Wire Images and Path Signals

Brian Trammell, ETH Zürich / MAMI

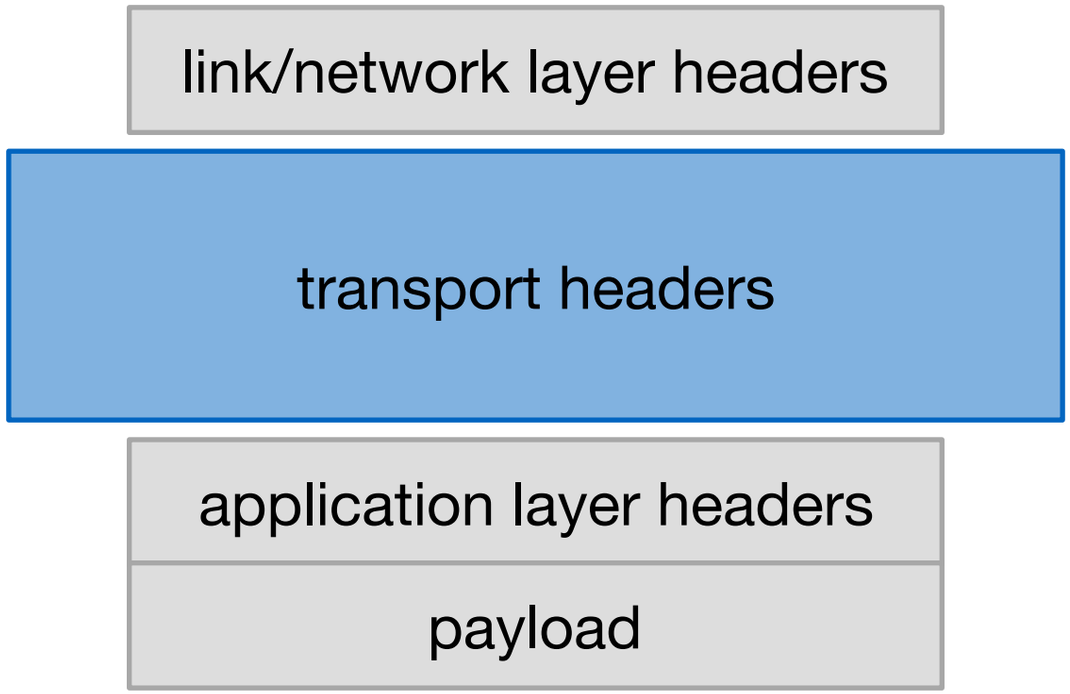
MAMI Management and Measurement Summit (M3S)
London, 16 March 2017

measurement | architecture | experimentation

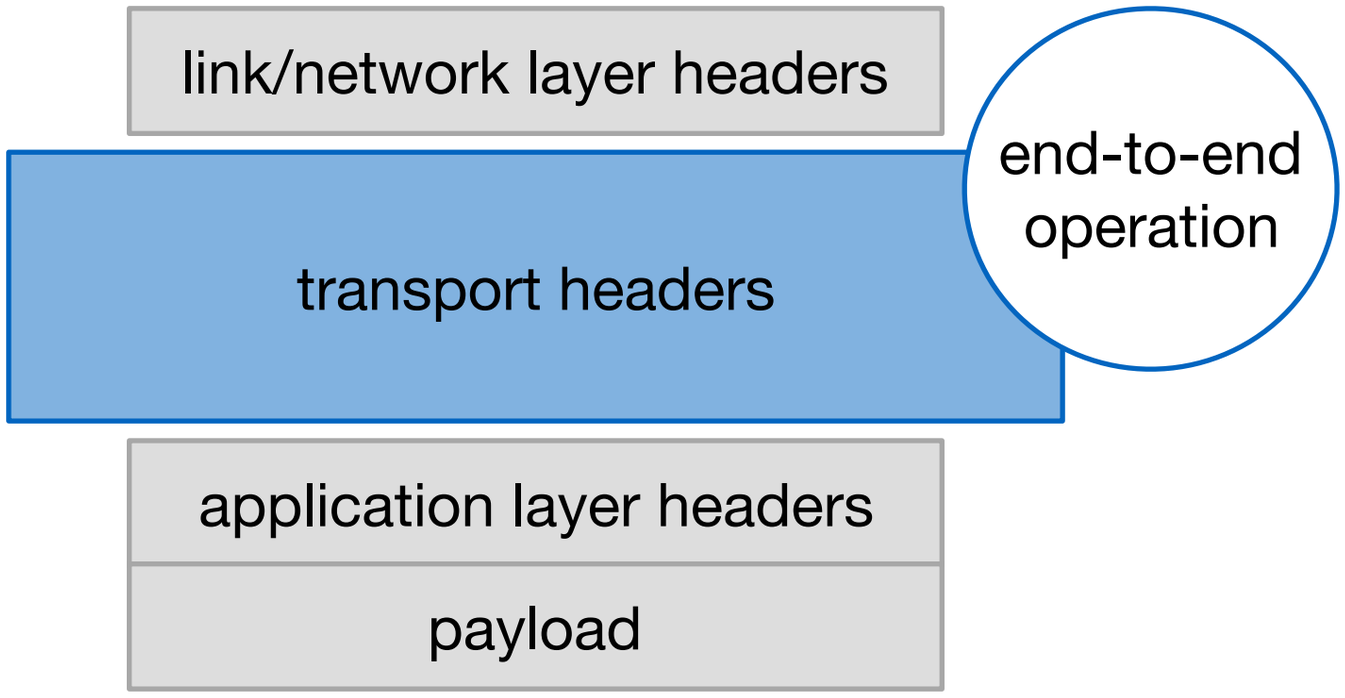
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421.The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information..

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.

