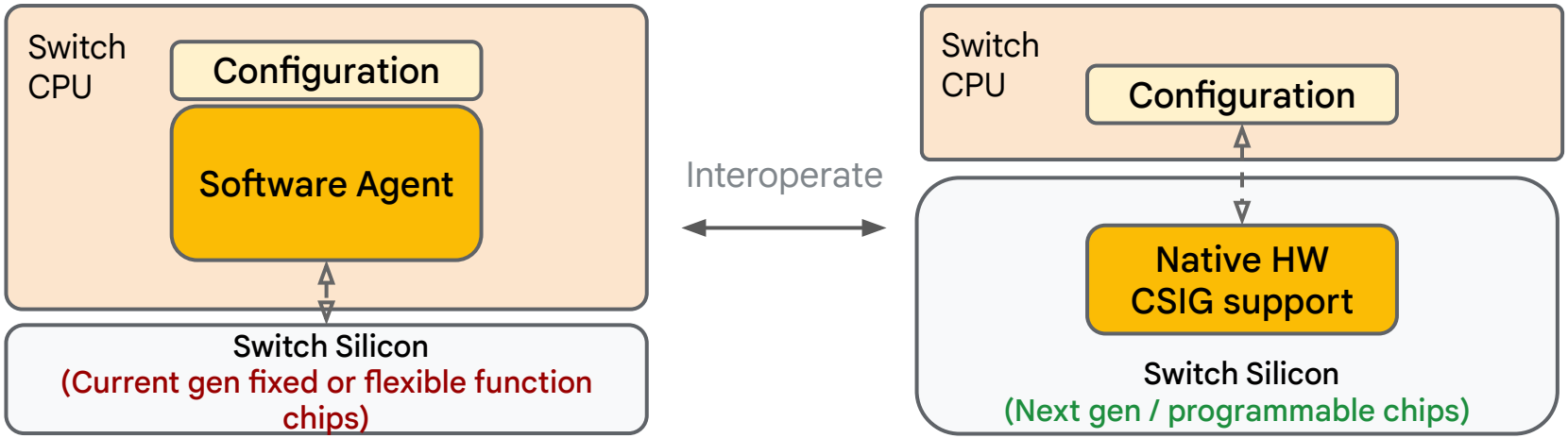
Simplifies hardware design - No need to write / relocate metadata to new offsets during encap / decap

Backward Compatibility

Co-exist: No impact to existing ECMP over L3/L4 headers

Inter-operate: At L2 so legacy switches can participate with a software-assisted approach and VLAN capabilities

