A Vision for Explicit Path-Cooperative Transport

Mirja Kühlewind and Brian Trammell, ETH Zürich
Joe Hildebrand, Cisco Systems

Innovations in Clouds, Internet, and Networks
Paris, 1 March 2016

mami
measurement and architecture for a middleboxed internet

measurement | architecture | experimentation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421.The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information.

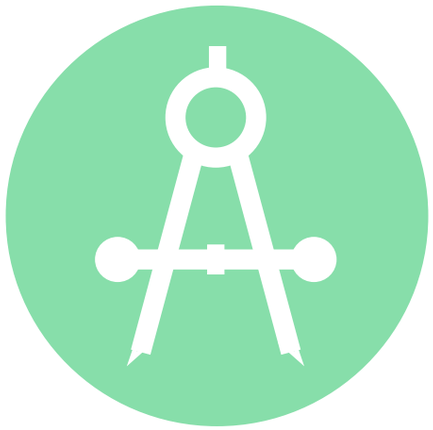
Supported by the Swiss State Secretariat for Education, Research and Innovation under contract number 15.0268. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the Swiss Government.

The MAMI Project

Measurement and Architecture for a Middleboxed Internet

measurement

of deployed middleboxes



architecture

for middlebox cooperation

experimentation

of use case applicability and deployability

- Strong interaction with relevant standards organizations for impact on deployment
- FIRE testbed (MONROE) support for measurement as well as experimentation, especially on mobile broadband access networks
- Learn more at **http://mami-project.eu/**

Overview

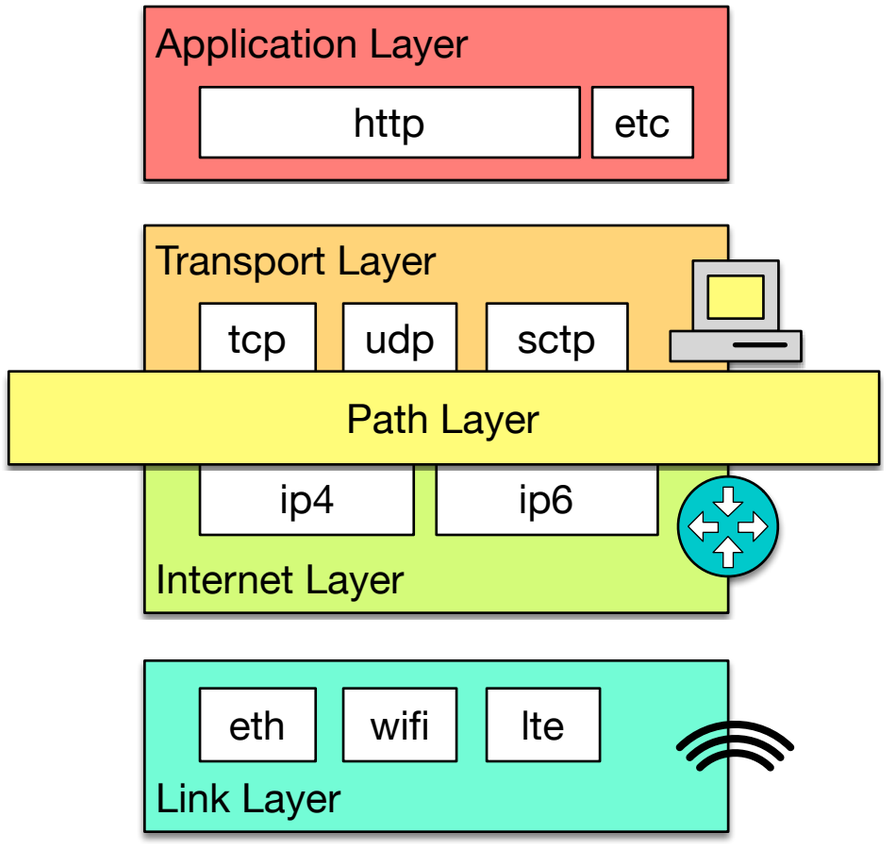
- Why do we need explicit middlebox cooperation?
- Why do we need a shim layer for this?
- How do we have to design the protocol to make it deployable?

Why explicit middlebox cooperation?

