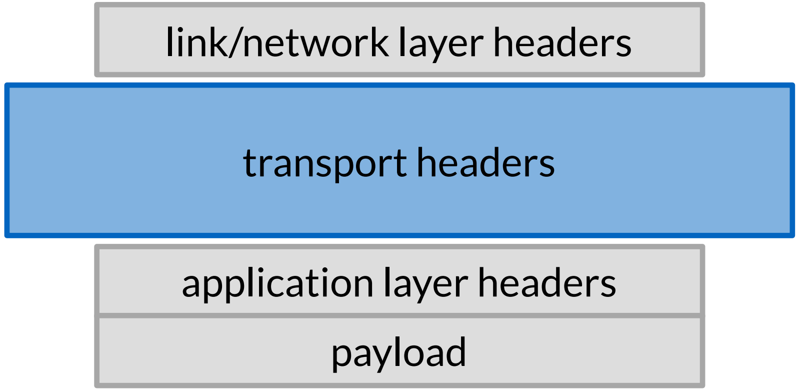
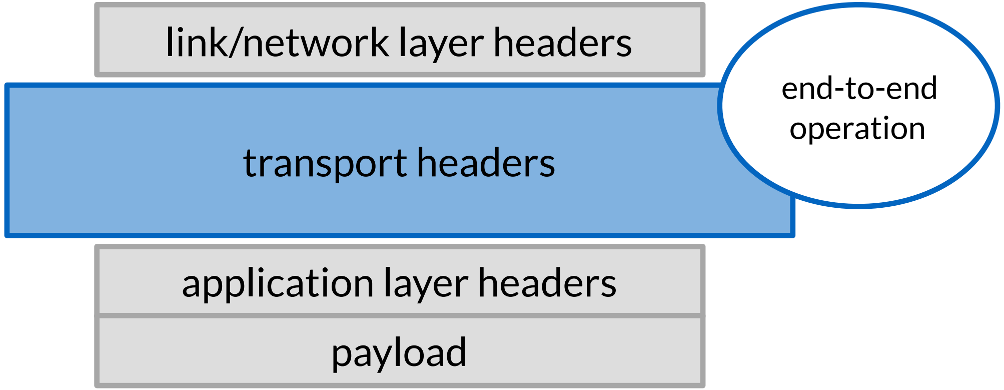
Wire Images, Path Signals, And the (Inter)network ahead

Brian Trammell
Ted Hardie
(And the Stack Evolution Program)