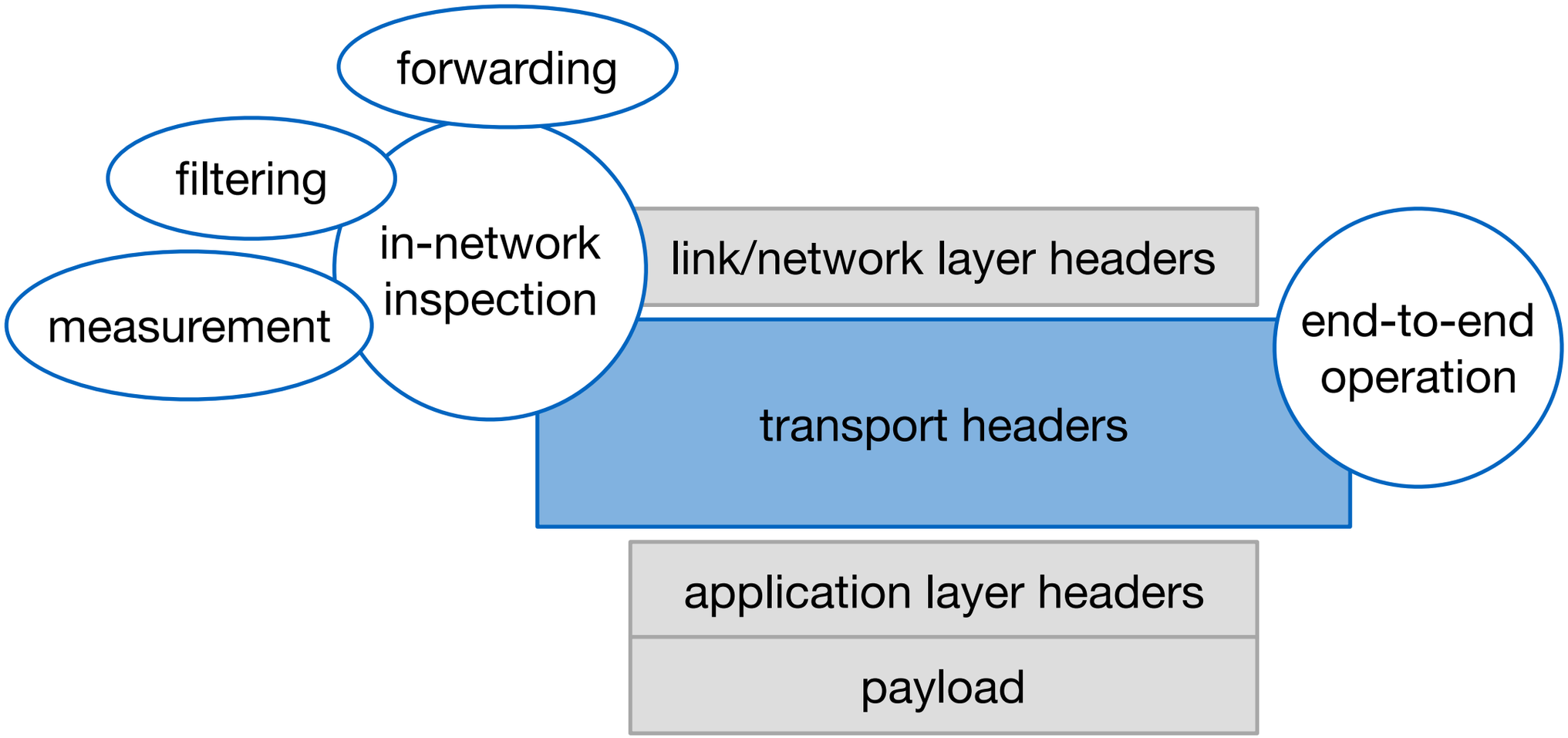
Transport protocol design, 1990s style



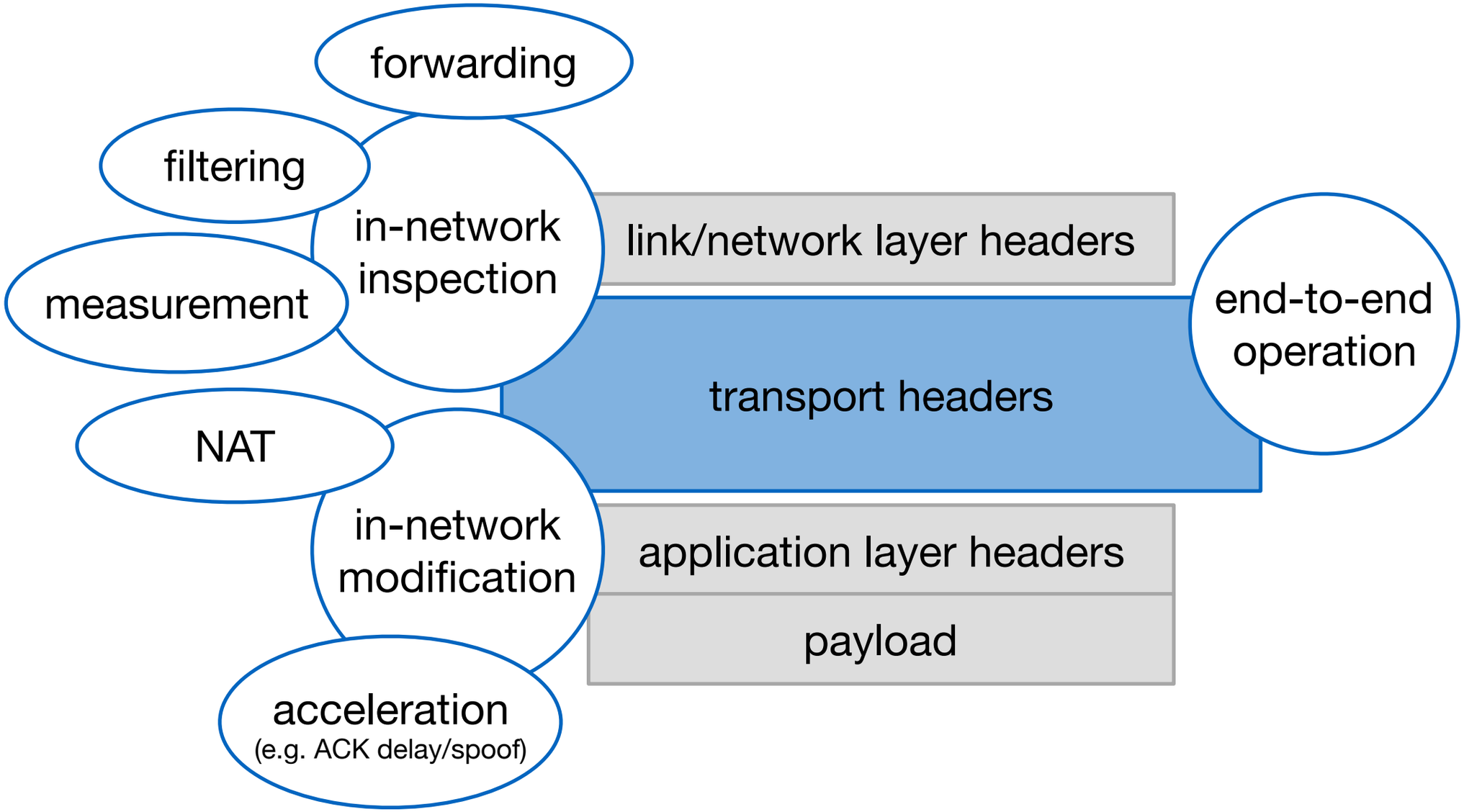
Transport protocol design, 1990s style



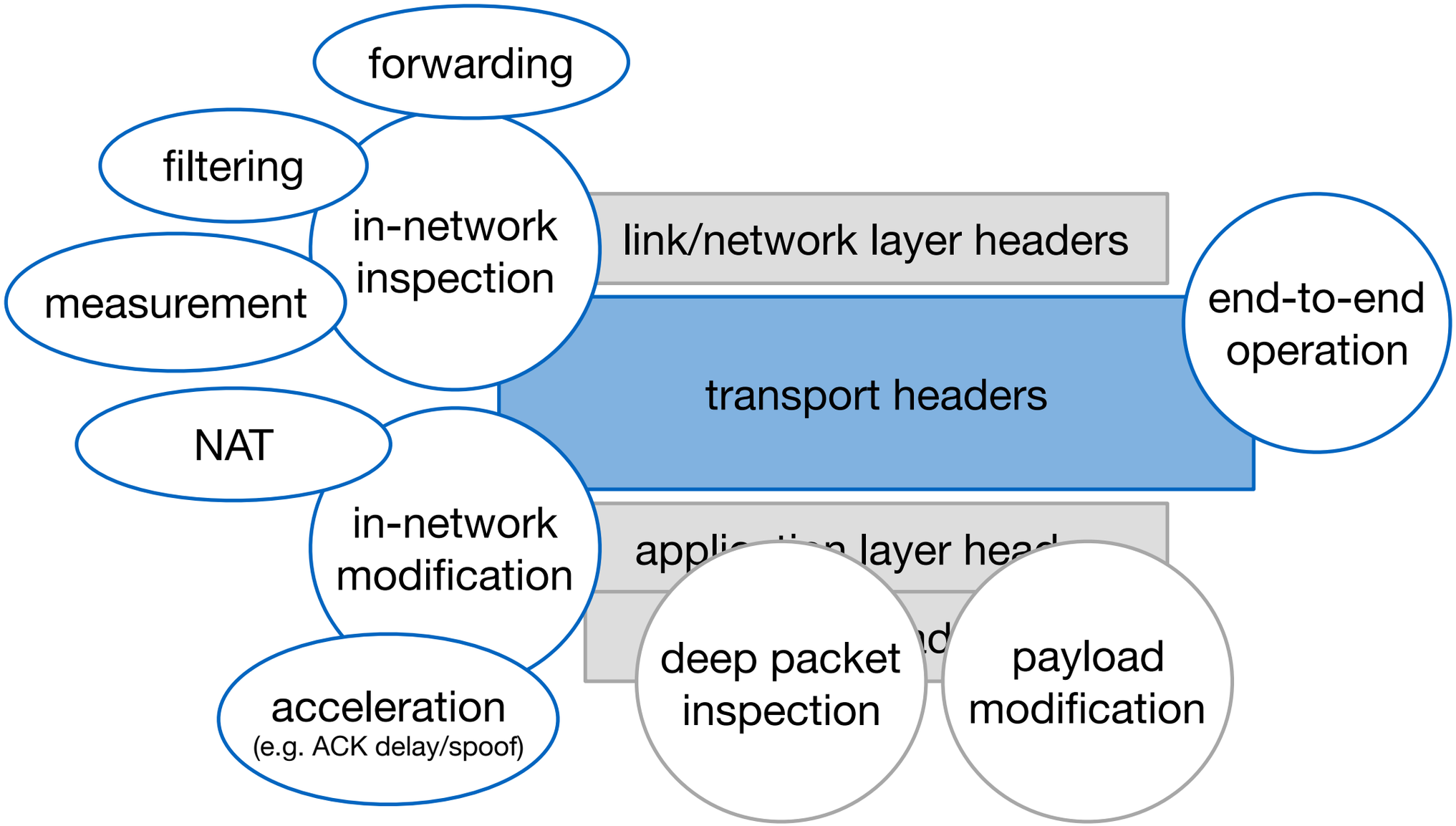
