Measurement-Driven Protocol Engineering

IAB Technical Plenary, IETF 94 Yokohama

Brian Trammell, IAB Stack Evolution Program

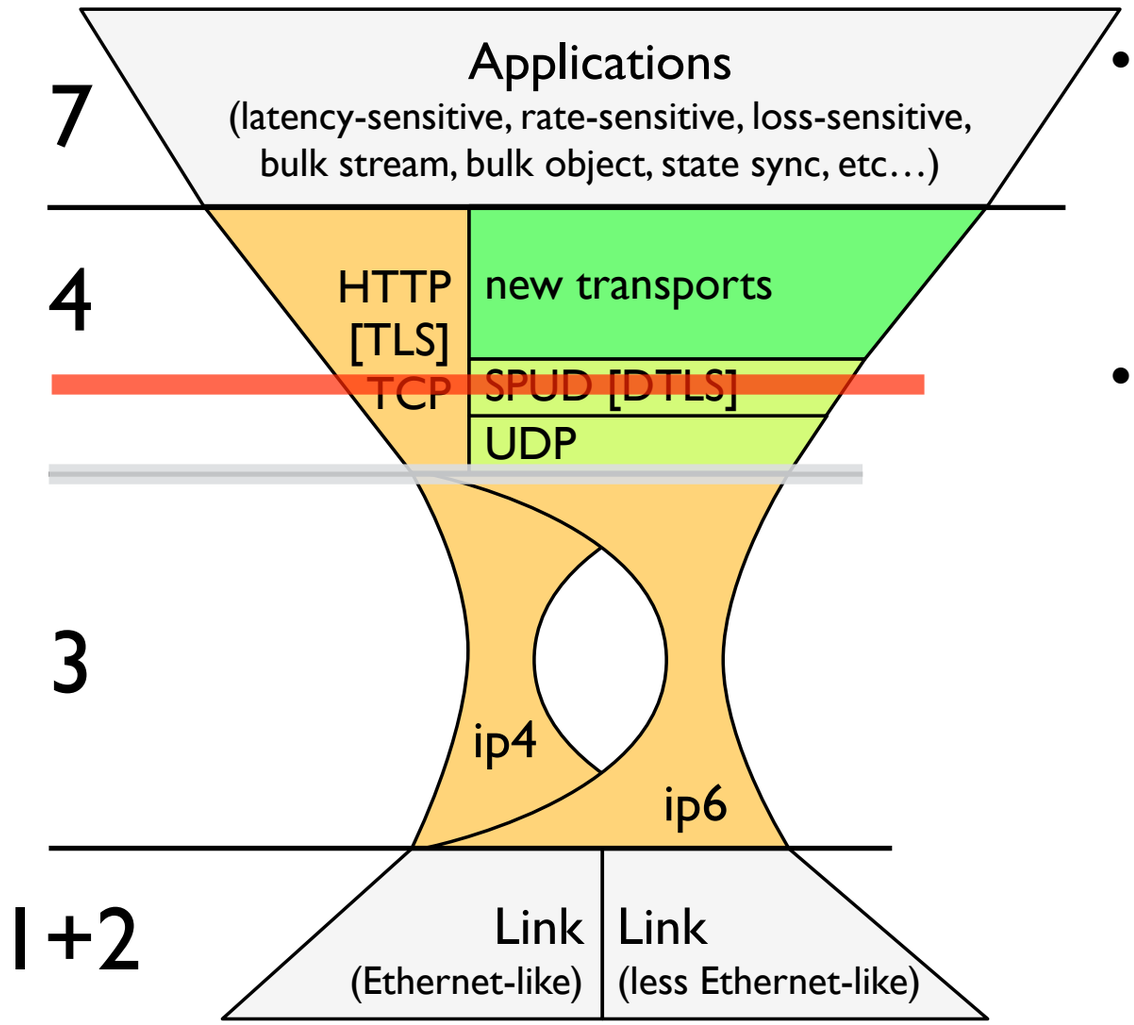
Today's talks

in one slide

- Engineering decisions about protocols to deploy in the Internet should be based on relevant data about the environment they face.
  - Design for common occurrences.
  - Know the risks of uncommon ones.
  - Apply measurement liberally to know the difference. Maybe even at runtime.
- Two areas of application:
  - IP stack evolution and path impairment
  - Understanding the interdomain topology of the Internet
- Discussion: principles of applied measurement?

