Consensus Building for Efficient Forwarding Header in Ethernet

2026 IEEE 802.1 & 802.3 January Interim Meeting

Authors

CMCC: Runlong Hu, Jieyu Li, Weiqiang Cheng

Tencent: Yinben Xia, Weifeng Zhang

Alibaba: Yazhu Lan, Kun Liu, Pengfei Huo

Supporters

Yanwei Wang (IEIT)

Yao Liu (ZTE)

Wei Ma (H3C)

Xiaojun Wang (Ruijie Network)

Jinyou Dai (FiberHome)

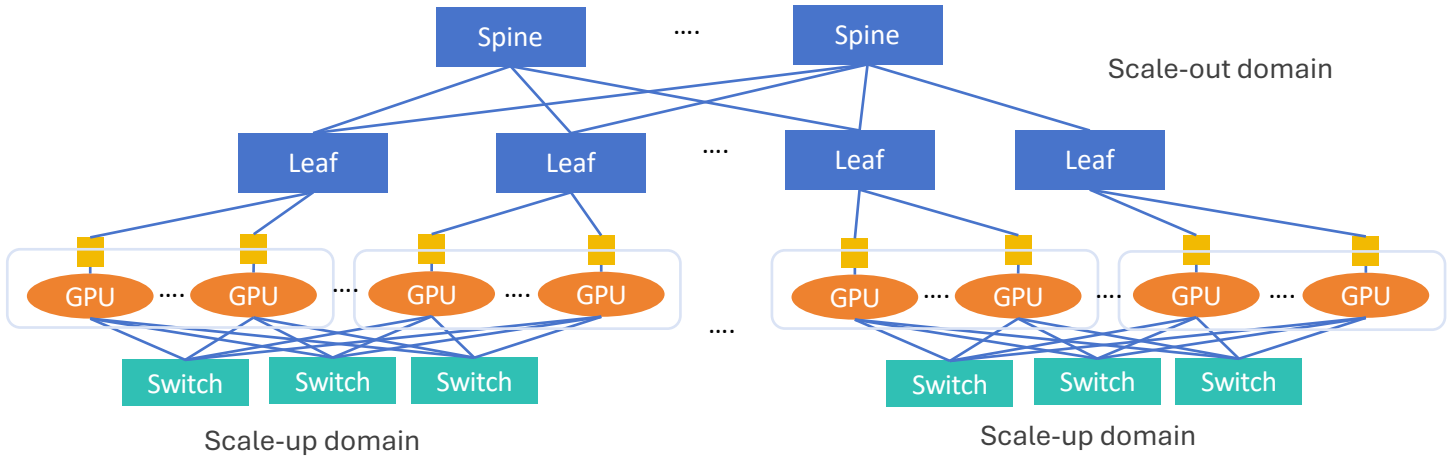
Outline

- **Background**: why small-message efficiency matters in AI cluster network
- **Industry situation**: multiple proposals emerging
- **Technical ideas**: compressing / reshaping forwarding information
- **Standards impact**: 802.3 / 802.1 considerations
- **Proposal**: what IEEE 802 can do

Background: Header Efficiency Requirement in AI cluster Network

AI cluster network overview

- Homogenous, relative static and managed network domain.
- Pursue extreme performance
  - High throughput
  - Low latency



- AI cluster traffic includes massive volumes of small messages like:
  - Control / synchronization operations;
  - Fine-grained memory access (load / store);
  - MoE all-to-all communication;

- Header overhead becomes a throughput limiter for small messages
  - The conservative MAC+IP+UDP/TCP overhead can be a large fraction of the packet for small payloads
  - For load/store operations with 128B packet payloads, bandwidth utilization only reaches 79.0% (for IPv4) or even 70.3% (for IPv6).

