Unified Forwarding Plane for Enhanced Services

NIPAA - Session C

https://icnp20.cs.ucr.edu/nipaaprogram.html

Time: **Oct. 13, 2020, 3:30 p.m. to Oct. 13, 2020, 5 p.m. (CEST)**

Toerless Eckert, Toerless.Eckert@futurewei.com, Futurewei

Stewart Bryant, sb@stewartbryant.com, Futurewei & Univ. Surrey

Abstract

Research into the evolution of IP forwarding is proposing enhanced end-to-end, per-hop services in support of requirements by current and even more so future applications, such as high-precision latency, qualitative and deterministic forwarding.

When such an evolved IP forwarding plane would be were deployed In a similar manner to as todays IP/IPv6 forwarding plane, these enhanced services would not be able to operate on every hop though, because todays network paths are built as a mix of L2 ethernet witching, L2.5 MPLS switching, L3 routing and various forms of "tunneling" especially in mobile networks.

A more fundamental study is therefore needed to understand how the new requirements are best met, and how we can migrate the network from the network of today to a network that can meet the new needs.

This talk is about ongoing work, it is meant to inspire attendees to think bout this area
no results will be presented

New Services require New Forwarding

- 25 years ago: IPv4 -> IPv6
    - *and all we got are more addresses ?!*
    - Other foundational goals of IP-NG effort leading to IPv6 not adopted with IPv6
    - Fundamentally: 50 year old protocol / services model
- What Future IP Evolution (FIPE) do we need ?
    - IPv6 Forever ? New IP ?
    - Will never know ... until we explore the options and compare:
        1. Best option with IPv6 (IPv6 “Law”: RFC8200) + Extensions
            - See IETF work on source routing extension headers: Small changes, decade of work...
        2. Best option for packet headers/design without these constraints (NewIP)
    - Compare
        - Likely: When urgent need and feasible: improvements will be “hacked” into IPv6, but “cleaner” / “better”? solution possible when not constrained by RFC8200 limits

“Network Layer” (Hop-by-Hop) Services

actually: applies to (L2, L2.5, L3, L3.5), not only “network layer”

Services are Functions of the forwarding plane that…

1. May happen on every hop supporting the service (hop-by-hop)
2. Are indicated / desired by the endpoints (user) (?!!!!)
3. Impact / Differentiate the traffic experience
   absolutely or relative to other traffic

Distinction “Service” vs. “Function” example:
Traffic Steering / Source Routing:

- Widely used in SP networks for “Capacity Optimization”:
  Networks steer traffic flows load-based around different, non-equal cost paths.
  Maximizes Return of Investment (RoI) of network infrastructure
- If applied to all traffic equally traffic steering is just a “function” as it does not meet 2, 3.

- If different user traffic would indicate different goals resulting in different steering results, then the steering becomes part of a service:
- Example: “low-latency” == steer across ”shortest path with metric=latency”:
  less bandwidth but lower latency

IP Services

- 8 bit TOS/TC = (DSCP, ECN) field in IPv4/IPv6 header. 30++ year old architecture
- ECN (2 bits in TOS field): Fast congestion signaling
  - Limited, challenging service model
- DiffServ (6 bits of TOS field):
  - Per-packet services
  - Locally significant indication of "traffic class". Limited standardization of semantics, no use outside of "controlled" networks. Many abuses
- IntServ (Integrated Services)
  - Per-flow service (flow = IP 5-tuple)
  - Service established through signaling. SDN or on-path (RSVP, {NSIS})
  - Only option today to support Guaranteed Services for traffic flows
    - Guaranteed Bandwidth, Guaranteed maximum end-to-end latency
    - Very little adoption / deployment
    - IETF DetNet effort raises need to support this again
- No fundamental recent Stds. improvements of this ca. 25 year old framework
  - Many signaling refinements to RSVP in the last 20 years, some new DSCP classes too
  - But nothing beyond "classes"

New (IP Services) – our vision, **How**: Contract header / BPP

- New header in packets to specify flexible extensible services
  - Tentatively called “Contract” header
  - Ongoing prototype work for New IP header architecture