Transport protocol design, 1990s style



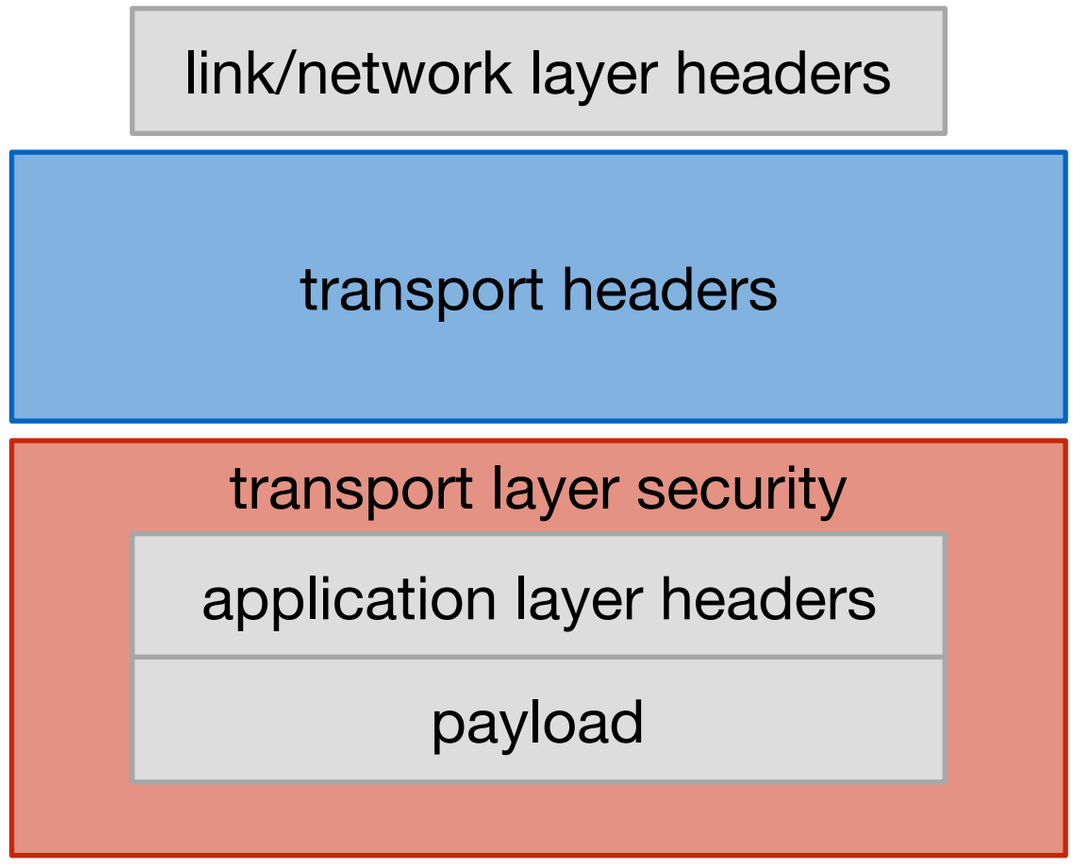
Transport protocol design, 1990s style



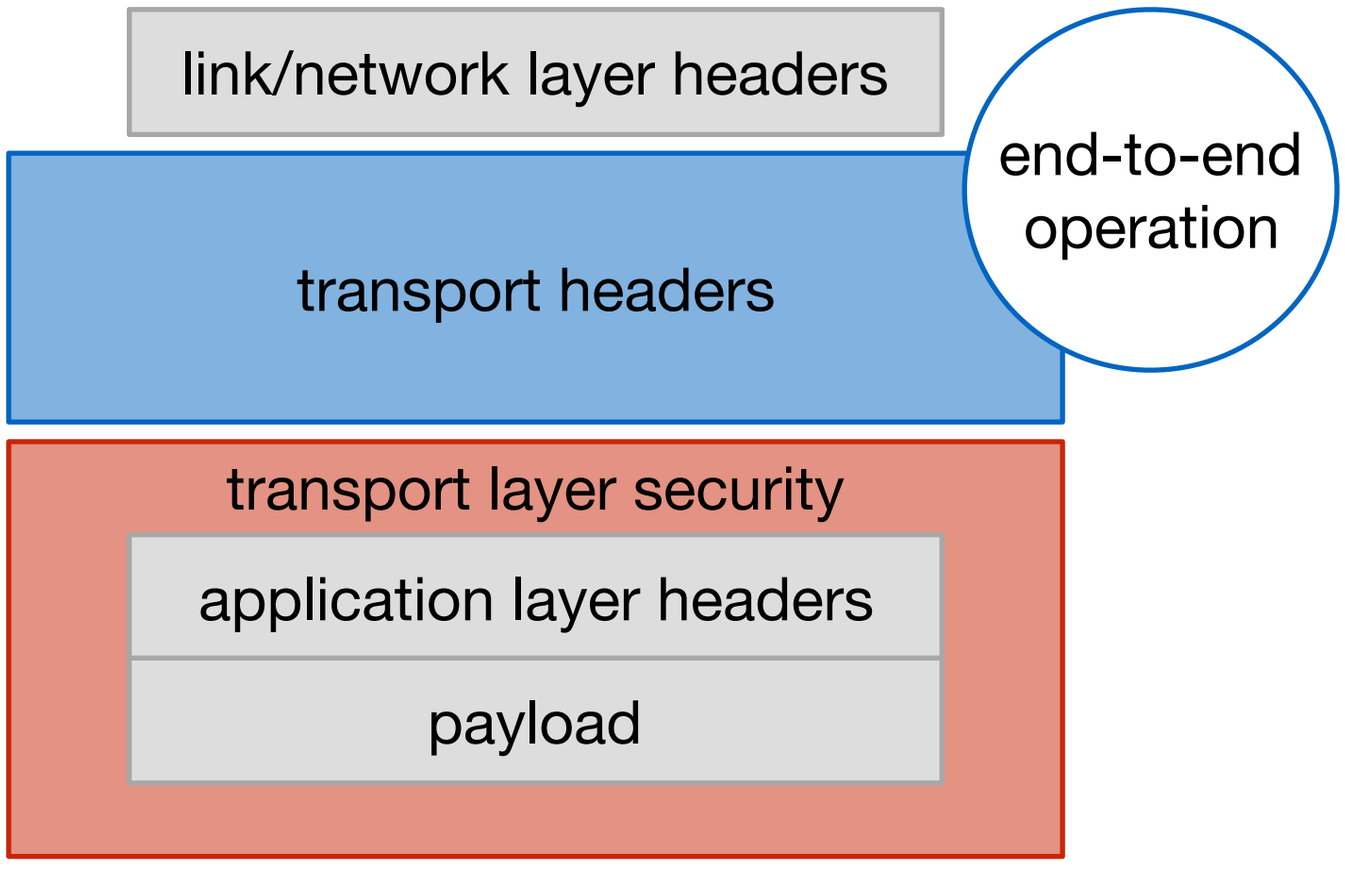
Transport protocol design, 1990s style