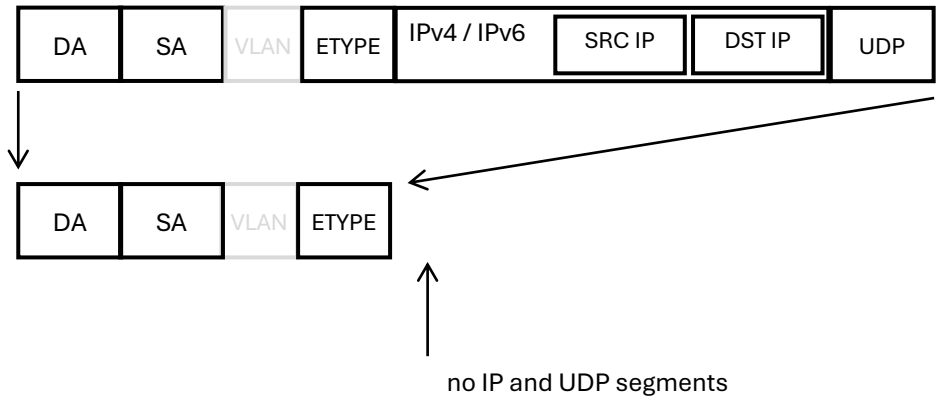
- An efficient forwarding header can directly improve bandwidth utilization and reduce latency

Industry Situation: Converging Need, Diverging Formats

- **Many groups are actively defining efficient headers for Ethernet scale-up**
  - EH that ESUN is currently working on
  - UEC reduced-size forwarding header work
  - AFH in SUE from Broadcom
  - Link Layer Bypass Header developed from Eth+ led by Alibaba Cloud and the Institute of Computing Technology, Chinese Academy of Sciences
  - ETH-X header compression concepts led by Tencent and CAICT
  - Other public fabric efforts (e.g., Ethlink from ByteDance)
- **Industrial Common Theme**
  - Reduced overhead for small messages while keeping forwarding practical
- **Risk of lacking unified standards**
  - Multiple incompatible header formats and behaviors
  - Ecosystem fragmentation across NICs, switches, and tooling
- **Opportunity for IEEE 802**
  - IEEE 802 can define a common framework including format and usage rules

Key Design Idea: Compress Forwarding Information (1)

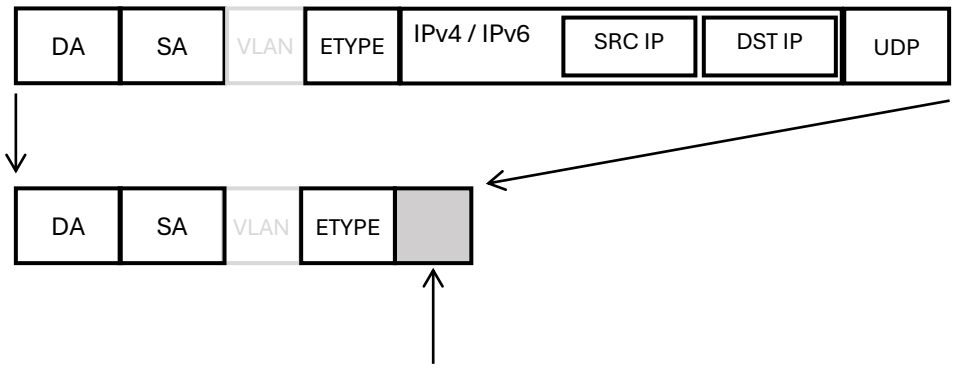
- **Characteristics enabling compression**
  - Known topology / managed domain
  - Limited endpoint scale relative to 48-bit global MAC space
- **Two forwarding scenarios**
  - MAC-like forwarding: limited endpoint scale, simple topology, low risk of ringing and congestion
  - IP-like forwarding: larger endpoint scale, more complex topology, potential risk of ringing and congestion
- **Truncate network (L3) and transport (L4) headers for MAC-like forwarding**
  - DA/SA provides enough information for making forwarding decisions
  - ring prevention (TTL/Hop Limit), load balancing (Entropy) become unnecessary



Key Design Idea: Compress Forwarding Information (2)