- Rethink encoding/programming of services packet header:
  - Service expressed through action commands with parameters in header
  - Additional metadata: shared data elements not tied to individual actions
  - Explicit indication of sequentialization / parallelization of actions
  - Multiple encoding options for same functional outcomes
  - “Statelets” for “per-flow” actions
    - BPP header as “in-band/in-packet” signaling replacing RSVP
  - Scalability beyond flows with per-packet services/actions
    - E.g.: SP core networks would require millions of flows

New (IP Services) – our vision, **What**:
High Precision Communication services

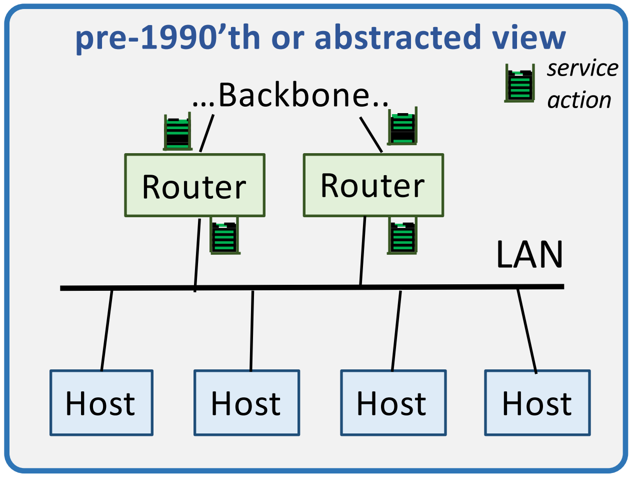
- Beyond 8 bit “QoS“ in packets
  - what could/should services be: existing / new research how would/could it be expressed in network header:
- Congestion Control (CC)
  - ECN vs. PCN - currently use same bits to signal, but deliver different experience.
  - More than 2 ECN bits for faster CC level signaling (LIVE research paper)
  - Upspeeding indication (minimum free path bandwidth ?!)
- Throughput
  - Explicit signaling associated with difference in target bandwdidth e.g.: as possible in NADA (weight)
  - signaling of flow-bandwidth for stateless weighted fair queuing (DPS research 2010)
- Loss
  - Signaling of discard priority (known benefits for video, e.g.: I/P/B differentiation – payload encrypted.
  - Signaling of flow copy MRT/Live-Live high-resilience deployments. sub-flow ?
  - Sequence number (as required by PEROF DetNet, not existing for IP)
- Latency
  - latency SLO , end-to end min/max, experienced latency, (see NOMS LBF paper)
  - signaling of experienced latency for better per-hopp-functions
  - discard packets early when they are too late (not based on hops)

Side note
New (IP Services) – our vision, more

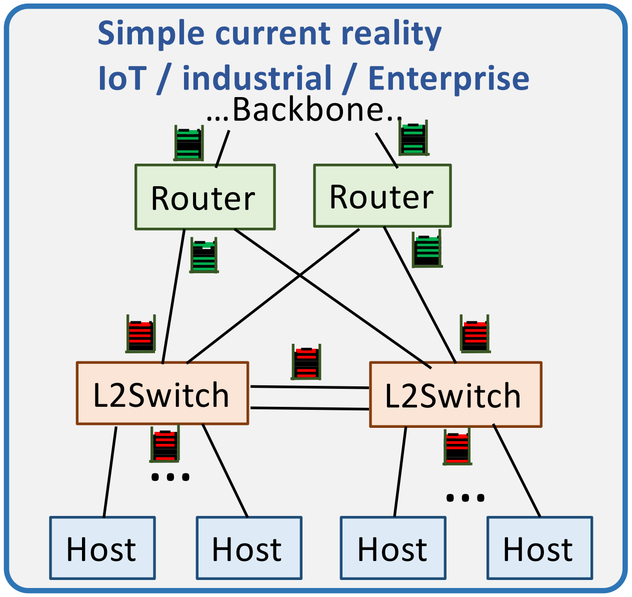
- For reference (Not relevant for rest of presentation)
- Qualitative Communications:
    - Beyond per-packet contract headers to indicate different services for different chunks in packets
- New path steering architecture (PPR)
- TBD: Unified destination representation (beyond unicast)
    - Send packets to individual-destination, named-group-of-destinations, member-of-a-group, enumerated-group-of-destinations, …
        - Aka: unification of current “historically evolved” approach in IP to deal with unicast, anycast, multicast, BIER – and beyond

