Measurement-Driven Protocol Engineering

IAB Technical Plenary, IETF 94 Yokohama

Brian Trammell, IAB Stack Evolution Program

Measurement-driven engineering
in one slide

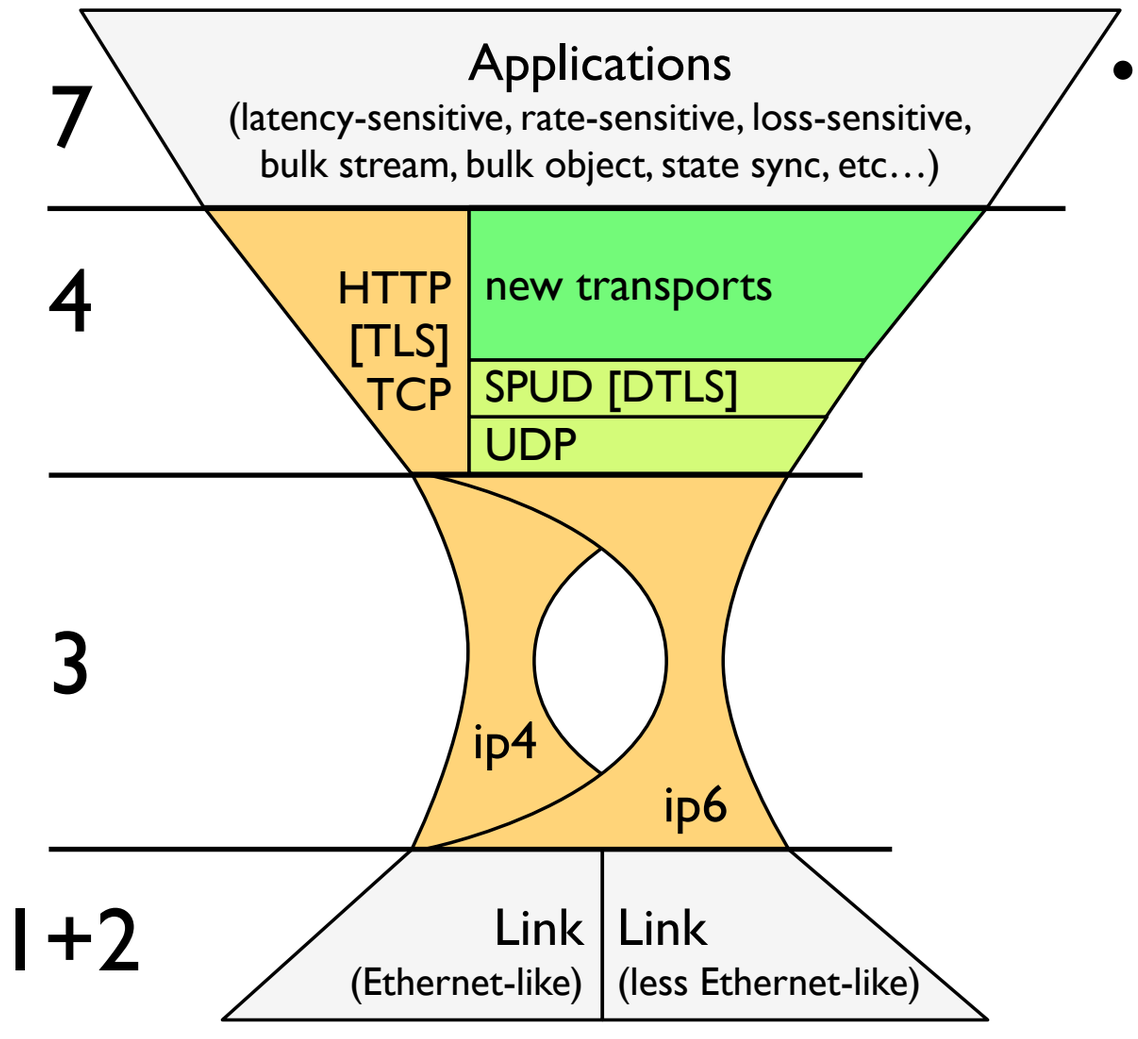
- Engineering decisions about protocols to deploy in the Internet should be based on **relevant data about the environment** they face.
  - Design for common occurrences.
  - Know the risks of uncommon ones.
  - Apply measurement liberally to know the difference. Maybe even at runtime.