IP Stack Evolution and Path Impairment

Evolving the stack:
explicit relayering and cooperation



- Goal: support deployment and user-space experimentation of new transport protocols in today's Internet.
- Approach: rethink the layer boundary
  - UDP encapsulation (ports for NAT)
  - crypto (reinforce the boundary between endpoint and path visible headers)
  - explicit cooperation (give back transport and application semantics the path actually needs)

Will any of this actually work?

Measuring path impairment

- Path impairment: the likelihood that traffic with given characteristics will experience problems on a given path.
  - Increased latency, reordering
  - Increased loss/connectivity failure
  - "Bleaching" or selective disablement of features
- Utopian goals:
  - given a proposed protocol feature, know the prevalence of different types of problems with that feature on different networks.
  - given a source and destination, know the types of protocol features that will work along the paths between them.
- First step: sharing what we know. **HOPSRG** (hops@ietf.org)

What can go wrong?

Modification	Planetlab	Ark
NAT	**74.9%**	**79.0%**
ECN IP	**13.7%**	**13.2%**
ISN	**10.7%**	1.8%
MSS	**10.8%**	5.9%
Exp. Option	8.8%	0.5%
MPCAPABLE	8.4%	0.3%
ECN TCP	0.6%	0.6%
SackOK	0.3%	0.0%
TS	0.3%	0.4%
WS	0.2%	0.2%

- NAT everywhere
- Many features mostly work
- Variation based on vantage point
- **Best studies look at O(10k) paths[1].**

[1]: R. Craven, R. Beverly, M. Allman. **A Middlebox-Cooperative TCP for a non End-to-End Internet**. SIGCOMM, August 2014.

Application to Stack Evolution

- We want our protocols to work when stuff breaks.
  - Engineering tradeoff: robustness against path conditions vs. robustness of implementation.
    - NAT? Design for it, even if it's hard.
    - Broken by common operational practices? Depends if they’re relevant.
    - Conflicts with a custom hack deployed in one network? Write a polite email, but no code.
- Need data to evaluate this tradeoff.

Measuring the Internet is hard

- Measurements often don't measure what you want.
  - e.g.: ICMP latency and connectivity correlate less than we'd like with application latency and connectivity.
- The Internet is not homogeneous.
  - e.g. how much crypto you see on a given link depends on application mix and the vagaries of CDN policy[2].
- Selection bias: what is easy to measure is not necessarily most relevant.
- Tradeoffs with visibility versus (business) confidentiality and (user) privacy.

[2]: P. Richter et al. **Distilling the Internet's Application Mix from Packet-Sampled Traffic**. PAM, March 2015.

Step 1: Improving the best available data

- We have lots of **tools**…
  - platforms and testbeds (e.g. Atlas, mLab, Ark, BisMARK, SamKnows, PlanetLab…)
  - protocols (e.g. O/TWAMP, PSAMP, IPFIX, LMAP)
  - …but lack a framework to bring **comparability** and **repeatability** to their observations.
    - Common information models for noting different observations mean similar things.
    - Common measurement control and query protocols[3].
- Goal: combine smaller studies and measurement surfaces into a larger, more comprehensive body of knowledge.

[3] e.g. mPlane, ict-mplane.eu, draft-trammell-mplane-protocol

Step 2: Measuring without measuring

- Lots of things that don't look like measurement actually are.
  - TCP measures itself as a side-effect of its operation.
  - Version negotiation and fallback mechanisms (e.g. MPTCP, WebSockets) generate data about where features work.
  - Platform-level diagnostics[4] a potentially rich source.
- Let's design protocols with this fact in mind.
  - Extend common information models to runtime logging.
  - Add explicit measurement primitives to protocols.
  - Exploit what we've learned from doing it the hard way.

[4] e.g. telemetry.mozilla.org

Understanding Topology

Discussion

Principles of applied measurement?

- There are many other insights to be gained from the Internet by measuring it in different ways.
  - Integration of measurements from different sources (active measurements, passive observation, application logs) can lead to more insight.
- Questions to ask:
  - What **assumptions** about the environment is protocol X based on? Do these hold?
  - What **sources** already exist that allow me to verify these assumptions?
  - What information does the protocol **generate** as a side effect that can lead to better insight?