A. Deployment problems of new protocols and protocol extension due to ossification in the Internet, e.g.
   - Multipath TCP
   - QUIC (over UDP)

B. Operation and management of in-network functionality hindered due to increasing deployment of encryption, e.g.
   - firewalls using port mapping or DPI
   - performance enhancements in mobile networks

Why a new shim layer?



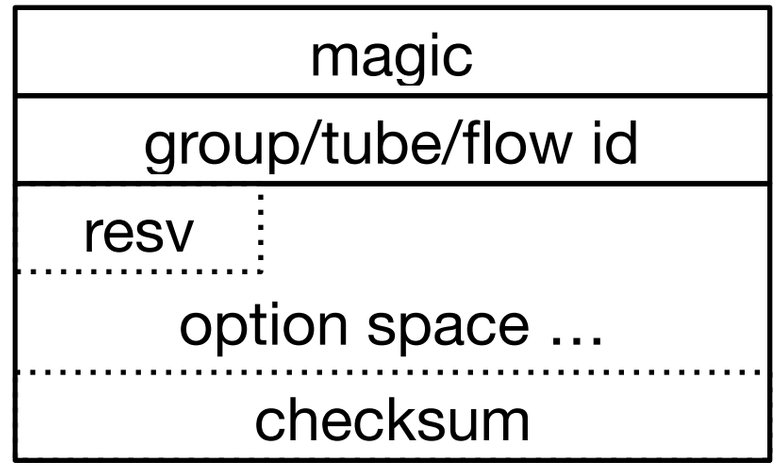
- Transport layer: end-to-end sockets
  - flow information
  - stateful and s g
  - at the
  - p handling
  - rmation
  - ss and simple processing in the middle

Missing:
Per-flow information for stateful in-network functions

➡ **Path layer** for explicit cooperation with middleboxes instead of implicit assumptions

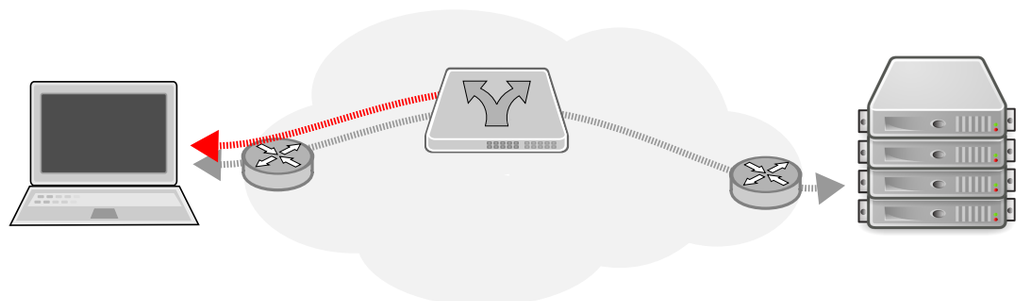
Path Layer:
(Basic) Functional Requirements

- Grouping of packets into flows



- Extensibility to provide per-flow network information

- Explicit feedback channel from middlebox to endpoint



Why should I trust what you say about your flows?

- **Default**: *trust but verify*
  - declarative signaling: **no** negotiation, **no** guarantees
  - the best way to prevent cheating is to make it useless to do so
  - minimize the information exposed!

- Leverage existing trust relationships for higher-assurance declarations
  - e.g. your enterprise firewall, access network middleboxes, etc.

Example 1: Firewall Traversal

Problem

UDP often blocked as it is hard to maintain state

Needed

- group ID
- start/stop signal and confirmation by receiver (‚SYN/ACK’)

Action

- firewall can forward first packet and set up state based on confirmation from receiver
- group ID must be large enough to not be guessable

Example 2:
Low Latency Support

Problem

Network service not optimized for latency sensitive traffic

Needed

Flag to signal loss sensitivity vs. latency sensitivity

Action

- network device can treat latency sensitive traffic differently, e.g. in a separate smaller queue
- trade-off between loss and latency gives no incentive to lie