Today's talks

- IP stack evolution and path impairment
  - **Can we run the Internet over UDP?**
    Need more data.
- Understanding interdomain topology
  and **BGP dynamics**.
  - Need more data, better tools for data we have.
- Discussion:
  what can measurement do for you,
  and what can you do for measurement?

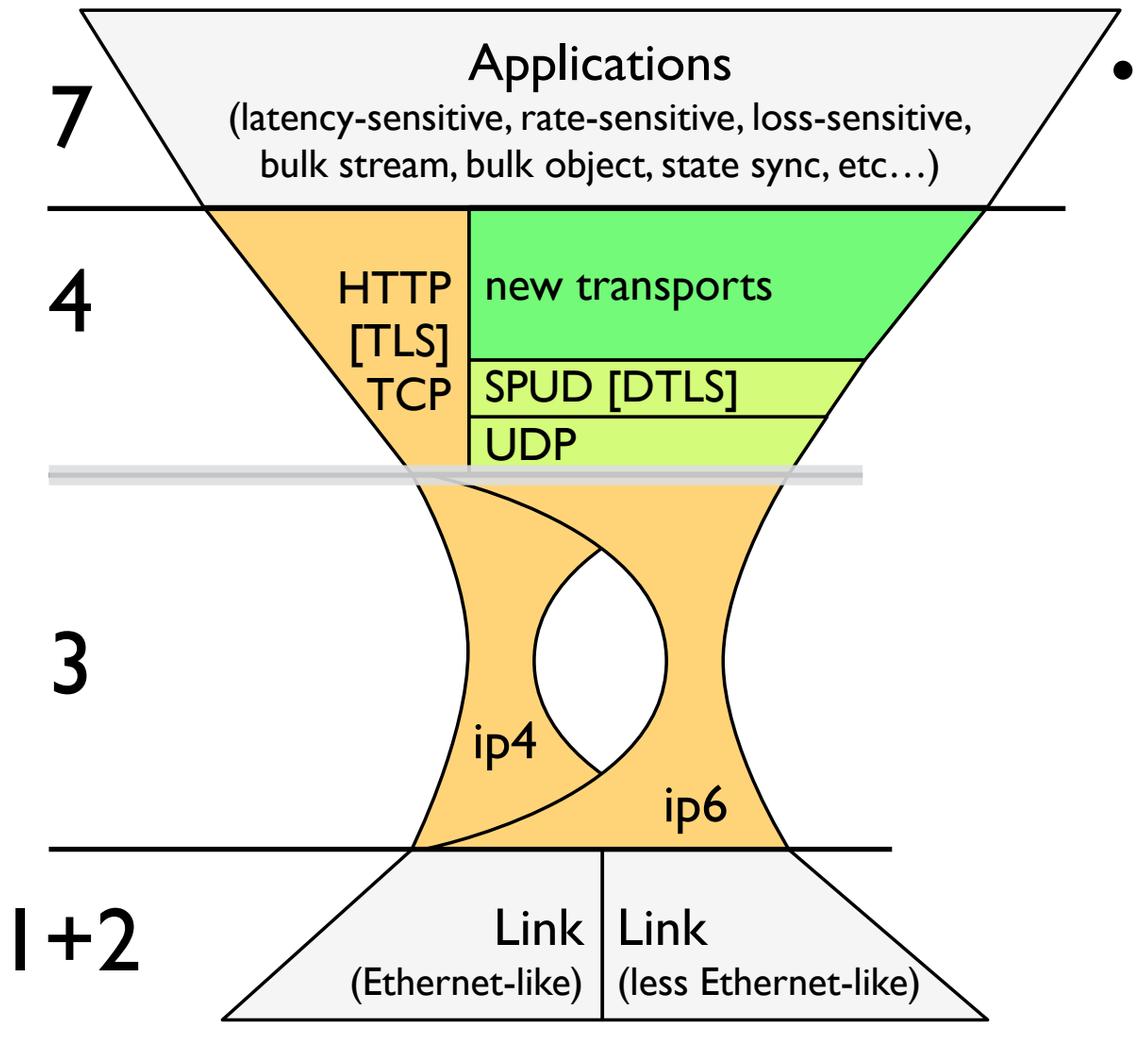