Most Networks are multi-layer

- And QoS/service-actions across different layers vary widely today
    - See e.g.: mapping DetNet (IntServ) ⇔ TSN
- IP DSCP =~ Ethernet COS =~ MPLS EXP
- Sufficiently different to make any deployment painful when more than Best Effort desired
- Mapping == lowest-common-denominator
- Main "working" deployment model
    - Lower layer QoS is just "aggregate", "coarser" service than higher layer
    - But this model can not provide high-precision across lower-layer hops



pre-1990'th or abstracted view

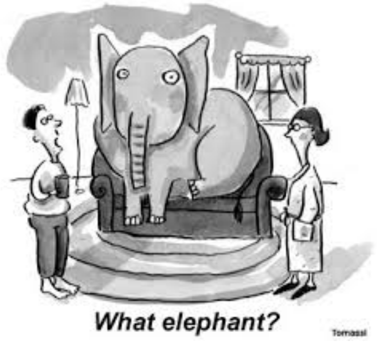


Simple current reality
IoT / industrial / Enterprise

Multiple forwarding layers:
~~Ignore~~ / eliminate / embrace ?

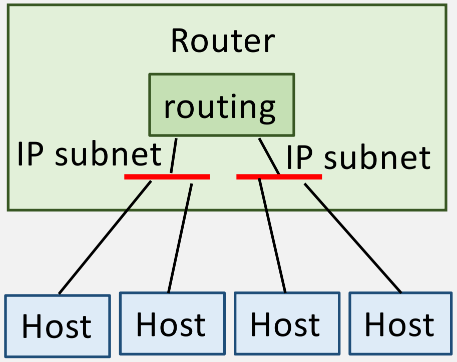
- IP/IP6 towards hosts require “Ethernet LANs”
  - Separate IPv6 subnet per host unmanageable
  - IP/IPv6 subnet with more than 1 host or more than 2 routers have “L2 switching” requirements
    - Even when they are hidden inside a single box called router
- Ignore: (Personal) grievance from 1980th… 1990th :
  - ISO CLNP was designed to eliminate need for L2 switching
  - 1990th IP-NG could have adopted this. But IPv6 did not!
  - *CLNP address (NSAP) “host routes” instead of L2-MACaddr “routes”*
    - *ES-IS NSAP addr assignment, IS-IS L1 “host-routes”*
- Eliminate: Can be done with IP(v6) as well… but challenging
  - IPv6 link local address space =~ LAN, but adds no benefits of MACaddr, but adds IPv6 complexities
  - Ethernet is attractive because of auto-addressing, auto-config / switching. Not defined for IPv6!
  - Some IPv6 network architectures use host-routes: IoT/RPL, Autonomic/ACP/RPL
- Embrace: Forwarding protocol with multiple address sizes could be reused across layers
  - ‘L3’ instance using e.g.: 128 bit addresses – or 32 bit addresses (why longer than needed) !
  - ’L2’ instance using e.g.: **48** bit addresses (or 16, 20, 24**,** 64…)
- If/where we embrace multiple forwarding layers:
  - How to deal with services and their packet headers ?

Please do not ignore the elephant!



What elephant?