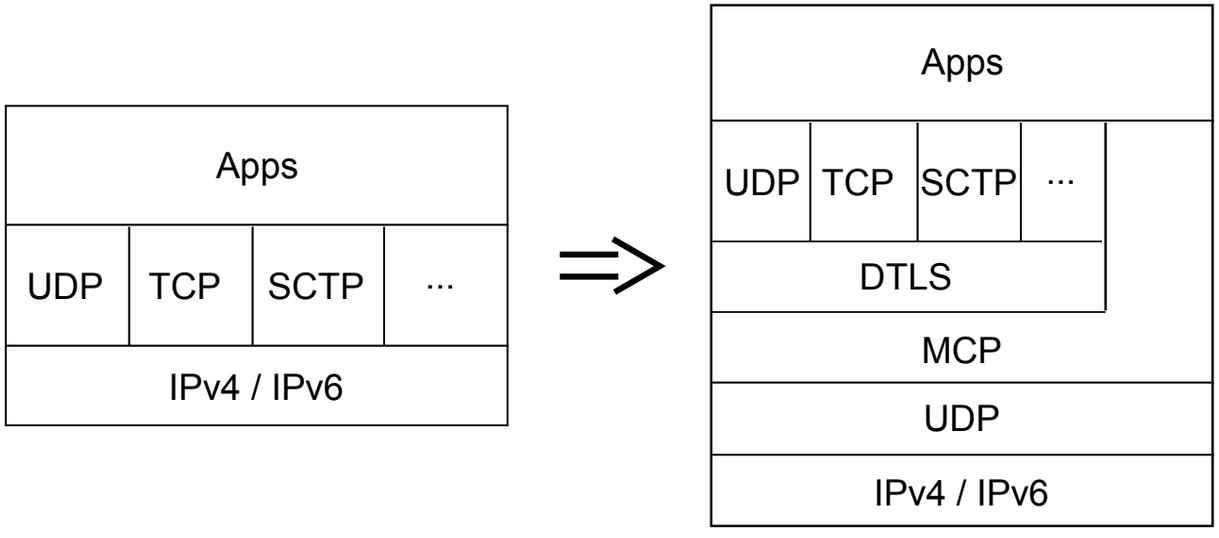
Will it deploy?

- Transport-layer **encapsulation over UDP**
  - Need ports for NAT
  - Impossible to deploy with new protocol number across the Internet
  - Userspace (and kernelspace) implementation possible
- **Magic number** for easy recognition, protection against reflection
- **Flags** for “SYN/ACK” condition for state decision delegation to endpoint
  - All traffic bidirectional
  - Data in first packet possible
- Signals fit in a single packet (**no segmentation or reliability**)
- **Checksum** for error detection, cryptographic integrity checks available

Implementing an Explicit Path Interface

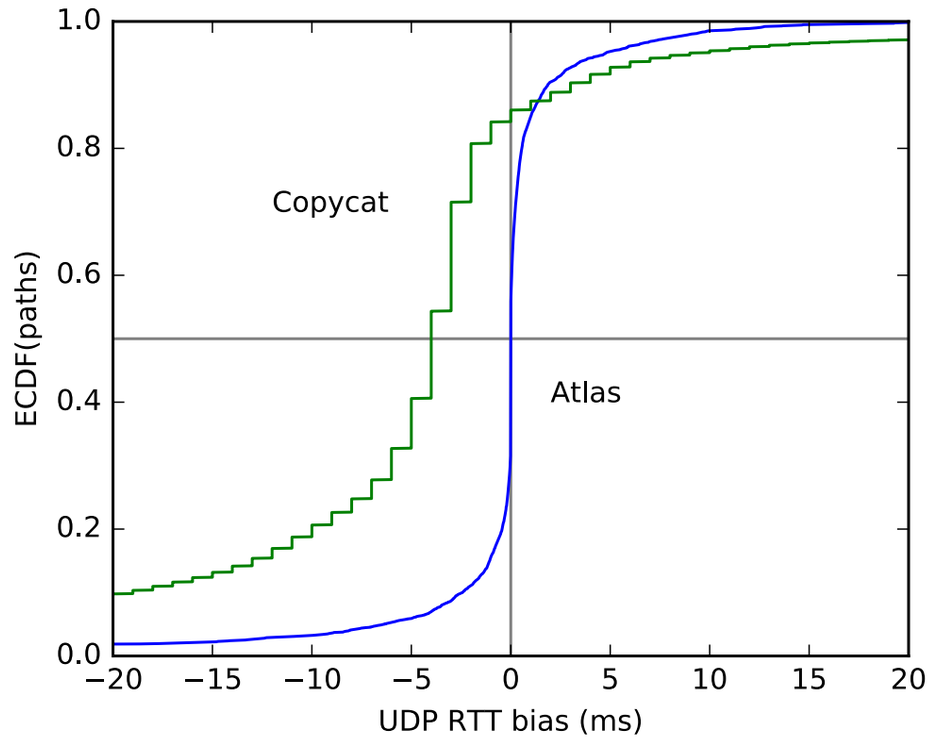
- Application can directly indicate requirements to path layer
- Transport can use the path layer to expose parts of its functionality/ intentions to the network
- *Middlebox Cooperation protocol* (MCP) signals these information appropriately to on-path middleboxes