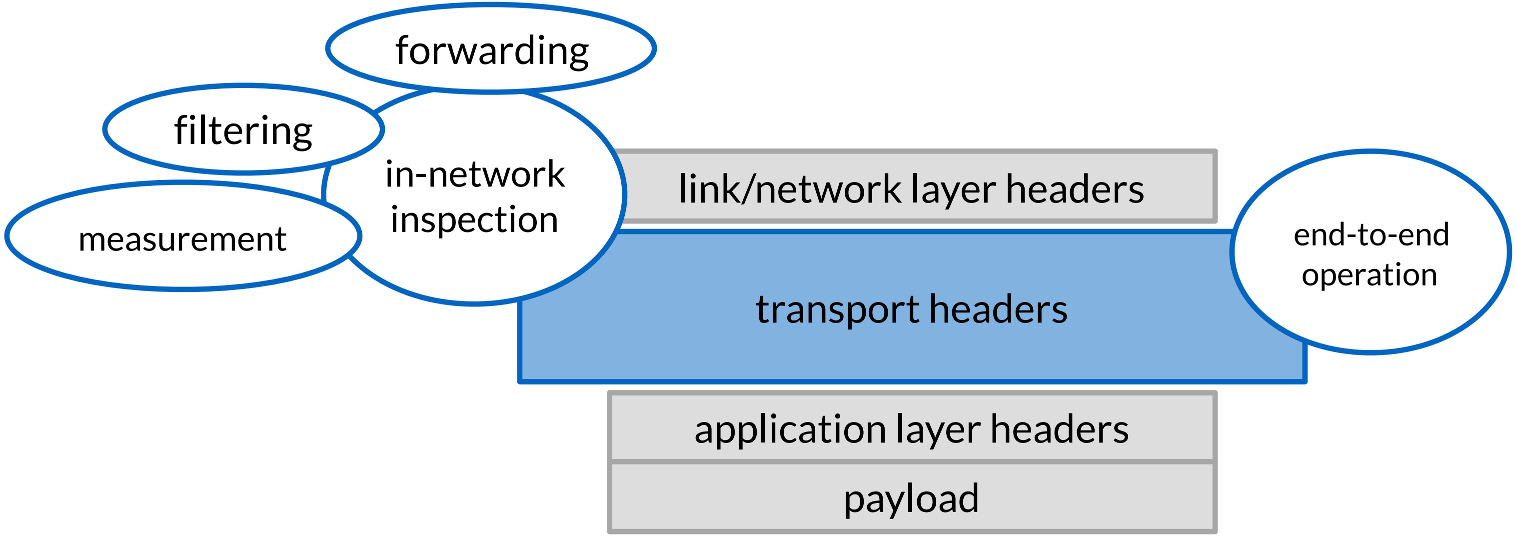
Transport protocol design:
1990s style



Transport protocol design:
1990s style