Backward Compatibility with Software-assist



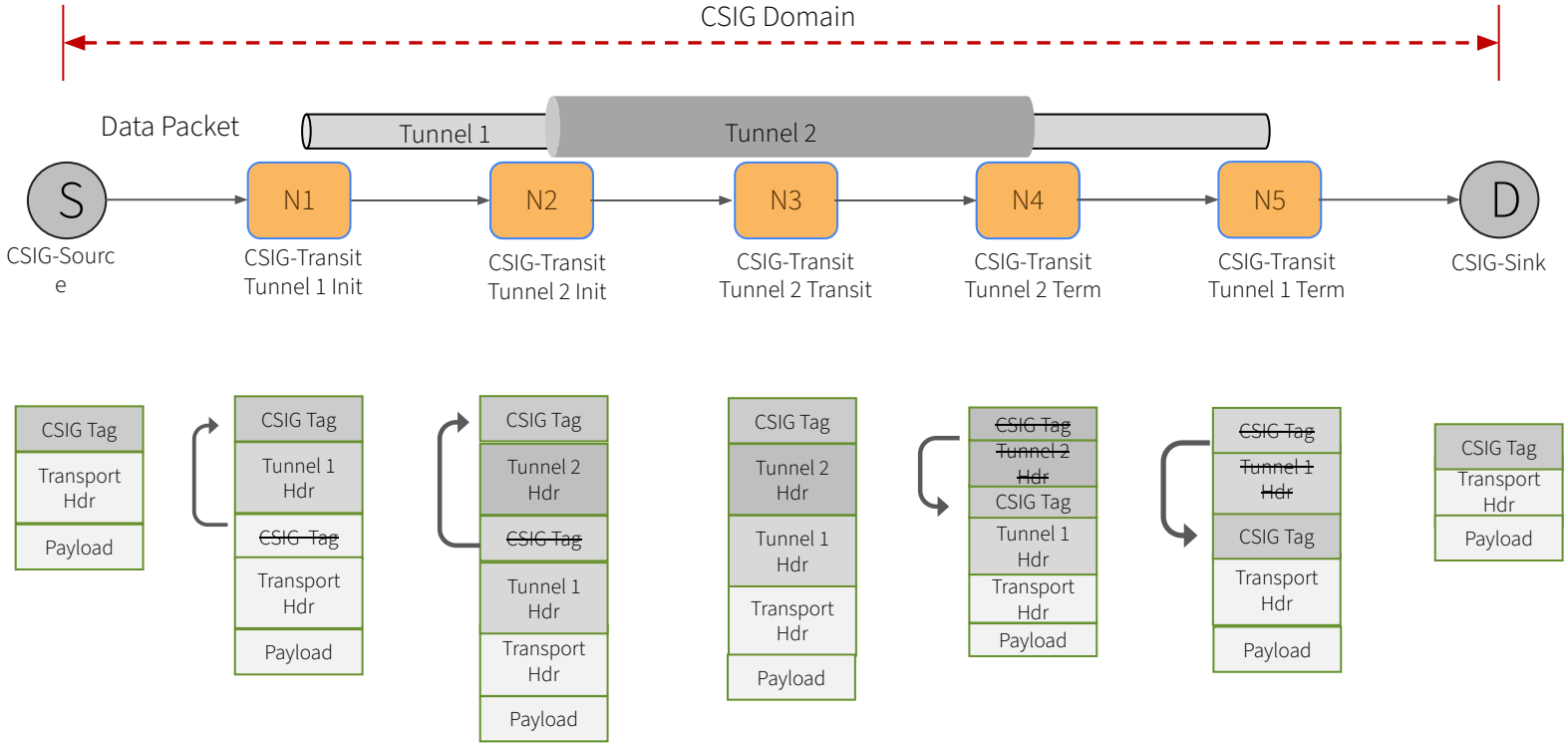
- Software-assisted CSIG (2B) implementation for existing devices
  - Packet header update performed via Match/Action rules in TCAM/Exact Match tables, by **repurposing VLAN capabilities**
  - **Software agent** runs on Switch CPU to periodically (i) read HW counters, (ii) compute available bandwidth / queue depth, and (iii) update Match/Action policy
- HW-offloaded CSIG for future devices
  - Software agent functionalities offloaded to hardware, configured through software SDK
  - Supports larger variety of signals
  - Supports both 2B and 6B formats

Hardware considerations for native CSIG support

- **Signal Computation Algorithms**
  - Available bandwidth per egress port: Raw-ABW / EWMA / DRE
  - Hop delay per packet: Ingress / egress timestamp deltas
- **Signal Quantization & Bucketing**
  - Uniform vs non-linear quantization
  - Fully configurable buckets to optimize encodings
- **Tag update functions**
  - Compare-and-replace, read-modify-write
  - Sum, Min, Max, Count
- **Tag propagation**
  - Parsing, editing, initialization
  - Tunnel propagation - encap / decap / turnaround

CSIG spec is HW-friendly and retains high performance switch / NIC architecture with near-zero impact on power budget, die size or pipeline latencies

Tunnel Propagation of CSIG Tag



Compatible with IPv6, IPv4, all IP-in-IP tunnels including 6-in-6 and 6-in-4, IP-in-IP-in-IP, VxLAN, Geneve, PSP, IPSec etc.

CSIG-tag Placement

CSIG-tag is always the last tag in the layer 2 header

ARPA	dstmac / srcmac / **csig-tag** / ethertype / payload
802.1q:	dstmac / srcmac / vlan-tag / **csig-tag** / ethertype / payload
802.1ad	dstmac / srcmac / vlan-tag / vlan-tag / **csig-tag** / ethertype / payload
802.1ad tunnel	dstmac / srcmac / vlan-tag / vlan-tag / vlan-tag / vlan-tag / **csig-tag** / ethertype / payload
802.1ae	dstmac / srcmac / security-tag / vlan-tag / **csig-tag** / ethertype / payload

How CSIG improves application network performance

Congestion Control
Reduce transfer latency

Ramp-up quickly and safely to available bandwidth in the absence of congestion

Respond precisely to the bottleneck hop in the presence of congestion

Traffic management
Maximize goodput

Multipathing and Load Balancing

- Select paths with the most available bandwidth
- Use locator to distinguish between incast and core congestion

Traffic Engineering

- Aggregate CSIG values spatially and temporally across flows, augment Traffic Matrix
- Provision routes for better burst absorption, route around bottlenecks

Network debuggability and operations
Better network provisioning

Debug and resolve bottlenecks in the network, as experienced by flows

Mesh probers reveal bottleneck trends between end-point pairs

Timely provisioning and repair processes based on aggregated CSIG data