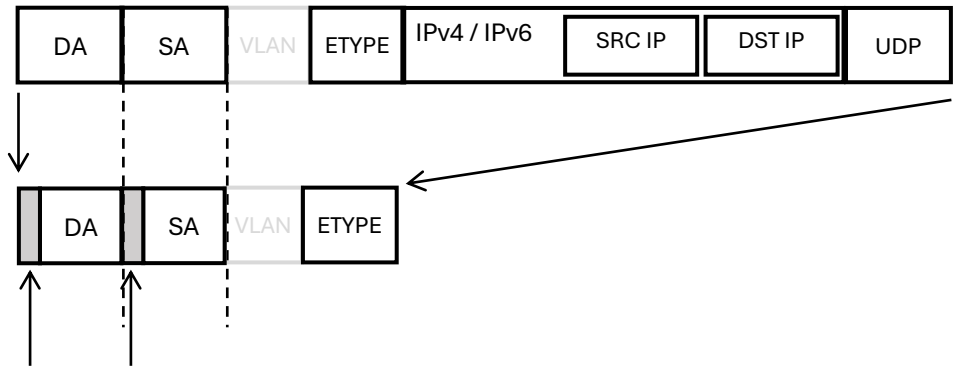
- **Compressed network (L3) and transport (L4) headers for IP-like forwarding**
  - MAC + IP address are redundant; unified into a single ID address
  - segments regarding ring prevention (TTL/Hop Limit) and load balancing (Entropy) need to be reserved by compressed headers
- **Two compressing directions**
  - Tag-based carriage: Keep DA/SA unchanged while adding compact forwarding-related fields using one or more tags
  - Header Redefinition: Re-map DA/SA space to carry compressed identifiers and selected semantics
- **Key question for standards**
  - Do we optimize for minimal disruption or maximum compression?

Option 1: Stacking tags to carry IP/UDP segments



tag stacking to carry fields from IP/UDP

Option 2: Collapsing IP/UDP segments into DA/SA



compress fields from IP/UDP into DA/SA segment

Standards Impact: Where are the Boundaries?

	Option 1: Tag-based carriage	**Option 2:** Header Redefinition
IEEE 802.3	**No Conflict**: DA/SA remains unmodified.	**No Conflict**: Collapsing MAC, IP and UDP into one still adheres to the 6-octet field requirement.
IEEE 802.1AC	**No Conflict**: Unmodified DA/SA ensures the standard MAC client interface can be reutilized.	**Conflict**: The current MAC client interface has no mechanism to accept separate routing ID and transport data parameters.
IEEE 802.1Q	**Conflict**: Apart from DA/SA and VLAN, extra segments from IP/UDP also influences the L2 forwarding behavior.	**Conflict**: the same reason as option 1 + entries of the MAC table have to be maintained at a flow-level granularity rather than a single DA/SA; path calculation depending on DA/SA in STP is corrupted.
IEEE 802.1X	**No Conflict**: Current service is preserved without modifying DA/SA	**Conflict**: authenticator record depends on a complete SA.
IETF	**No Conflict**: L3/L4 processing is entirely bypassed when using efficient forwarding header. Even though some IP/UDP segments are carried, the handling is transparent to L3/L4.	**No Conflict**: the same reason as option 1.

- 802.3 MAC frame format constraints: field sizes, Ethtype usage
- 802.1Q bridging behavior: If L3/L4-like semantics are carried at L2, are they purely opaque metadata or inputs to forwarding decisions / congestion control / retry / ordering?
- 802.1AC Service interface expectations: What a MAC client can provide/consume
- DA/SA-dependent services such as IEEE 802.1X

Proposal

- **Work for initiating a new project in IEEE 802.1 WG, with involvement of IEEE 802.3 WG, to standardize efficient forwarding header in ethernet.**

- **Potential items to work on:**
  - Specify the approach
    - choosing from tag-based carriage and/or header redefinition, or keeping both available
  - Define a standard efficient forwarding header format
    - determine positions and sizes of each field
    - assign the corresponding Ethtype for the updated header pattern to be identified
  - Determine processing rules and boundaries
    - define rules of making forwarding decisions taking L3/L4-like semantics into consideration
    - regulate congestion control / retry / ordering behaviors based on newly carried L3/L4-like semantics
    - develop a new MAC client interface satisfying the expectation from the revised MAC header

Q&A