IP Stack Evolution and Path Impairment

Evolving the stack:
explicit relayering and cooperation



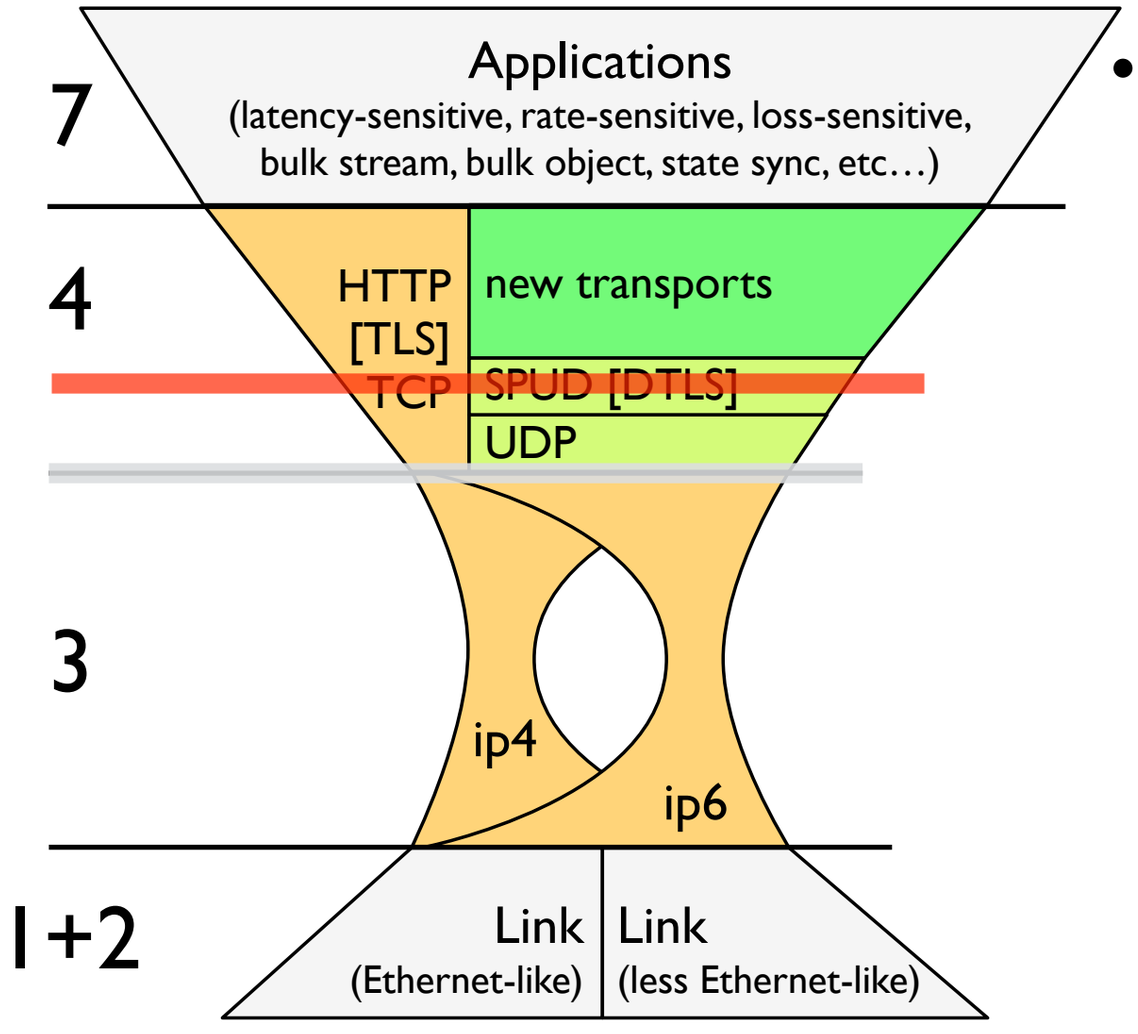
- Rethinking the layer boundary
  - UDP encapsulation (ports for NAT)
  - crypto (reinforce the boundary between endpoint and path visible headers)
  - explicit cooperation (give back transport and application semantics the path actually needs)