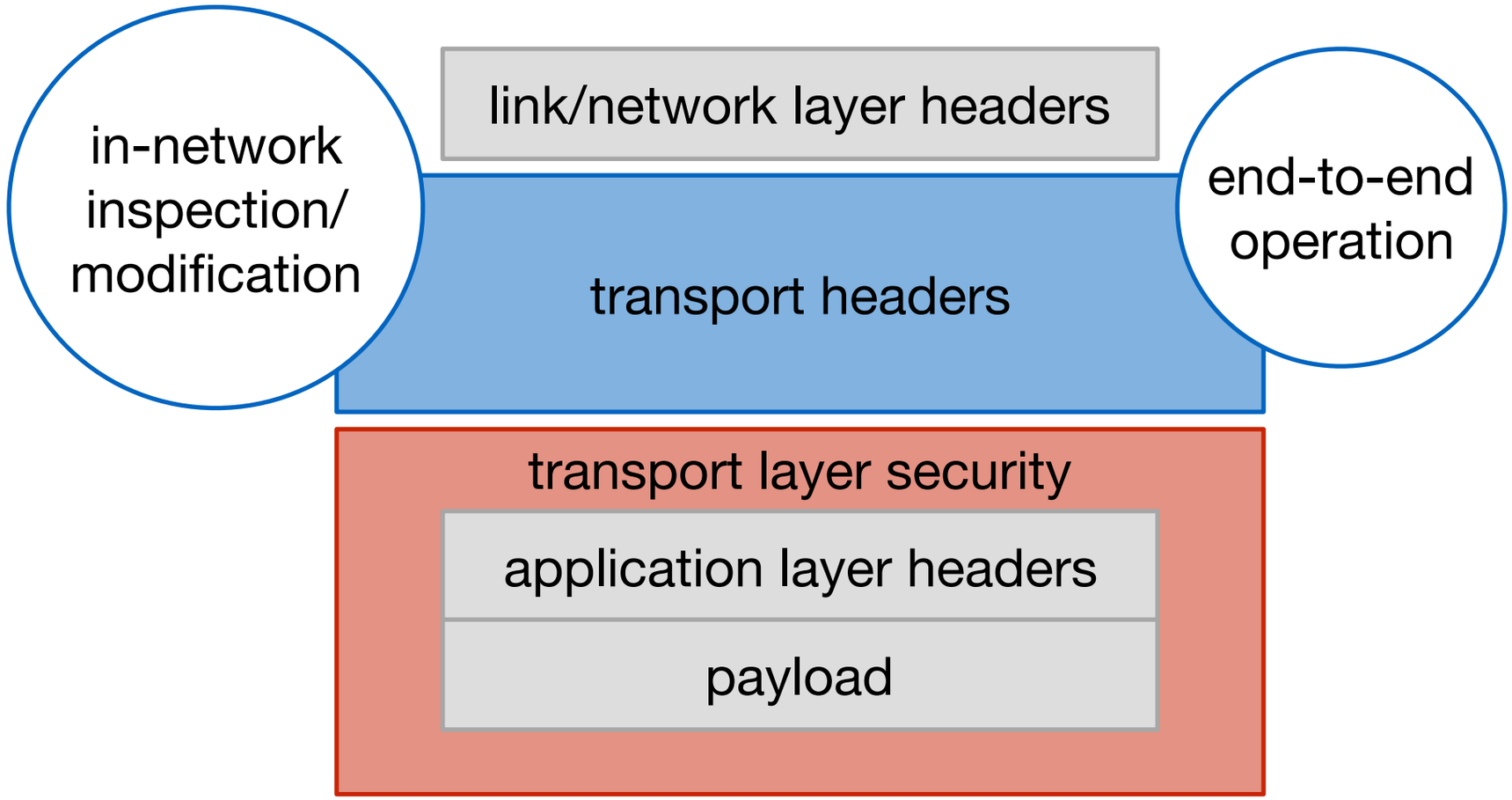
Transport protocol design, now with security!



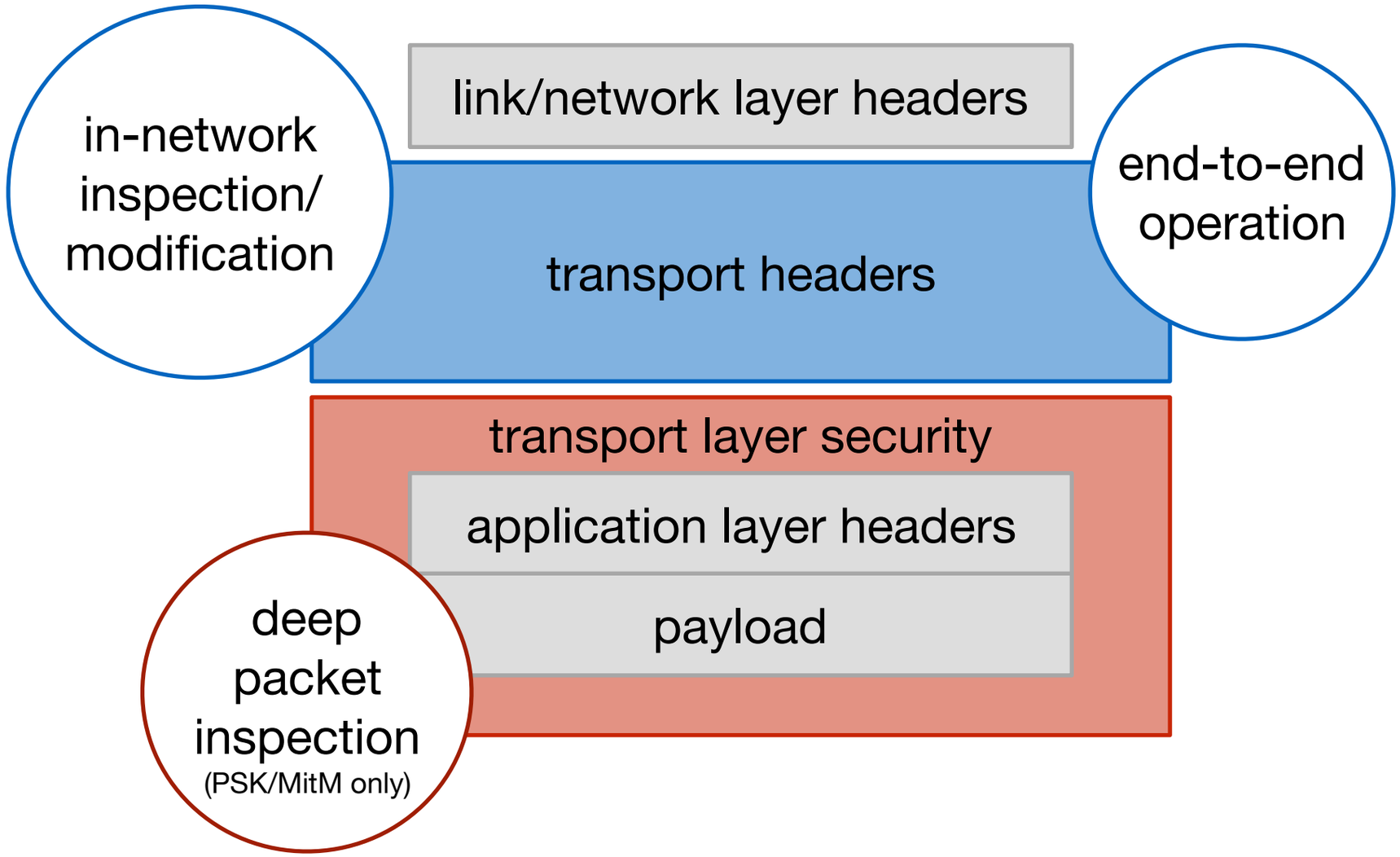
Transport protocol design, now with security!