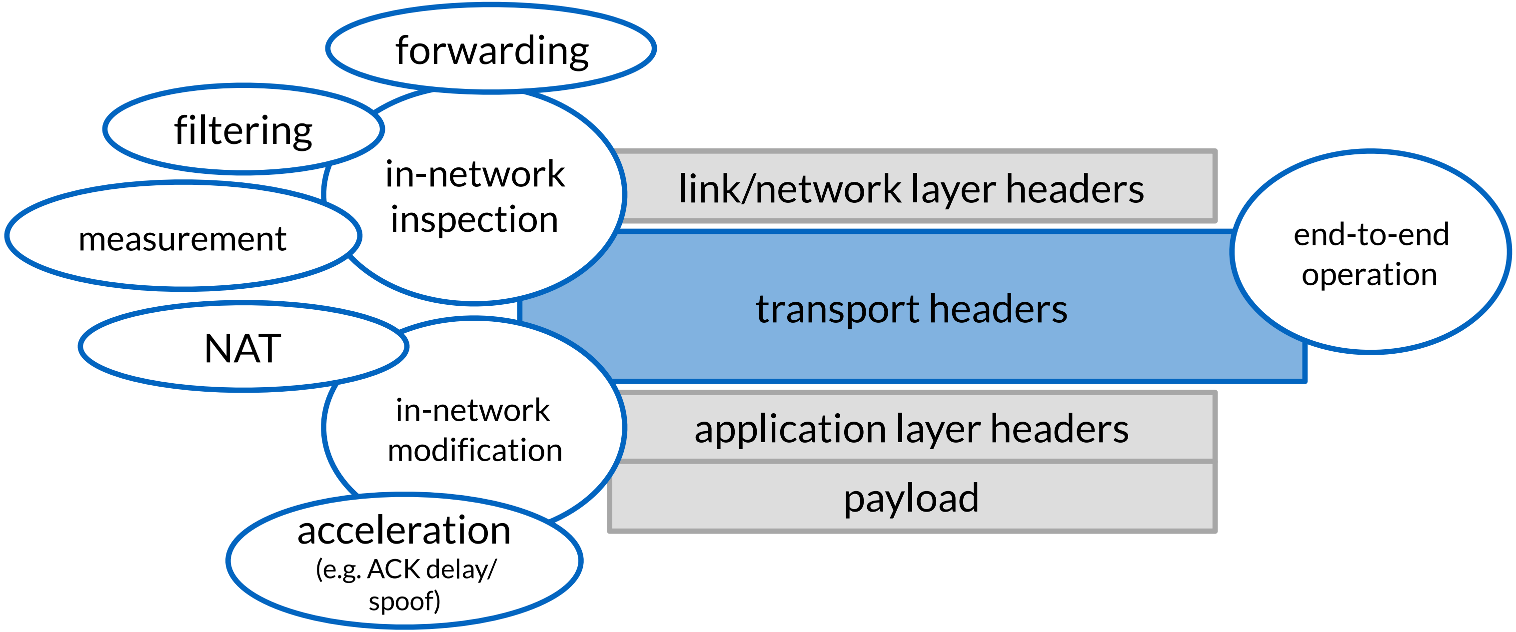


Transport protocol design:
1990s style

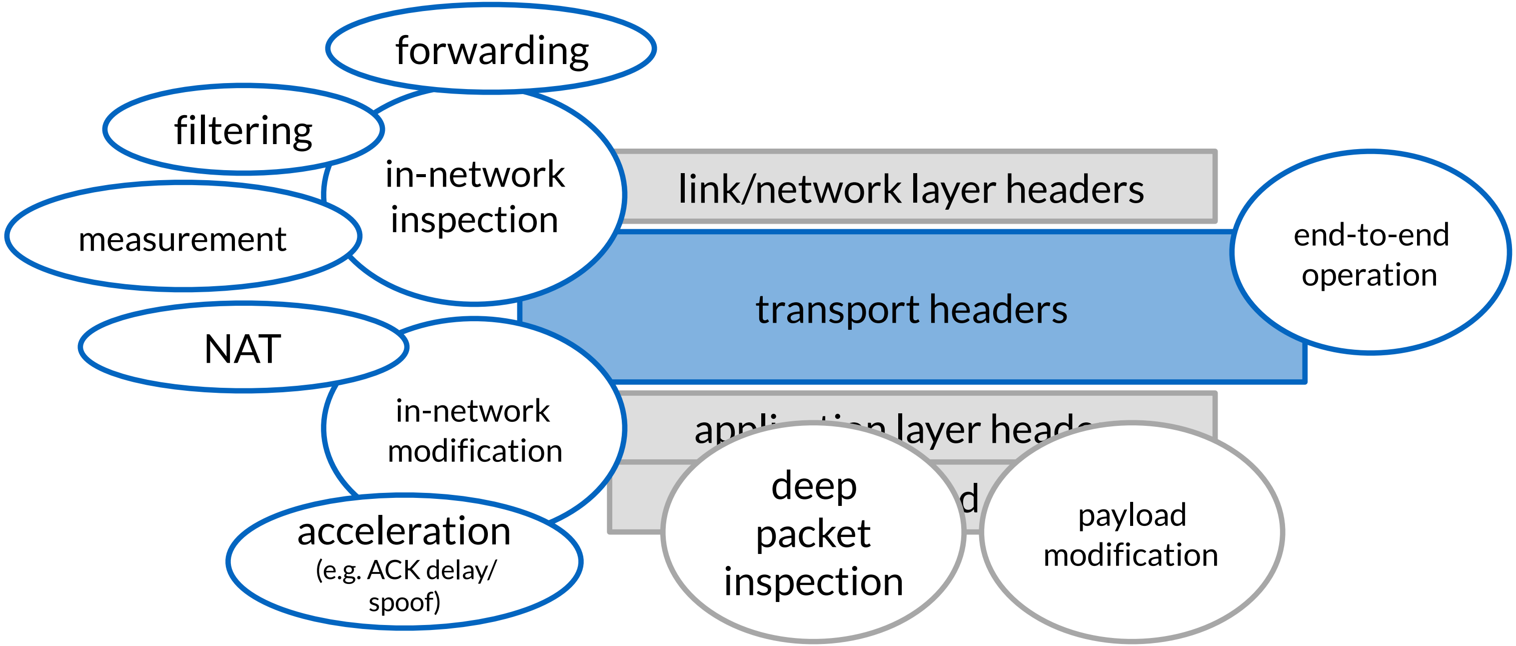


Transport protocol design:

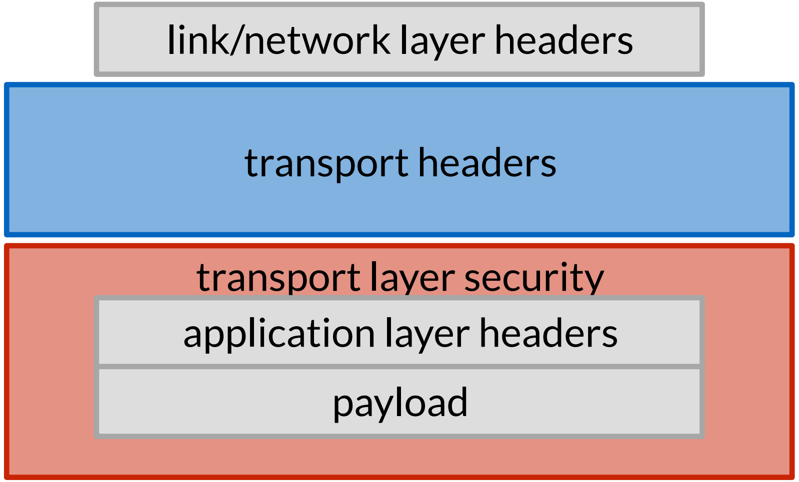
1990s style



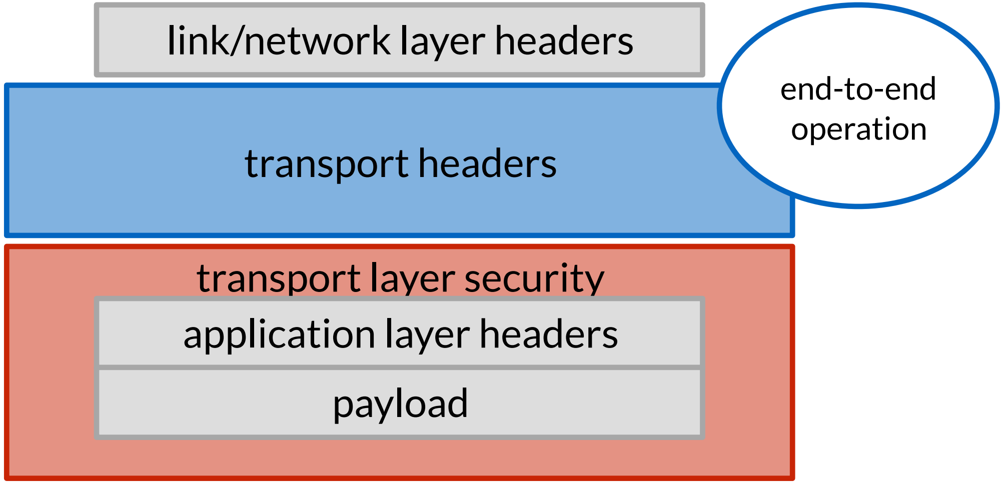
Transport protocol design:
1990s style