Evolving the stack:
explicit relayering and cooperation



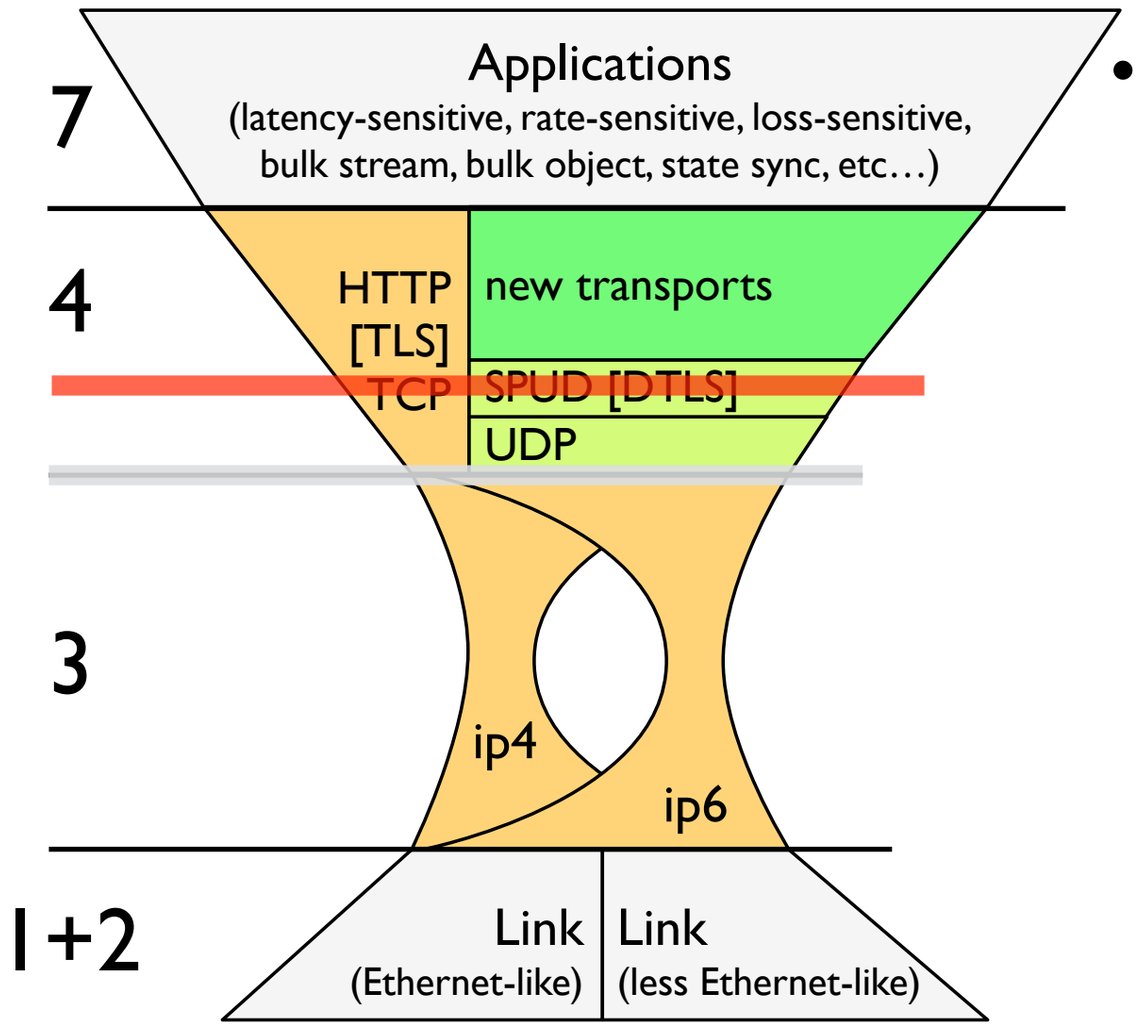
- Rethinking the layer boundary
  - UDP encapsulation (ports for NAT)
  - crypto (reinforce the boundary between endpoint and path visible headers)
  - explicit cooperation (give back transport and application semantics the path actually needs)