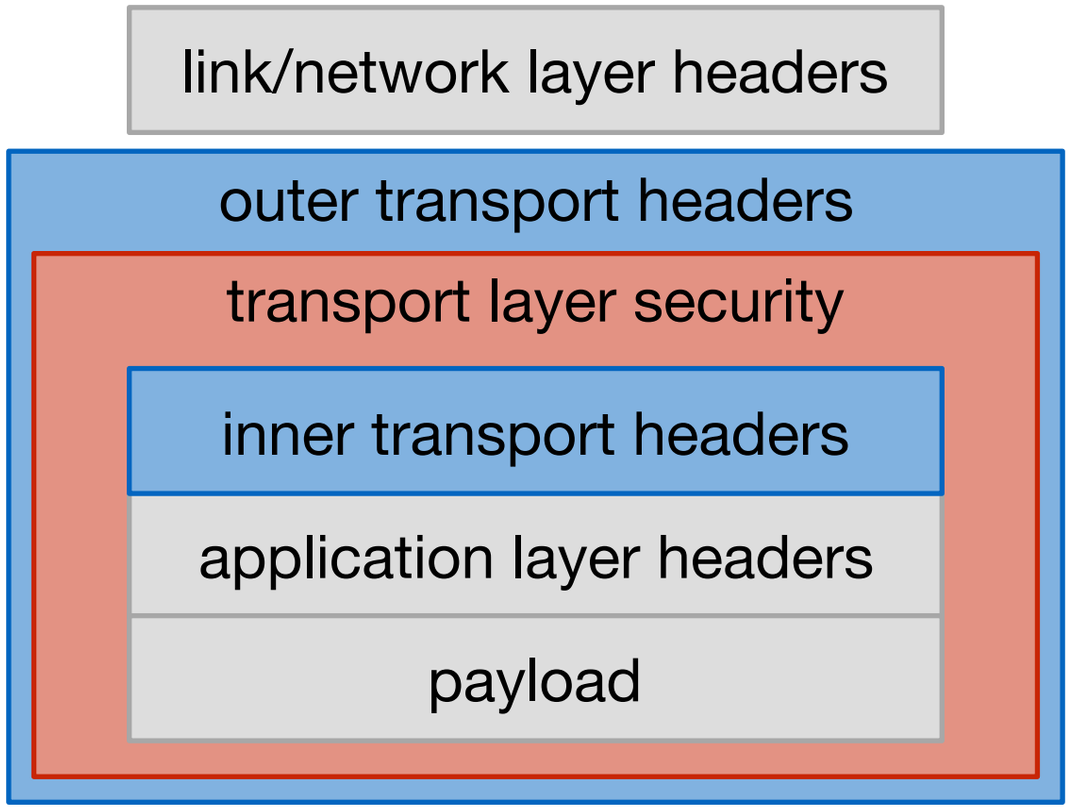
Transport protocol design, now with security!



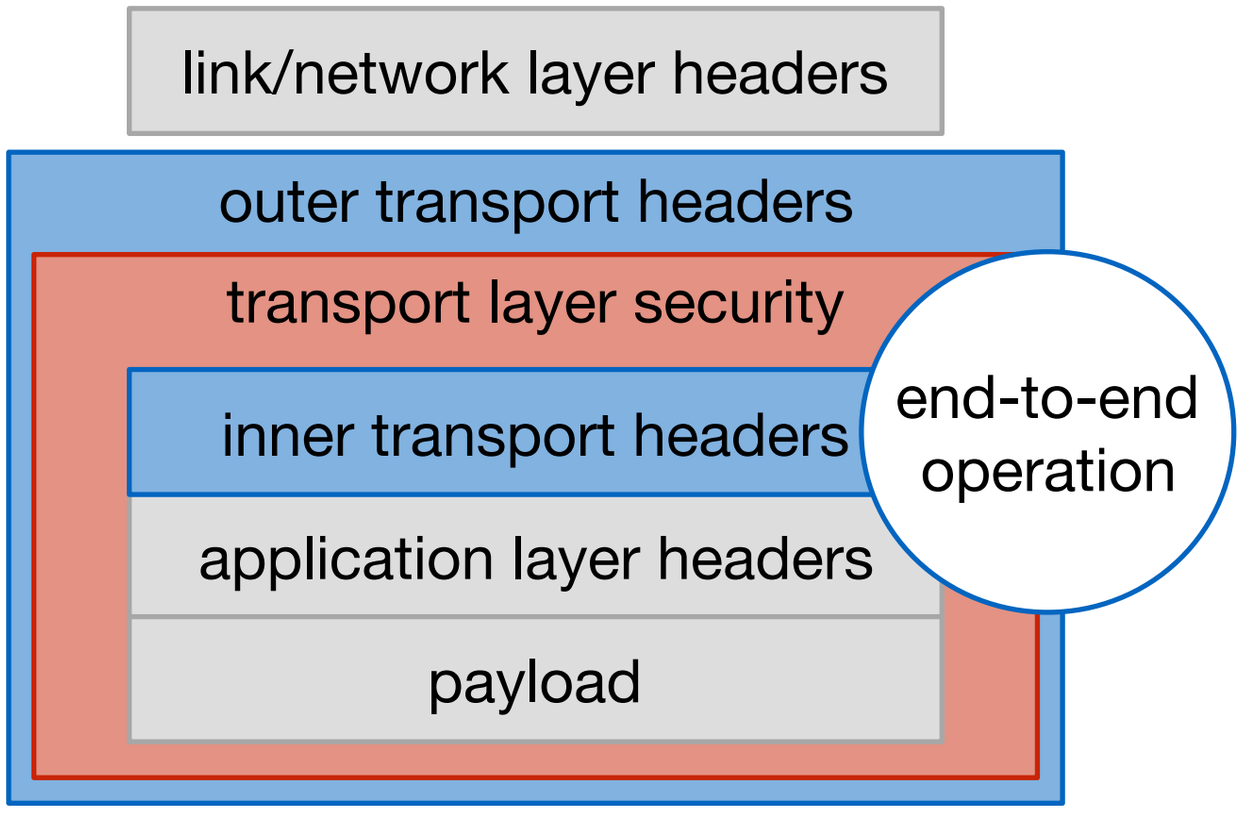
Transport protocol design, now with security!



