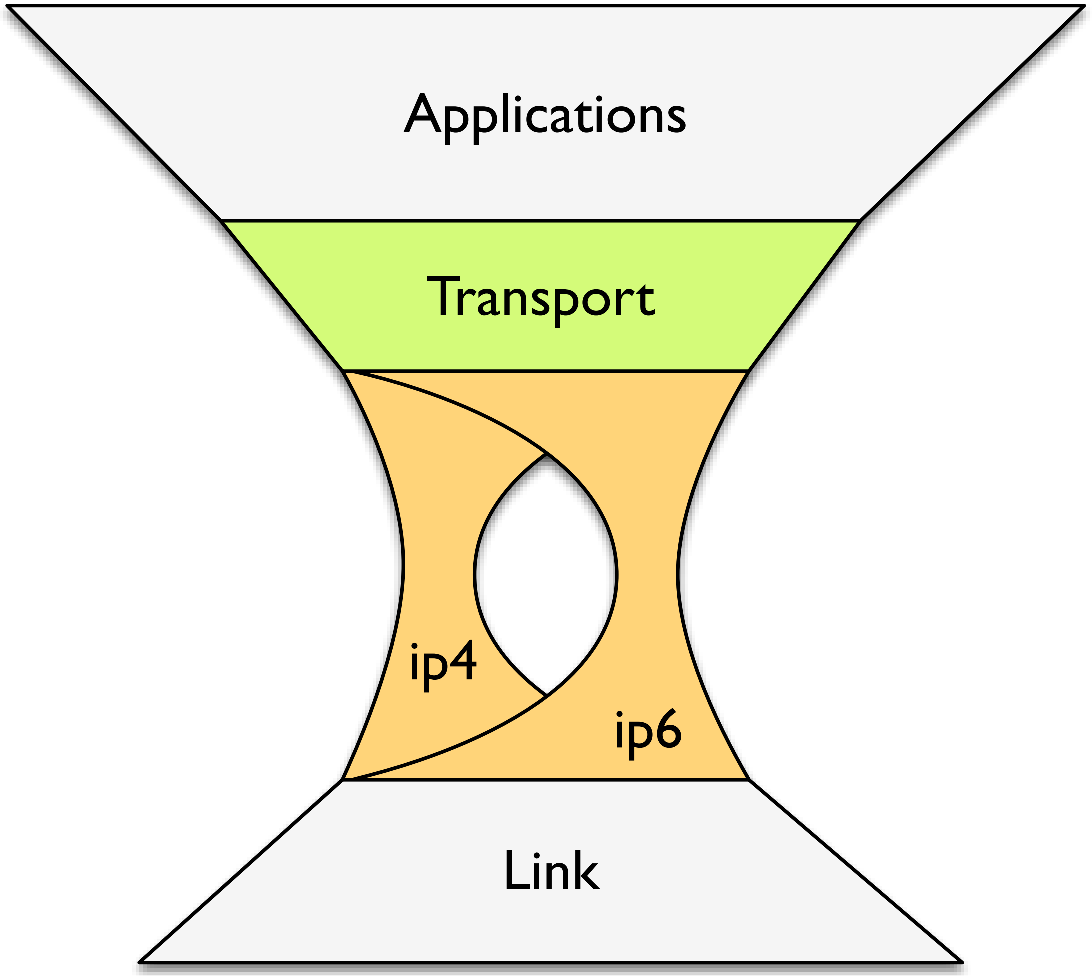
Layer 3.5
or: Things We Need to Admit

Brian Trammell, CSG, ETH Zürich, IAB
Joe Hildebrand, Cisco Systems, IAB

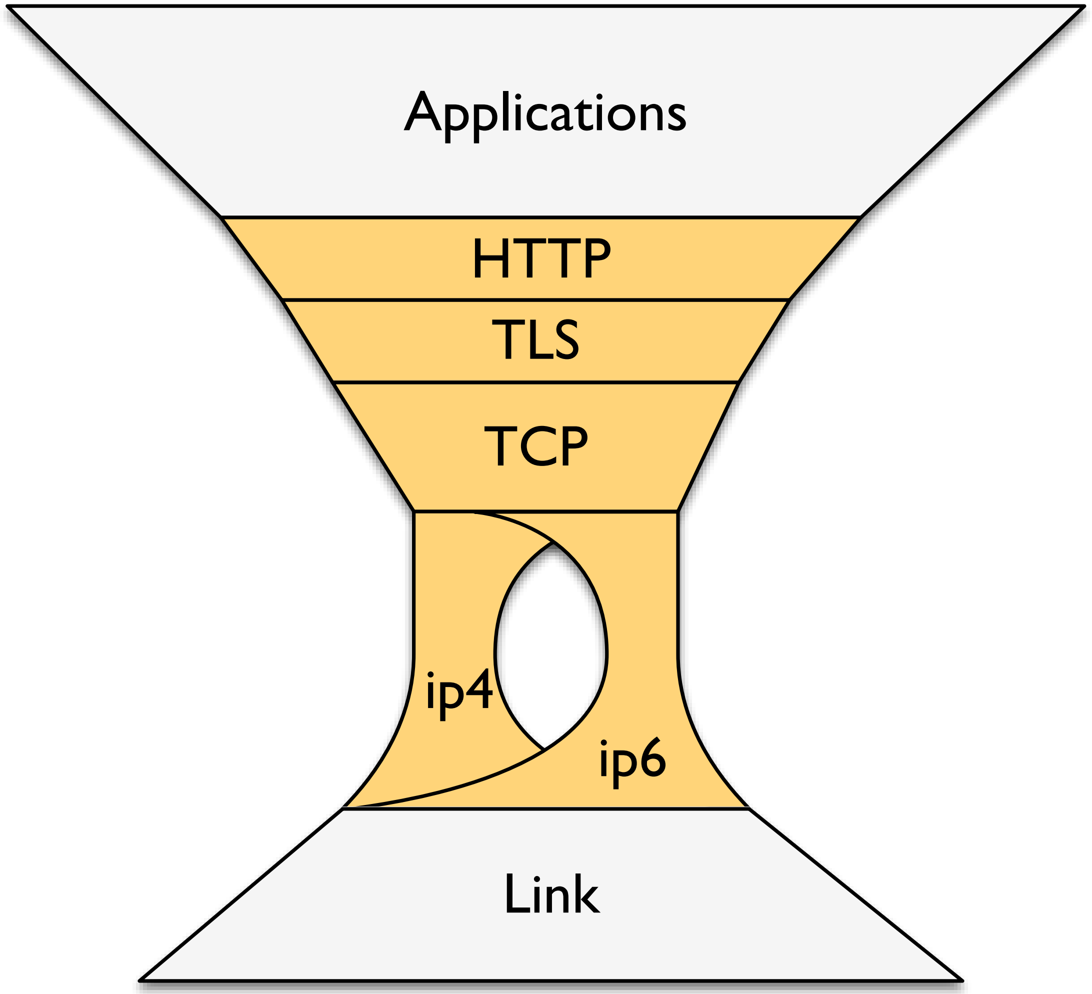
A Taller, Thinner Hourglass

- We’ve evolved IP to have a dual stack waist…
- …but this picture is decreasingly accurate above the network layer.



A Taller, Thinner Hourglass

- HTTP (+TLS) is a universal session layer
  - Driven by endpoints (browsers as front-end) as well as the network (HTTP-/proxy-only connectivity)
- HTTP implies TCP, which is not always what we want.
  - Transport stagnates, or innovation happens beyond the stack.



Narrow Interfaces

- `fd = socket()`: yay, the network is a special kind of file. But…
  - `SOCK_STREAM:` single-streaming with full reliability and head of line blocking…
  - `SOCK_DGRAM:` record-oriented transport with zero reliability and MTU issues…
  - And no other (realistic) choices for transport
- Identifier bound to location: roaming is “difficult”.
- Security bolted on, and OpenSSL’s API is “unique”.
- Kernel/userspace boundary in the wrong place: proven inability to deploy new transports over IP.