Evolving the stack:
explicit relayering and cooperation



- Rethinking 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)

Evolving the stack:
explicit relayering and cooperation

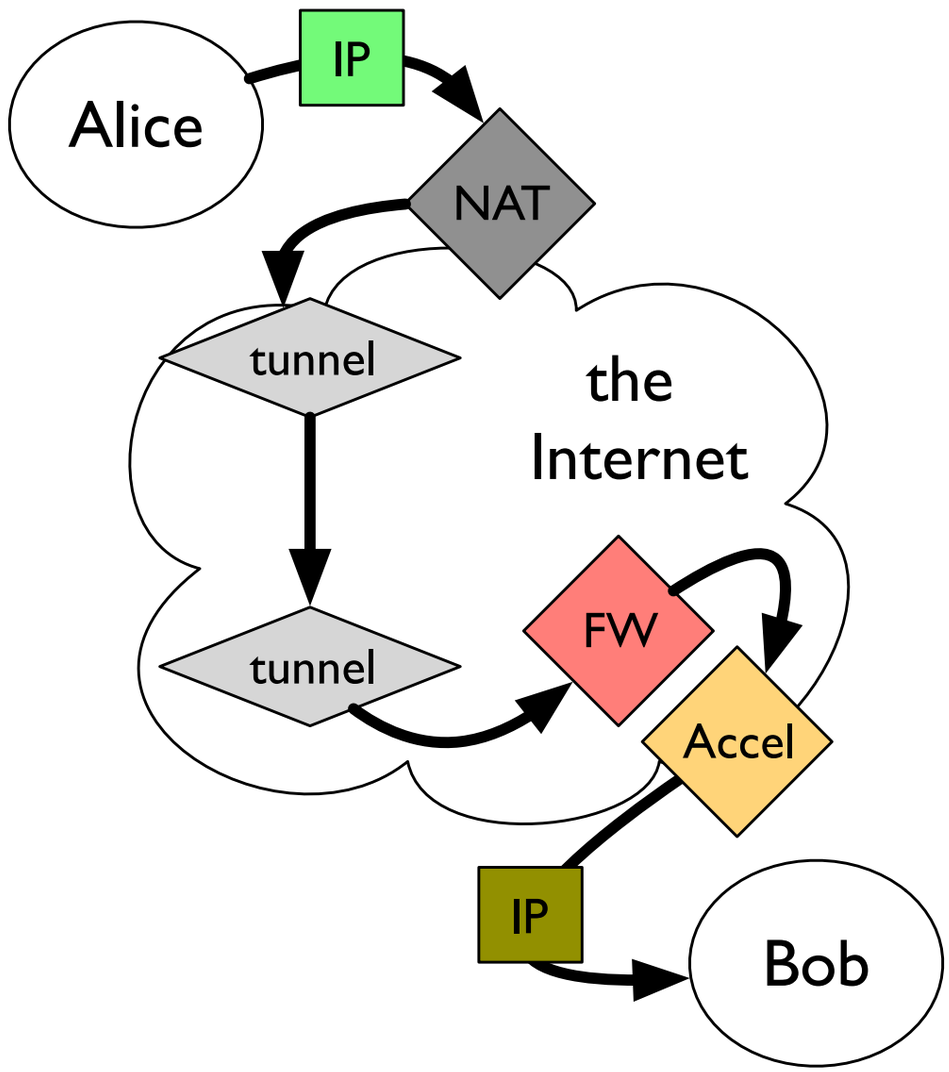


- Rethinking the layer boundary
  - UDP encapsulation (ports for NAT)
  - crypto (reinforce the boundary between endpoint and path visible headers)
  - explicit cooperation (give back transport and application semantics the path actually needs)

We assume that UDP works. Does it?