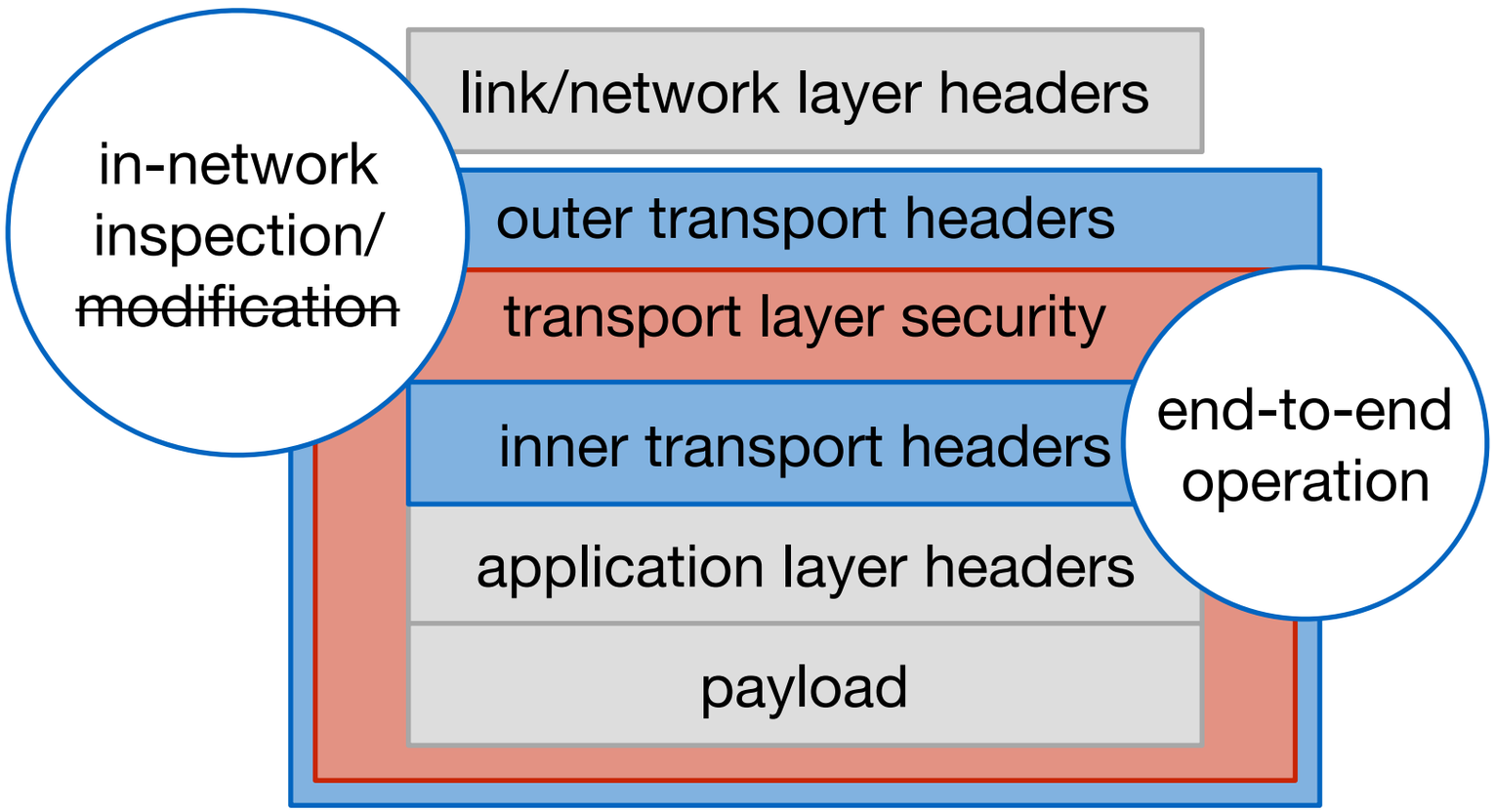
Encrypted transport protocol design



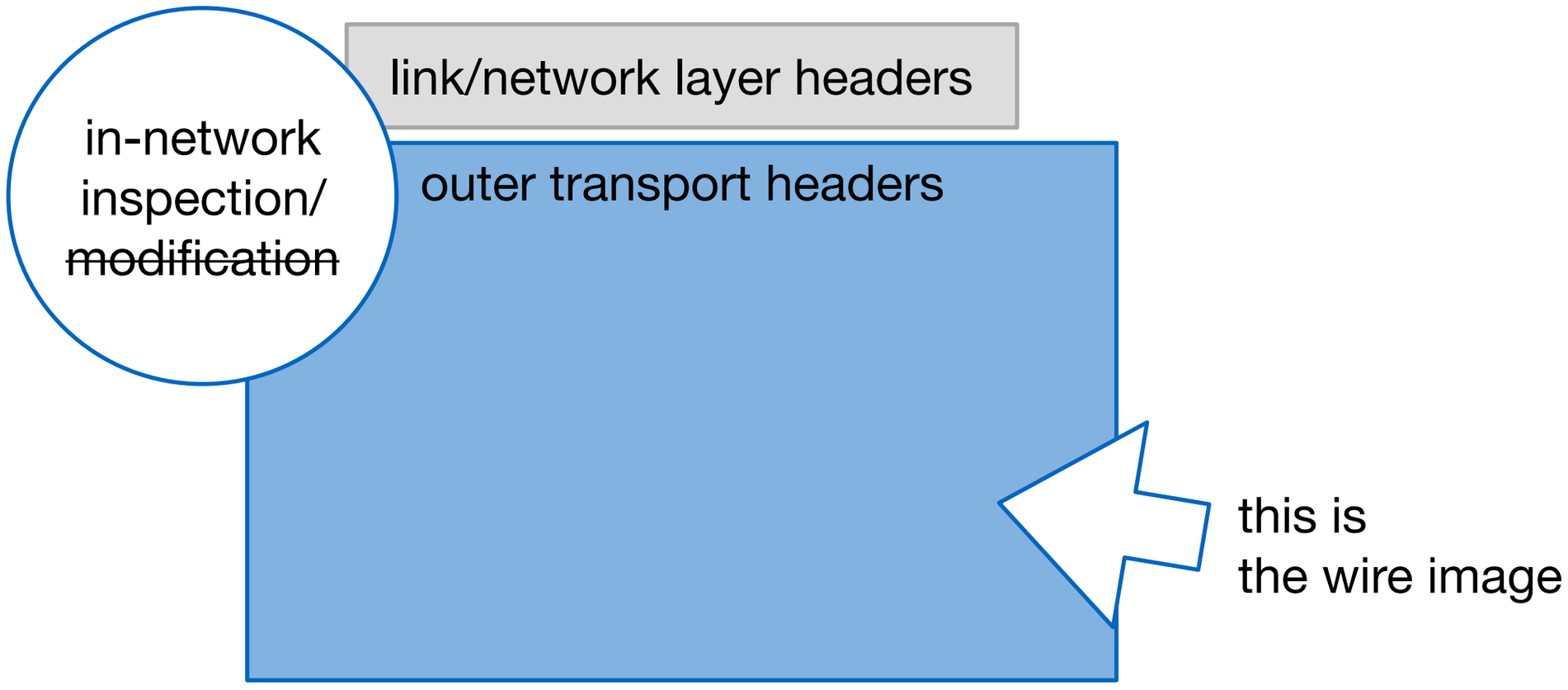
Encrypted transport protocol design



Encrypted transport protocol design



Encrypted transport protocol design



Three (and a half) levels of accessibility

- Unprotected (as TCP today): all bits can be seen at all points along the path, and can be modified without endpoint knowledge.
  - Supports manipulation of transport internals.
- Integrity-protected (e.g. QUIC outer header): all bits can be seen at all points along the path, but modification is endpoint detectable (and leads to packet rejection).
  - "Scratch space" (e.g. PLUS extended header): bits intended for modification on the path; content not protected, length/meaning integrity-protected.
- Encrypted (e.g. QUIC frame headers): no bits can be interpreted, modification leads to corruption and rejection.