➡ **Minimize the information exposed!**



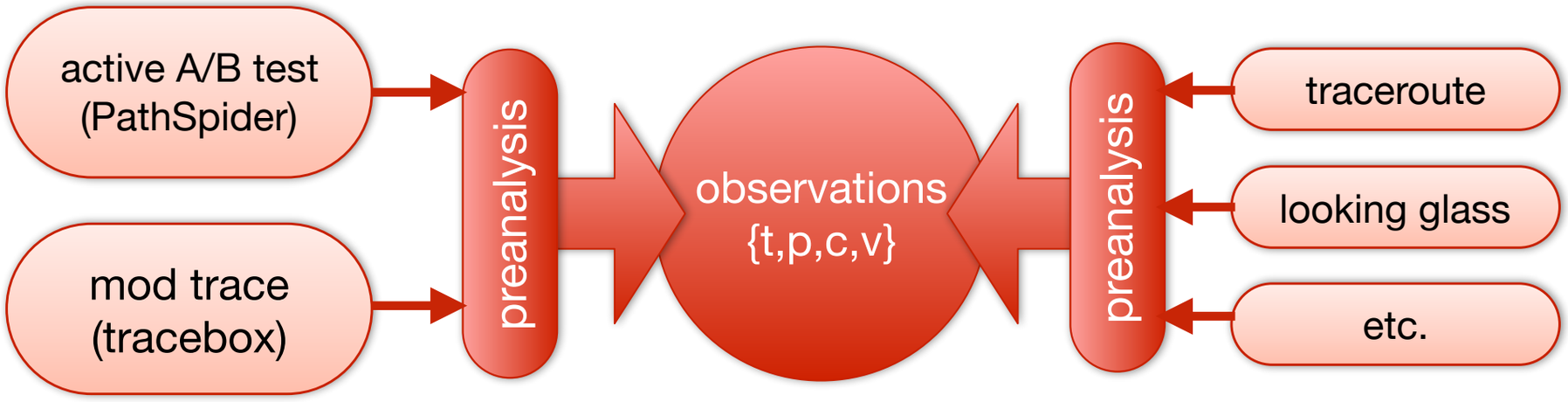
Is it possible to run the Internet over UDP? Preliminary Results

- A/B testing for TCP/UDP connectivity
  - Copycat tool on 120 PlanetLab nodes
    - 3,67% UDP blocking on port 33435
    - 2,7% UDP blocking on all tested ports (33435,1228, 8008, 12345)
  - RIPE Atlas traceroute
    - 3.661% UDP blocking based on existing traceroutes
- We are currently running more measurements!
  - Use all existing testbeds available, e.g. CAIDA Ark, MONROE
  - Other impairment measurements: TCP Options, SCTP, …



Path Transparency Observatory

- Observatory (public release end 2016) to derive common ***observations*** about *conditions* on a given *path* at a given *time*
  - Active measurements, made by the project
  - External measurements (e.g. traceroutes, BGP, traces)
- Combining disparate measurements leads to better insight
  - How likely is it that a certain path impairment impacts my traffic?



Follow http://mami-project.eu for updates on data model & availability!

References

- Substrate Protocol for User Datagrams (SPUD) in the IETF: spud@ietf.org
  - draft-trammell-spud-req
  - draft-kuehlewind-spud-use-cases
  - draft-hildebrand-spud-prototype
- IAB Stack Evolution Program
  - Workshop on Stack Evolution in a Middlebox Internet (SEMI) 2015 [RFC7663]
  - B. Trammell, J. Hildebrand: Evolving Transport in the Internet
- IRTF research group on Measurement and Analysis for Protocols (MAPRG): maprg@irtf.org
- MAMI webpage (mami-project.eu) or twitter (@mamiproject)