Reasonable Brokenness

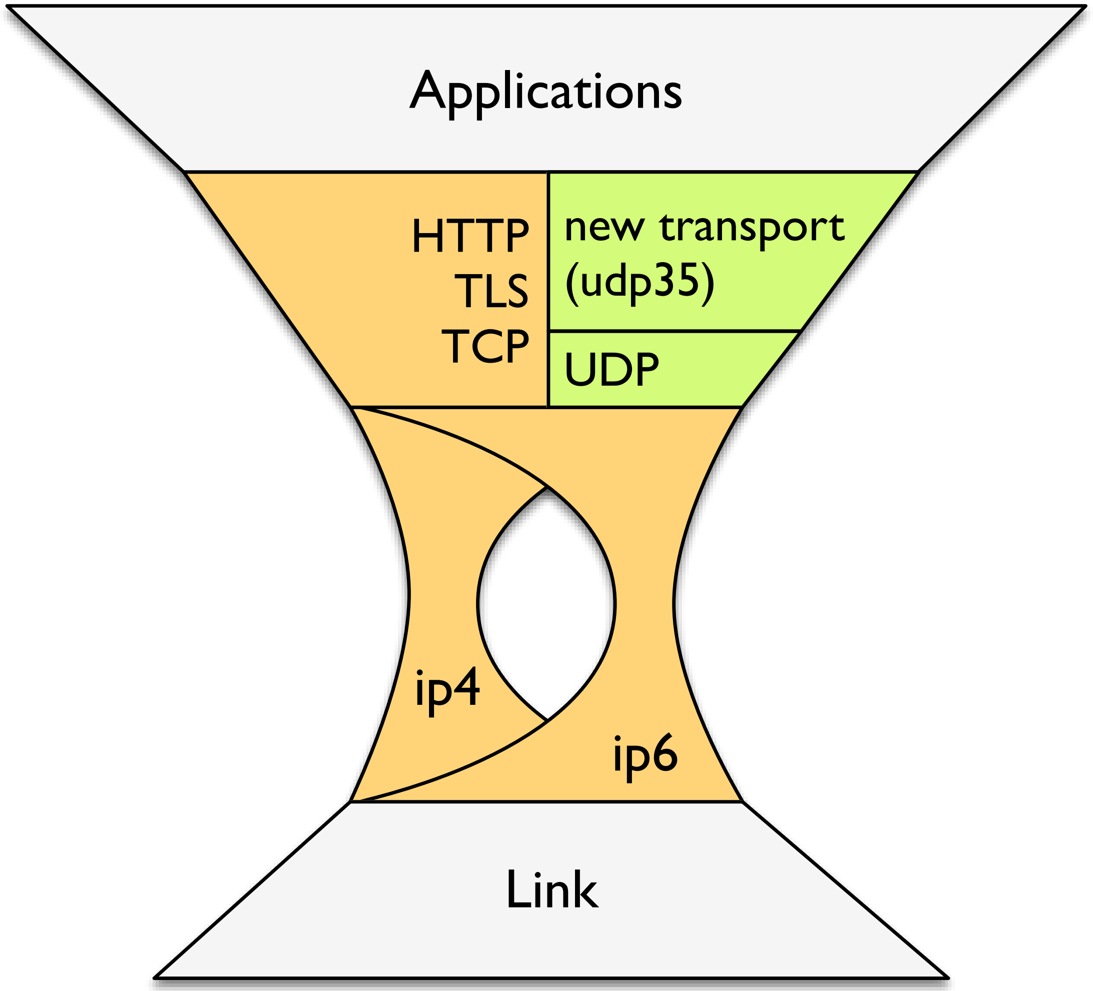
- End-to-end wasn't broken out of spite:
- Developers want to build cool new apps
- Platform providers want to make this easy to do
- Operators want to connect people and stuff
- Vendors want to sell boxes that make this possible
- Everyone wants it to be deployable
- *We need a way to make everyone happy without breaking end-to-end.*

Things we’ve tried

- SCTP (RFC 2960, 2000 / RFC 4960, 2007)
  - Configurable layered transport
  - + has much of the flexibility we want
  - - late integration of security (cf. RFC 6280)
  - - knobs on the API don’t match app requirements
  - - - undeployable in a middlebox world (cf. RFC 6951)
- Protocol-negotiation protocols with worst-case MTI (cf. BEEP)
  - Risk: lots of machinery that lets us find out that all that works is `SOCK_STREAM`.

Meet The New Wire, Same As The Old Wire

- UDP gives us a defense against middleboxes, provides port multiplexing, and works from userspace.
  - Application developers already exploiting this to build new transports (QUIC)
- Building recommendations / “mix-ins” for transport services atop UDP will make this work better.
  - Congestion control in particular is hard to do well.
- Allow evolution beyond and coexistence with “Internet over HTTP”.



Dimensions of Transport

	SOCK_STREAM	SOCK_DGRAM	SOCK_SEQPACKET
Message atomicity		✓	✓
Stream fragmentation	✓	(IP layer)	✓
Sequence preservation	✓		(optional)
Head-of-line blocking avoidance		✓	(streams)
Sub-channels			(streams)
Full reliability	✓		(optional)
Latency-limited reliability		(app layer)	(optional)
Loss-sensitive congestion control	✓		✓
Delay-sensitive congestion control	(kernel hackers only)		(kernel hackers only)
Endpoint address agility	(MPTCP)		(optional)
Privacy and integrity	(TLS)	(DTLS)	(DTLS…)
Path-state propagation (NAT/FW)	(initiator-out)		

What can the IETF/IAB do?

- Possible IAB documents:
  - Guidelines for userspace transport over UDP
  - Architectural considerations for each transport service dimension (e.g. “ordering”, “reliability”, “object atomicity”, “confidentiality”, “latency sensitivity”, etc., etc.)
- Mailing list: udp35@ietf.org
  - and a side meeting Saturday evening in Toronto
- Organization of a BoF for IETF work on mix-ins, CC recommendations, …?