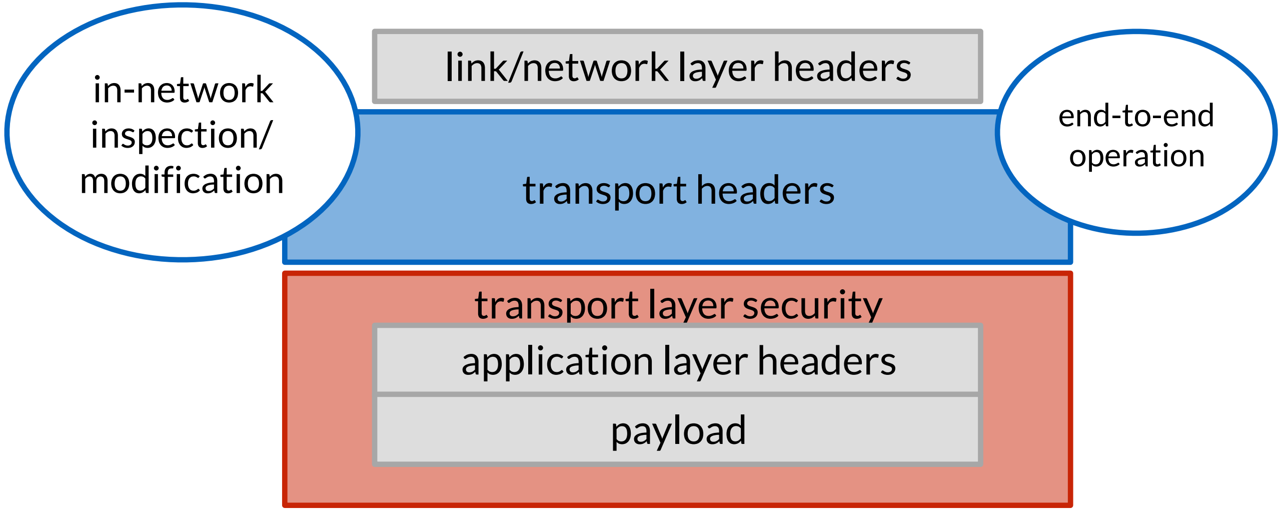
Transport protocol design:
now with security!



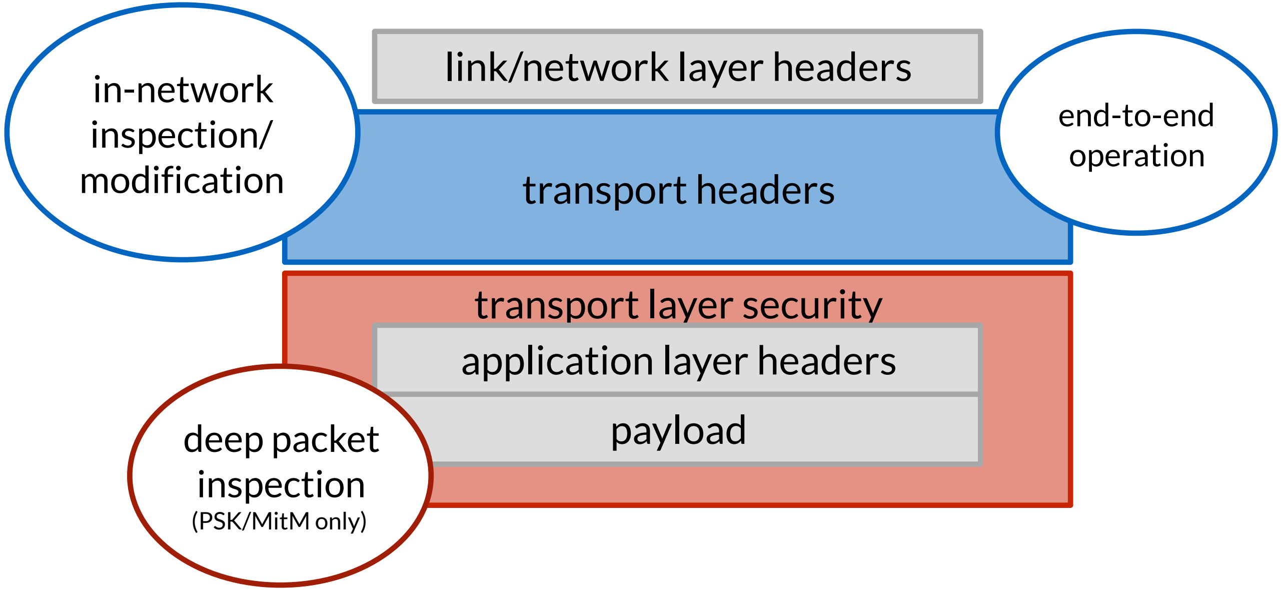
Transport protocol design:
now with security!