The Wire Image, more formally (draft-trammell-wire-image)

- The sequence of messages sent by each participant in the protocol
  - each expressed as a sequence of bits,
  - with an associated time at which each was sent.
- Only unencrypted bits have assignable semantics
  - encrypted size → upper bound on information content

- Separating transport machinery from the understandable parts of the wire image is new, presents both problems and opportunities.

Signaling in the wire image (draft-hardie-path-signals)

- Encrypting transport mechanics limits the availability of *implicit signaling* in transport headers to in-network functions.
- We have four options to deal with this:
  - Do nothing.
  - Replace transport information with network-layer signaling.
  - Add *explicit signals* on a per-transport basis
  - Add *explicit signals* on a common wire image shared by multiple transports

Signaling in the wire image (draft-hardie-path-signals)

- Encrypting transport mechanics limits the availability of *implicit signaling* in transport headers to in-network functions.
- We have four op
  - Do nothing.
  - Replace transport information with network layer signaling.
  - Add *explicit signals* on a per-transport basis
  - Add *explicit signals* on a common wire image shared by multiple transports

> Transport-layer headers were never meant to be used by the network. It's the network's fault for abusing them, and solely the problem of in-network function developers and users to solve.

Signaling in the wire image (draft-hardie-path-signals)

- Encrypting transport mechanics limits the availability of *implicit signaling* in transport headers to in-network functions.
- We have four options to deal with this:
  - Do nothing.
  - Replace transport information with network-layer signaling.
  - Add *explicit s*
  - Add *explicit s*
    multiple transports

Talking to the network is a network-layer function. Therefore any explicit signaling should happen in the network layer.

Signaling in the wire image (draft-hardie-path-signals)

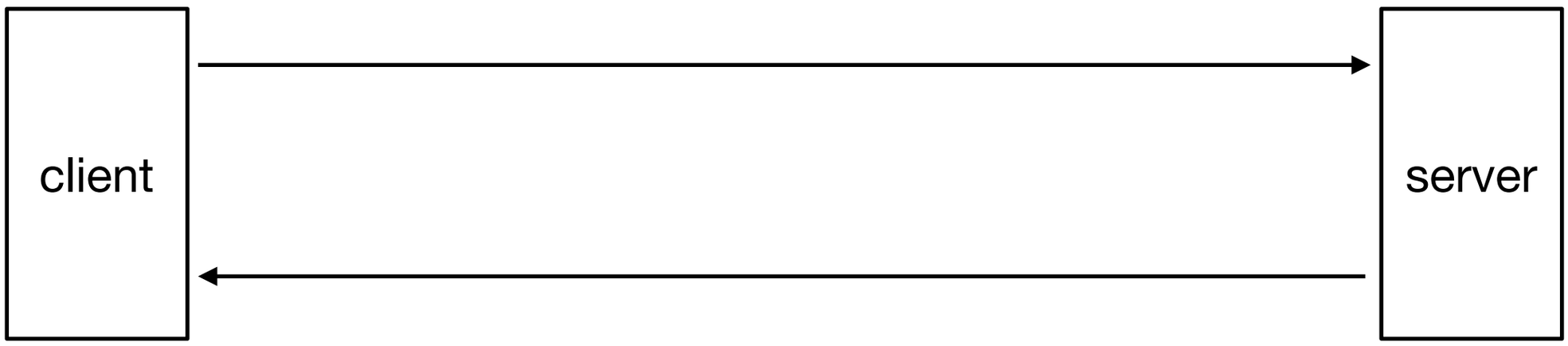
- Encrypting transport mechanics limits the availability of *implicit signaling* in transport headers to in-network functions.
- We have four options to deal with this:
  - Do nothing.
  - Replace transport information with network-layer signaling.
  - Add *explicit signals* on a per-transport basis
  - Add *explicit signals* on a common wire image shared by multiple transports

Signaling in the wire image (draft-hardie-path-signals)

- Encrypting transport mechanics limits the availability of *implicit signaling* in transport headers to in-network functions.
- We have four options to deal with this:
  - Do nothing.
  - Replace transport information with network-layer signaling.
  - Add *explicit signals* on a per-transport basis
  - Add *explicit signals* on a common wire image shared by multiple transports

An example explicit signal: draft-trammell-quic-spin

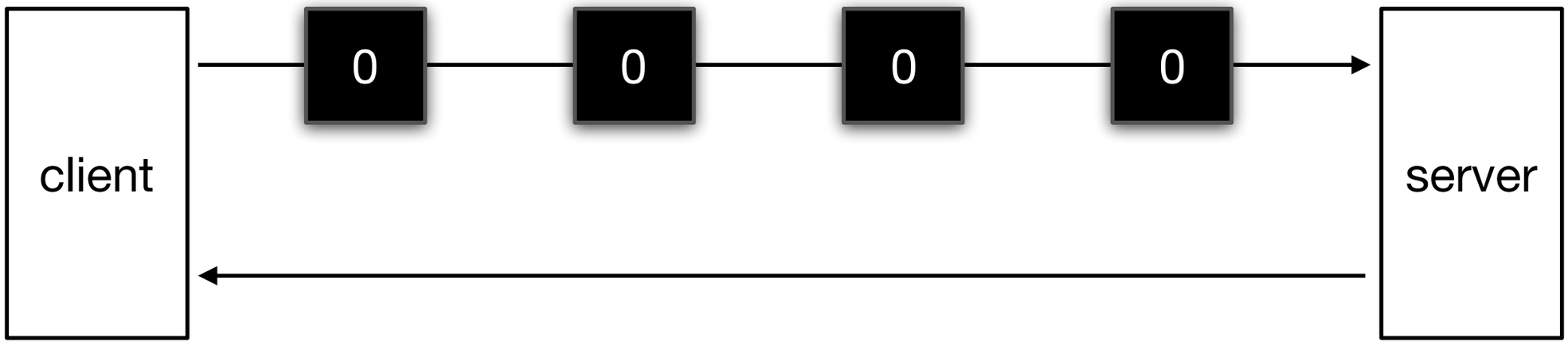
- Proposed bit in QUIC short header, visible/integrity-protected.
- Algorithm ensures bit changes 0→1 or 1→0 once per RTT.
- Allows on-path determination of RTT on a per-flow basis.
- Replaces TCP seq/ack and tsval/tescr for this purpose.



- Discussion in QUIC WG, 09:30-12:00 Thursday

An example explicit signal: draft-trammell-quic-spin

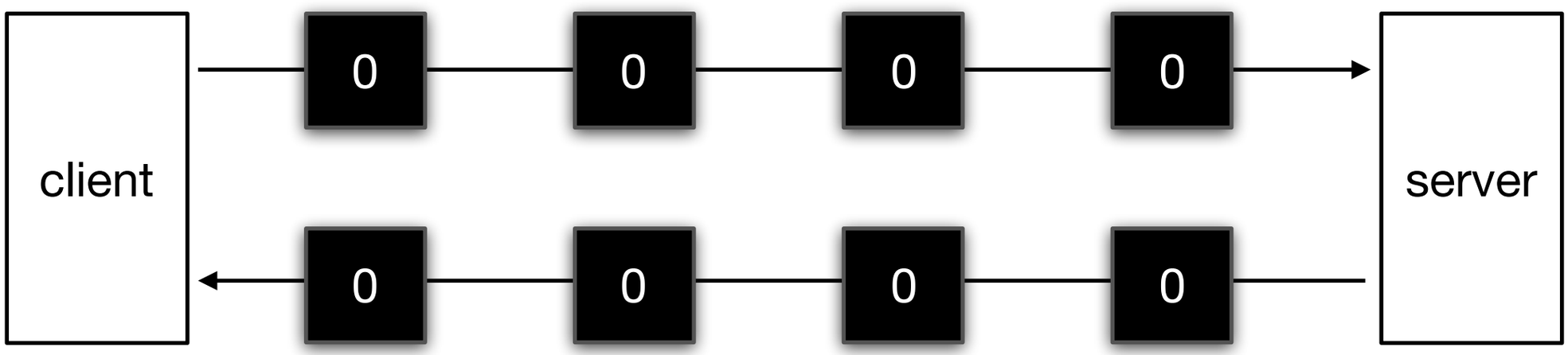
- Proposed bit in QUIC short header, visible/integrity-protected.
- Algorithm ensures bit changes 0→1 or 1→0 once per RTT.
- Allows on-path determination of RTT on a per-flow basis.
- Replaces TCP seq/ack and tsval/tescr for this purpose.