Measuring path impairment

- Path impairment: the likelihood that traffic with given characteristics will experience problems on a given path.
  - Increased latency, reordering
  - Loss/connectivity failure
  - "Bleaching" or selective disablement of features
- Utopian goals:
  - Given a proposed feature, how and how often does it break?
  - Given a path, what works over it?
- Specific question: can we run the Internet over UDP?



- First step: sharing what we know. **HOPSRG** (hops@ietf.org)

What can go wrong?

Modification	Planetlab	Ark
IP Address	**74.9%**	**79.0%**
ECN IP	**13.7%**	**13.2%**
TCP ISN	**10.7%**	1.8%
TCP MSS	**10.8%**	5.9%
TCP Ex.Opt.	**8.8%**	0.5%
MPCAPABLE	**8.4%**	0.3%
ECN TCP	0.6%	0.6%
TCP SackOK	0.3%	0.0%
TCP TS	0.3%	0.4%
TCP WScale	0.2%	0.2%

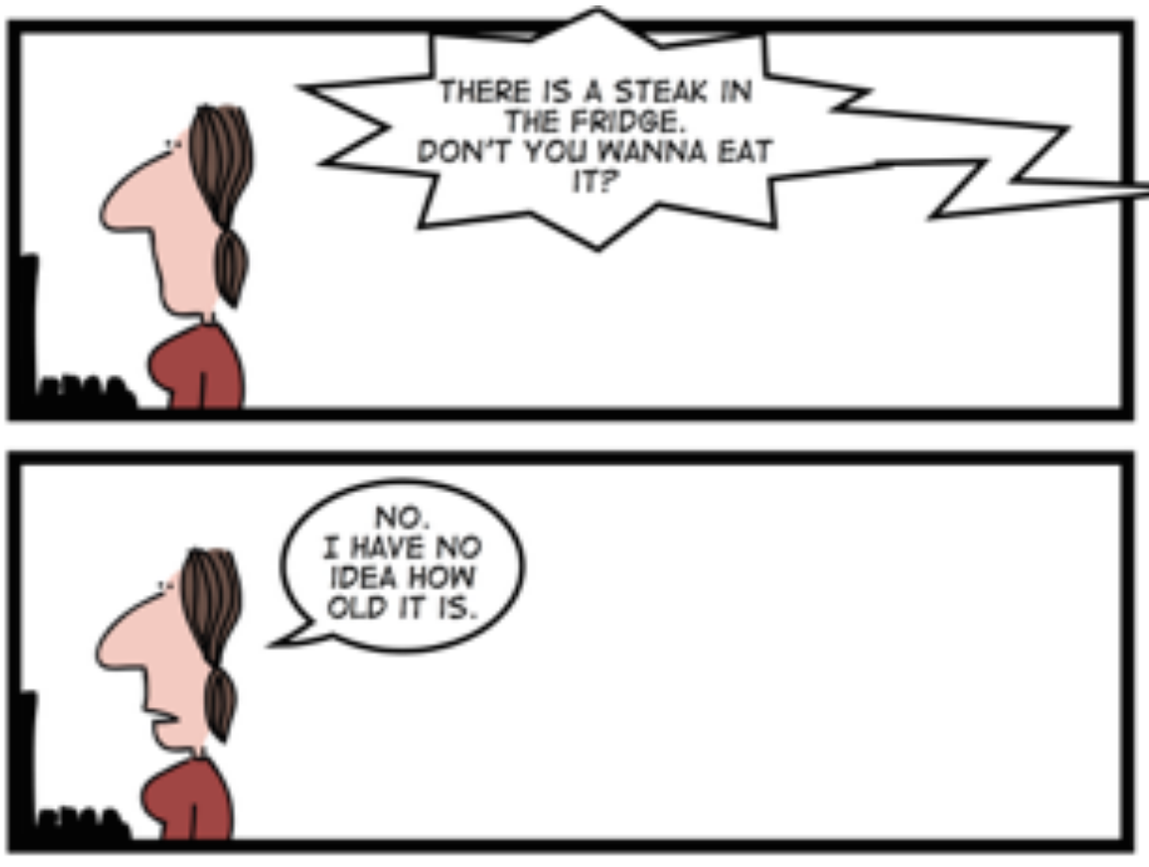
Percentage of paths modifying selected packet feature on two research-oriented testbeds.