Transport protocol design:
now with security!

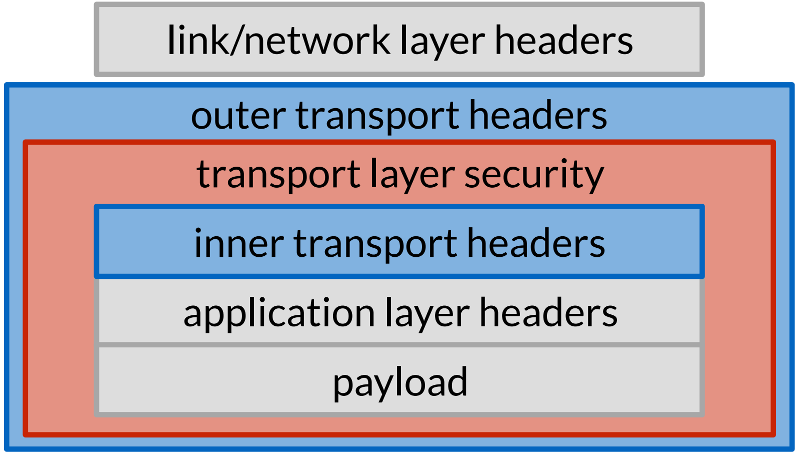


Transport protocol design:
now with security!



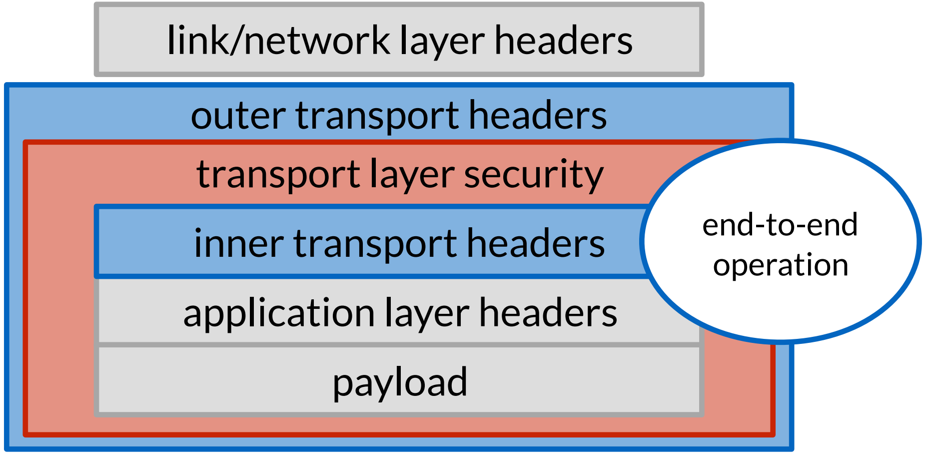
Encrypted transport protocol design:

introducing the wire image



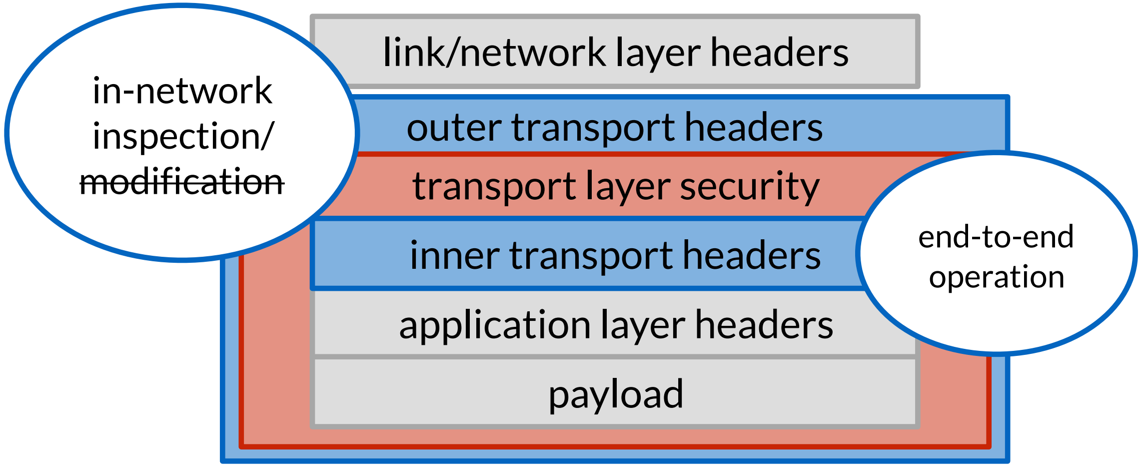
Encrypted transport protocol design:

introducing the wire image



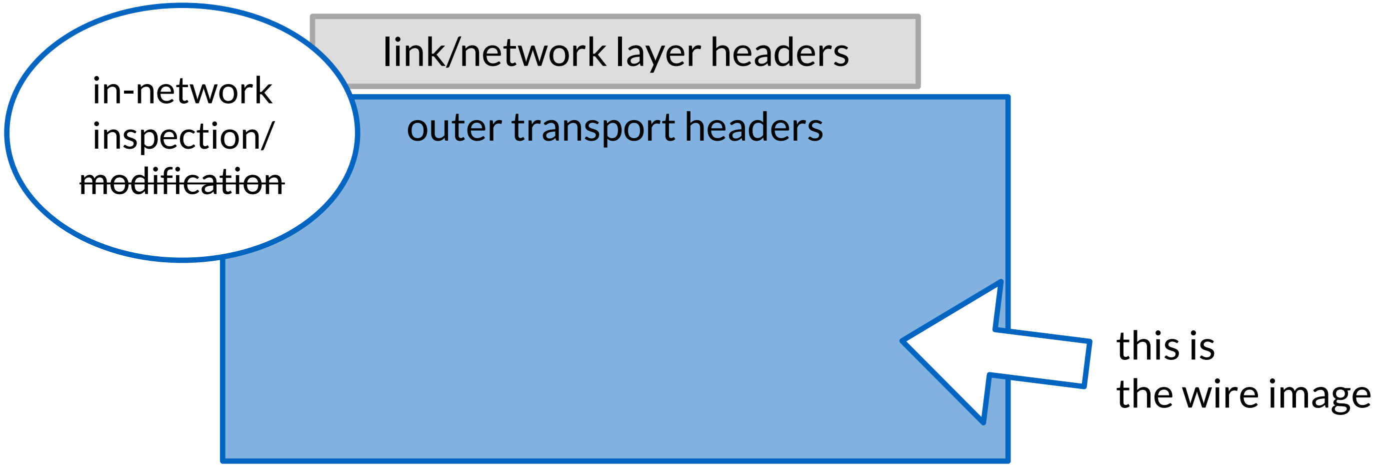
Encrypted transport protocol design:

introducing the wire image



Encrypted transport protocol design:

introducing the wire image



What’s in the wire image?

Information in unencrypted bits in the protocol headers.
(this is the obvious part)

Length and entropy of all bits in the packet.
(provides an upper bound on information content, even for the encrypted bits)

Timing of packet observation (transmission, arrival)
(information about the sender’s behavior)

Why does this matter?

