Meta-Headers: Top-Down Networking Architecture with Application-Specific Constraints

Murat Yuksel
University of Nevada, Reno
Reno, NV

yuksem@cse.unr.edu
http://www.cse.unr.edu/~yuksem
Motivation: The trends

- The variety of applications possible is increasing, especially in wireless
  - wireless peer-to-peer, mobile data, community wireless

- The size is increasing:
  - million-to-billion nodes

- The dynamism is increasing:
  - vehicular networks, sensor networks, MANETs

- What is unavoidable?: More dynamism, more disruption tolerance, more entities, and more varieties
### Motivation: The big picture

#### (a) OSI
- **Application**
  - Presentation
  - Session
  - Transport
  - Network
  - Data Link
  - Physical

#### (b) Wireline
- **Application**
  - Transport (TCP, UDP)
  - Network (IP)
  - Data Link (Ethernet 802.3)
  - Physical (Fiber, Cable)

#### (c) Wireless
- **Application**
  - Transport (TCP, UDP)
  - Network & MAC (IP, Mobile IP, 802.1x)
  - Physical (RF, Fiber, Cable)

#### (d) MANET, peer-to-peer
- **Application**
  - Network & Routing
  - Physical (RF, FSO, Fiber, Cable)

---

**Economics always has the bigger force: economically attractive applications will keep forcing more vertical components into the stack!**

---

**We need a systematic way of implementing vertical components to avoid an unhealthy monolithic stack architecture.**
Motivation: Response to the trends

- Wireless research has been responding with
  - optimizing via cross-layer designs
  - adding custom-designed vertical components to the stack
  - Old hat: layered vs. cross-layer tradeoff

- **Bottom-up cross-layer** has been the main approach
  - Scarcity of wireless resources dominated the economics
Motivation: Response to the trends

- A paradigm shift: wireless resources are becoming massively available
  - Community wireless
  - WiFi hotspots
  - Google WiFi, AT&T Metro WiFi
- Spectrum resources may still be scarce but connectivity is already ubiquitous
- The key metric to optimize is becoming application utility rather than the wireless resources
- App-specific vertical designs are needed..

We need top-down cross-layer designs in addition to the traditional bottom-up ones.
Why not continue merging layers?

- Merging layers:
  - A greedy approach
  - Makes it hard to standardize - bad for sw engineering

- Which layers must be absolutely isolated?
  - Application, Network, Physical?

- Integrating lower level functions with a higher layer function will prevent them becoming a substrate for other higher layer protocols
  - Cellular provisioning in the US - jailbreaks
Motivation: Application Layer Framing (ALF)

- Layering was a main component of the e2e architecture.

  “a major architectural benefit of such isolation is that it facilitates the implementation of subsystems whose scope is restricted to a small subset of the suite’s layers.”

  Clark and Tennenhouse, SIGCOMM’90

- But, Integrated Layer Processing (ILP) was there too!
  - To achieve better e2e efficiency and resource optimization
  - ILP never become a reality due to the lack of a systematic way of doing it.

- An ALF-based approach is needed:
  - network protocol services at lower layers can best be useful when applications’ characteristics and intents are conveyed to the lower layers.
Meta-Headers: A vertical design tool

- A packet meta-header:
  - vertically travels across the network stack
  - establishes a vertical communication channel among the traditional layers
  - co-exist with the traditional per-layer packet headers

- Applications can communicate their intent across all the protocol layers by attaching the meta-headers to data.
  \(<\text{meta-headers, message}>\)
Headers vs. Meta-Headers

Explicit Meta-Headers

Implicit Meta-Headers

IEEE GLOBECOM FutureNet, Miami, FL, Dec 2010
Meta-Headers: Demultiplexing

Demultiplexing with traditional headers

Demultiplexing with meta-headers

Layer 3

Layer 4

Protocol 1

Protocol 2

H4 MH4 MH3 MH2 MH1 message

H3 H4 MH4 MH3 MH2 MH1 message

H4 MH4 MH3 MH2 MH1 message

Service 1

Service 2
Informing Applications about Lower Layer Services

- How will upper layers know about the service primitives of the layers lower than the one below?

- Reactive - Meta-Headers in Reverse Direction
  - detect lower layer services in an on-demand manner as connections arise
  - meta-headers rewritten by lower layers in reverse direction
  - Requires a closed-loop - connectionless or multi-receiver services may not work
Informing Applications about Lower Layer Services (cont’d)

- Proactive - Pre-informed Designer
  - inform layer k designers about services of layers k-2 and below apriori
  - too much complexity as the number of lower layer services increases - rank ordering might help
  - May not be desirable by ISPs
  - Regional service discovery via broadcasting - connectionless
End-to-End Coordination

1. Application at source prepares meta-headers with default options and sets flags to probe for available services.

2. Meta-headers may or may not get converted to traditional headers.

3. Meta-headers are filled with summary of available end-to-end L1-L4 services, and optionally fed back to the source application.

4. Meta-headers are filled with summary of available end-to-end L1-L4 services, and fed back to the source application.

5. Application at source readjusts meta-headers for joint vertical optimization of end-to-end performance.

Optional feedback loop for conveying available L1-L3 services.

Optional feedback loop for local optimization of last hop(s) of the end-to-end path.

Feedback loop for conveying end-to-end multi-hop L1-L4 services, possibly as a sequence of options over multiple hops.

A dynamic end-to-end negotiation.
An optimization perspective

Vertical optimizations are possible

More dynamic

Meta-headers as Lagrange multipliers
Summary

- A top-down networking architecture with meta-headers
- Vertical optimizations at finer temporal and spatial granularity
- A variety of top-down optimizations:
  - Top-down routing (layers 5, 3)
  - Top-down QoS/value management (layers 5, 3, 2)
  - Top-down dynamic transport (layers 4, 3, 2)
- A new class of optimization problems aiming to improve joint performance of multiple layers while respecting the isolation among them.
THE END

Thank you!

This work is supported in part by the U.S. National Science Foundation awards 0721600 and 0721609.
An optimization perspective

Vertical optimizations are possible:

More dynamic

Meta-headers as Lagrange multipliers