Every access router is a mixed L2/ethernet + L3 IP router device



IP subnets ~= VLANs =~ L2 switching

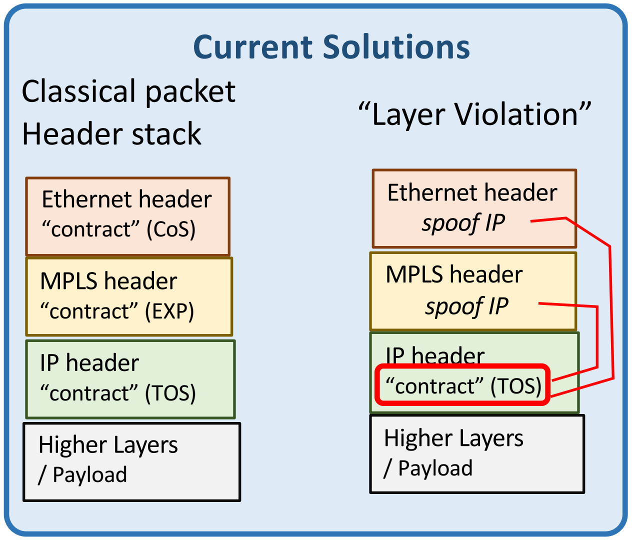
Multiple layers … and Contracts / Services

- Protocols evolved from solving different problems providing different services.
  - Ethernet… MPLS... SRv6 .. IP/IPv6
- Historically, protocols where designed to do as little as necessary
  - Solution for the problem where pre-existing protocols have a gap
  - 'Simply' layer multiple protocols to solve multiple problems
  - No Black & white: often good, often problematic
- Main challenge: re-inventing High Precision Services for every layer
  - Repeatedly solving the same problems... Incompatible to each other
  - Such as per-hop actions for (QoS / high precision communication) services
  - Duplicating efforts
  - Slowing down adoption of those services
- Explore what a future forwarding plane protocol needs to support so it can be re-used across multiple layers easily
  - Needs to have a superset of services required at different layers
  - Should be backward compatible/interoperable for migration/investment protection

“Contract” header options

- Todays “correct” / “dogma”
  - Every layer has it’s own contract header
  - Mapping, duplication of standardization, loss of functionality,…
- Todays “Layer violations” / “pragmatic”
  - Spoof higher layer contract.
  - E.g.: per-IP-flow actions in Ethernet switches, MPLS LSR

What are better options to explore ?


Current Solutions

Classical packet Header stack
“Layer Violation”
Ethernet header “contract” (CoS)
MPLS header “contract” (EXP)
IP header “contract” (TOS)
Higher Layers / Payload
Ethernet header *spoof IP*
MPLS header *spoof IP*
IP header “contract” (TOS)
Higher Layers / Payload