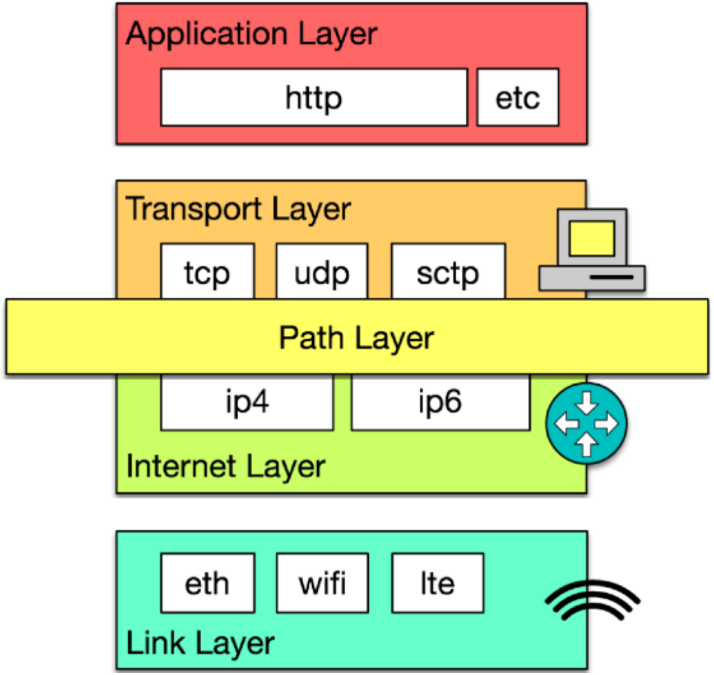
The advent of encrypted transport protocols means that a protocol’s end-to-end operation is separate from its appearance on the wire, and how intermediate devices interact with it.

This is new.

What are path signals?

When transports used cleartext metadata, on-path devices read it and used it to create state, manage resources, and infer permissions.

That is, NATs, Firewalls, and their virtual cousins consumed metadata as if it was intended to signal to them.

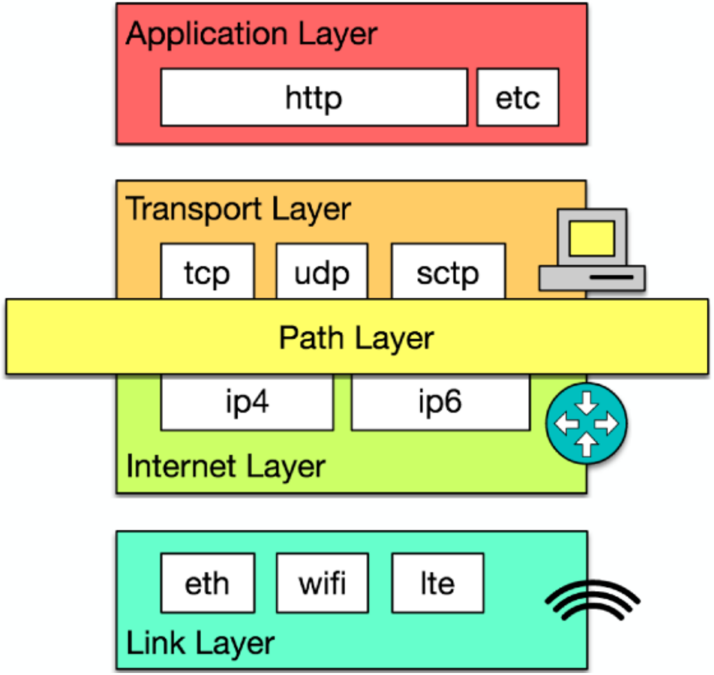


Explicit path signals

When transports use encryption for metadata like packet numbers, these inferences fail.

NATs, Firewalls, and their kin can fall back to default parameters.

Alternatively explicit path signals send data **you intend for the path to consume.**



Where does the signal go?

You could use Internet layer facilities to send these signals.

You could add these signals onto each transport.

You could do nothing.

This is TSVAREA, right?

You could add these signals onto each transport.

The Latency Spin Bit

QUIC experiment: the bit is set by the client and echoed by the server; the client changes the bit once per RTT. Integrity protected by each side.

Exposes the RTT to on-path observers without exposing session state.

Every bit needs to be designated and considered

There’s no default for determining what signals to send; it needs to be determined per transport.

And it needs to be optional; if a client or server don’t want to send that signal, it can’t be needed for session state.

Further Reading

Two IAB drafts on this topic:

draft-iab-path-signals

draft-trammell-wire-image