- **Best studies look at O(10k) paths[1].**
- The Internet has billions and billions.
- Results highly dependent on vantage point.
- Need more diversity to answer the question.

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

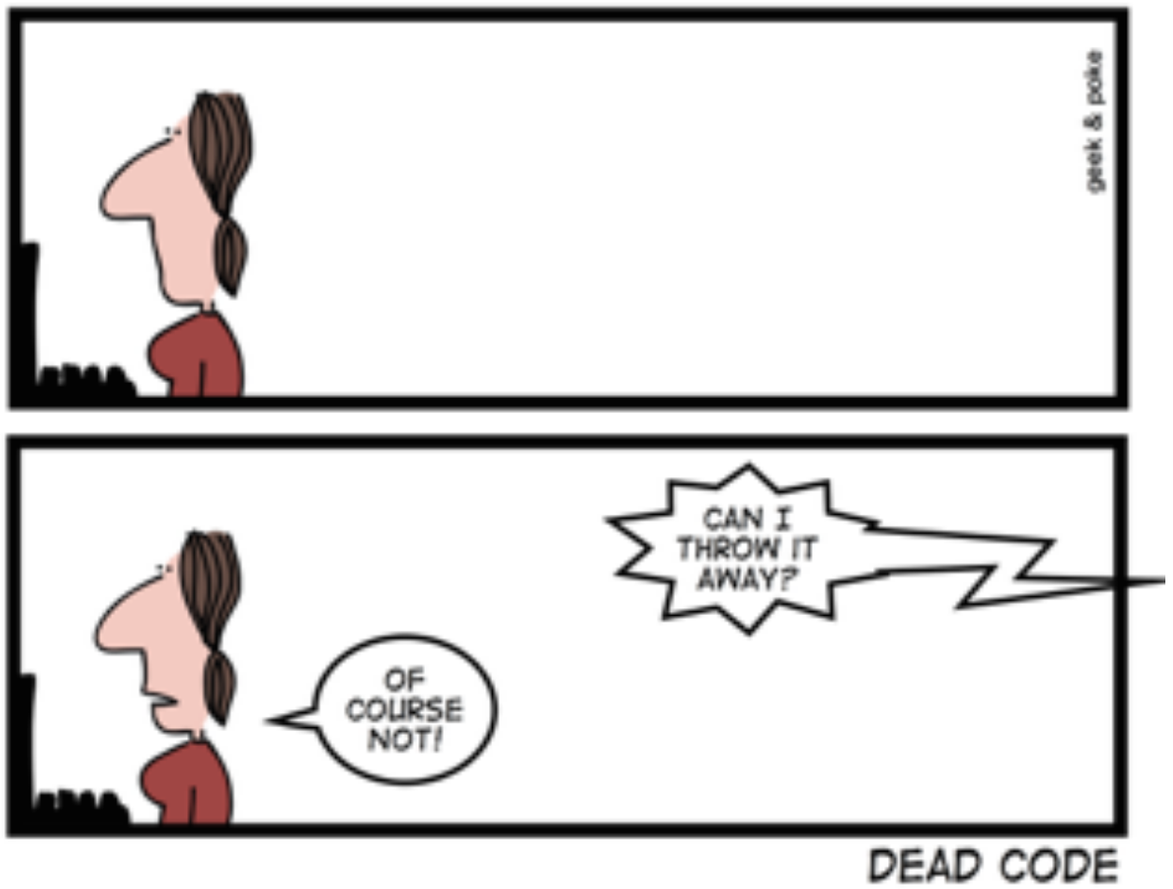
Application to Protocol Engineering

- We want our protocols to work when stuff breaks.
- Engineering tradeoff: **robust code** v. **robustness against the path.**



Application to Protocol Engineering

- Special cases that never happen lead to dead code.
  - NAT?
    Design for it.
  - Custom hack deployed in one network?
    Write a polite email.
- We need information about prevalence to make informed decisions.

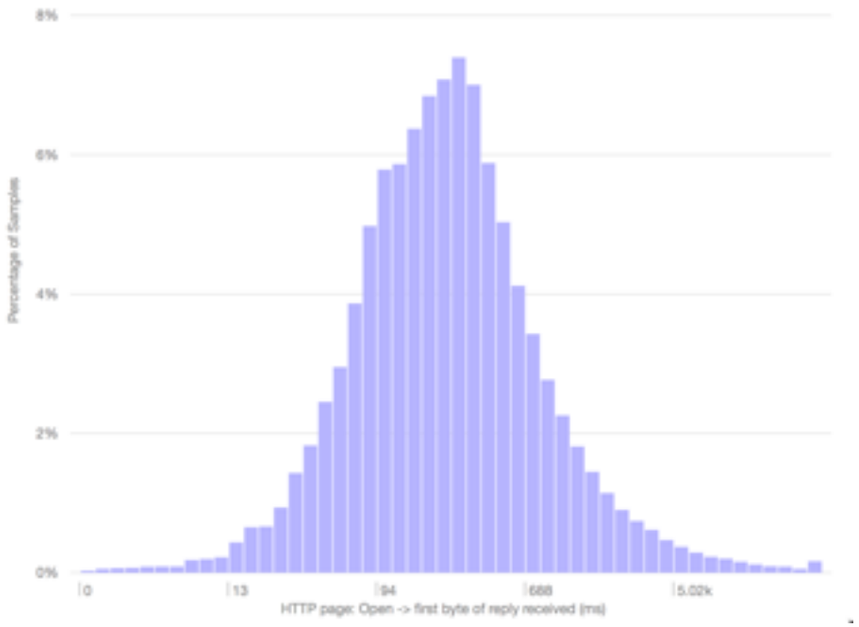


Measuring the Internet is hard

- 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 encryption you see on a given link depends on application mix and the vagaries of CDN policy[2].
  - What is easy to measure not necessarily most relevant.
- Not enough data and too much data *at the same time.*

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

Measuring without measuring



e.g. HTTP first-byte time, telemetry.mozilla.org

- Lots of things that don't look like measurement are.
  - TCP
  - Version negotiation and fallback
  - Platform-level diagnostics
- **Vision**: Let's design protocols with this fact in mind.
  - Make instrumentation accessible.
  - Explicitly measure and react at runtime.

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.
- Goal: combine measurements from different vantage points and data sources for wider and deeper insight.
  - Develop common information models and query sources[3].
  - Common coordination and control protocols[4].

[3] e.g. BGPstream (see next talk)
[4] e.g. mPlane, ict-mplane.eu, draft-trammell-mplane-protocol

Understanding real-world BGP Dynamics

Discussion

Ask what measurement can do for you…

- Questions to ask during protocol design:
  - 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 sources would help that **don’t exist**?
  - What information does the protocol **generate** as a side effect that can lead to better insight? Can implementations use this at runtime?

…and what you can do for measurement

- There are many other insights to be gained from the Internet by measuring it in different ways.
- Integration of diverse measurements leads to better insight.
- Data generated as a side effect of a protocol's operation might be useful in other contexts.