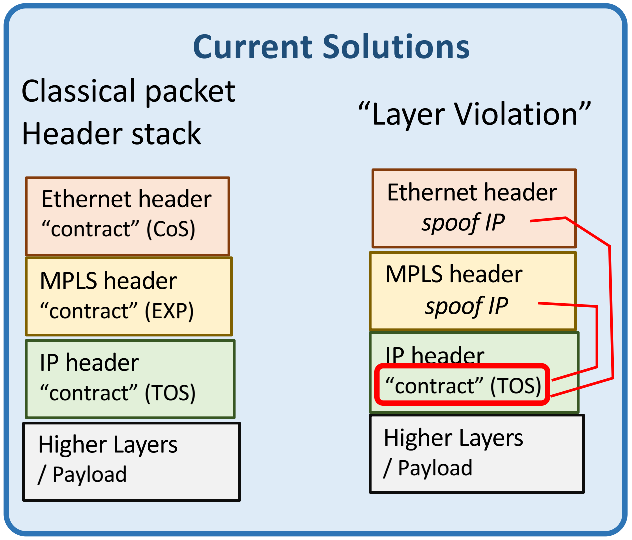
“Contract” header options

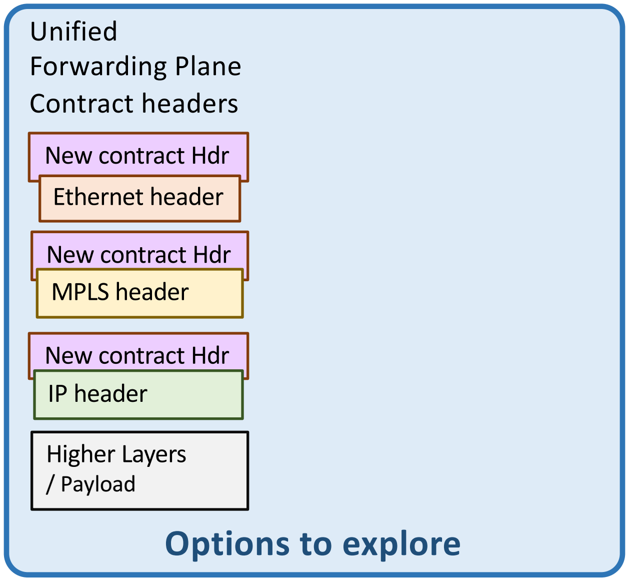
- Todays “correct” / “dogma”
  - Every layer has it’s own contract header
  - Mapping, duplication of standardization, loss of functionality,…
- Todays “Layer violations” / “pragmatic”
  - Spoof higher layer contract.
  - E.g.: per-IP-flow actions in Ethernet switches, MPLS LSR

What are better options to explore ?

(1) New reuseable contract headers

- Support superset of service aspects of existing layers. Each layers instance may just be drop-in replacement of existing layer contracts for backward compatibility
- Benefit: allow definition of new contracts (e.g.: “DetNet” once – reuse same contract at L2, L3). Eliminate new work duplication
- Does not eliminate mapping betwen layers





“Contract” header options

- Todays “correct” / “dogma”
  - Every layer has it’s own contract header
  - Mapping, duplication of standardization, loss of functionality,…
- Todays “Layer violations” / “pragmatic”
  - Spoof higher layer contract.
  - E.g.: per-IP-flow actions in Ethernet switches, MPLS LSR

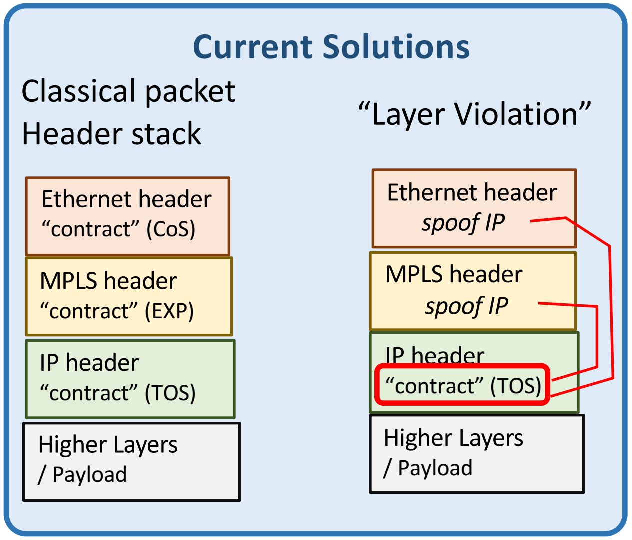
What are better options to explore ?

(1) New reuseable contract headers

- Support superset of service aspects of existing layers. Each layers instance may just be drop-in replacement of existing layer contracts for backward compatibility
- Benefit: allow definition of new contracts (e.g.: “DetNet” once – reuse same contract at L2, L3). Eliminate new work duplication
- Does not eliminate mapping betwen layers

(2) Cross-layer semantic for contract header

- Single contract header can be sufficient
- Each node determines at which layer (e.g.: ethernet, MPLS, IP it operates), but uses same contract heder for hop-by-hops services
- Ideal when multiple layers used within single administrative entity (e.g.: IoT, Industry, Enterprise).



Current Solutions

Classical packet
Header stack

Ethernet header
“contract” (CoS)

MPLS header
“contract” (EXP)

IP header
“contract” (TOS)

Higher Layers
/ Payload

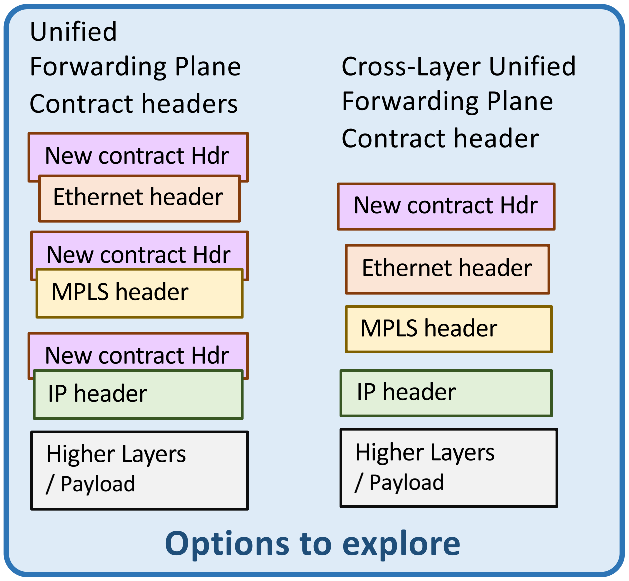
“Layer Violation”

Ethernet header
spoof IP

MPLS header
spoof IP

IP header
“contract” (TOS)

Higher Layers
/ Payload



Unified
Forwarding Plane
Contract headers

New contract Hdr
Ethernet header

New contract Hdr
MPLS header

New contract Hdr
IP header

Higher Layers
/ Payload

Cross-Layer Unified
Forwarding Plane
Contract header

New contract Hdr

Ethernet header

MPLS header

IP header

Higher Layers
/ Payload

Options to explore

Full cross-layer re-useable packet headers / protocols

- Learn from IPv6/SRv6 experience:
  - SRv6 is attempt to reuse IPv6 at the layer previously occupied by MPLS/SR-MPLS
    - “Network Layer”, compared to “Internet Layer” (IPv6 Internet end-to-end)
  - Challenged by inheriting all aspects of Internet IPv6 into this layer… even when they do not match well:
    - Large header overhead, long addresses, limited services (only packet steering, no better QoS). (“Hackery” to shorten address steering header ongoing in IETF...)
- Creating a fully cross-layer reuseable protocol :
  - More research / experimentation required
  - Separating out the (QoS, High-Precision) Contract part and make that reuseable may be a good first goal!
  - Replacing the “base header” (MPLS, Ethernet, IP) with common header requires variable length addresses, but also a lot more thought on migration, integration, optimization, ...

Summary / Conclusions

- Multiple per-hop forwarding layers are common today
  - Sometimes beneficial, sometimes undesirable but today unavoidable
- In a NewIP world these should be eliminated where not beneficial
- A NewIP should be able to operate at multiple layers where beneficial
- Per-hop services are ideally common across per-hop forwarding layers
- Contract / Services packet headers ideally shareable across layers
- Solving these challenges could be key for success of a New IP

The End

- Thank you for listening
- Please reach out to us if you are interested in this topic
  - Emails on title slide
- Some References for further reading on next slide (sorry, incomplete).

References

NewIP Packet headers:

https://github.com/network2030/BPP (2020)

- Repository of papers about New IP contract header research papers.
- Also first research paper (NOMS 2020) about “Latency Based Forwarding”, a hop-by-hop service for Latency based SLO (Service Level Objectives), stateless latency management.

Differentiated / high precision traffic treatment (incomplete)

https://github.com/network2030/lbf-poc (2020)

- Source code for LBF validation prototype

https://tools.ietf.org/html/draft-stoica-diffserv-dps-02 (2003)

- Dynamic Packet State (DPS) – stateless, per-flow weighted fair bandwidth shaping.

https://www.rfc-editor.org/rfc/rfc8698.txt

- NADA – Real time traffic congestion control with support of per-flow differentiated weights.

Layered Internet Video Engineering (LIVE): Network-Assisted Bandwidth Sharing and Transient Loss Protection for Scalable Video Streaming, 2010 Proceedings IEEE INFOCOM

- Better than ECN congestion signaling

IPv6 routing solutions without need for (switched) L2 subnets

RFC6550 (RPL), https://tools.ietf.org/html/draft-ietf-anima-autonomic-control-plane