- Discussion in QUIC WG, 09:30-12:00 Thursday

An example explicit signal: draft-trammell-quic-spin

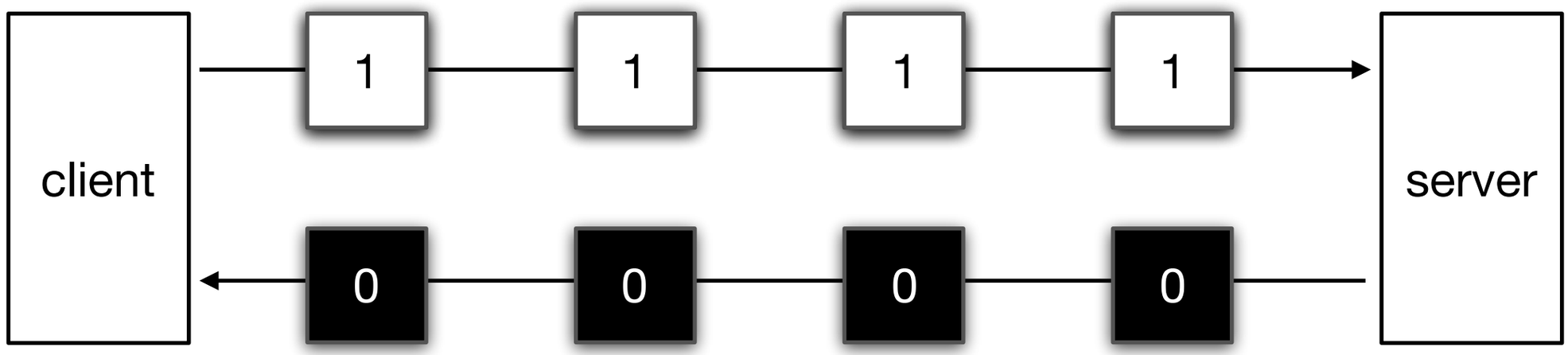
- Proposed bit in QUIC short header, visible/integrity-protected.
- Algorithm ensures bit changes 0→1 or 1→0 once per RTT.
- Allows on-path determination of RTT on a per-flow basis.
- Replaces TCP seq/ack and tsval/tescr for this purpose.



- Discussion in QUIC WG, 09:30-12:00 Thursday

An example explicit signal: draft-trammell-quic-spin

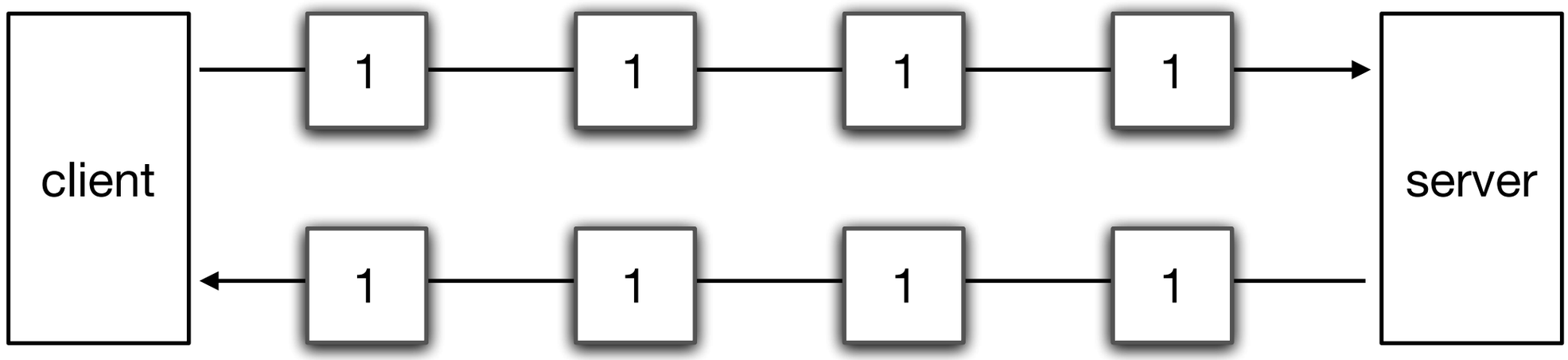
- Proposed bit in QUIC short header, visible/integrity-protected.
- Algorithm ensures bit changes 0→1 or 1→0 once per RTT.
- Allows on-path determination of RTT on a per-flow basis.
- Replaces TCP seq/ack and tsval/tescr for this purpose.



- Discussion in QUIC WG, 09:30-12:00 Thursday

An example explicit signal: draft-trammell-quic-spin

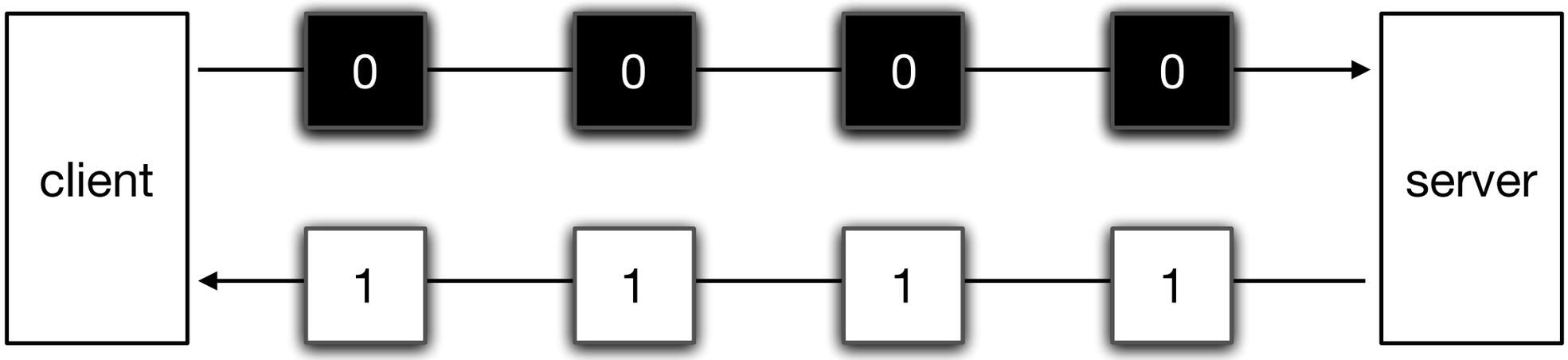
- Proposed bit in QUIC short header, visible/integrity-protected.
- Algorithm ensures bit changes 0→1 or 1→0 once per RTT.
- Allows on-path determination of RTT on a per-flow basis.
- Replaces TCP seq/ack and tsval/tescr for this purpose.



- Discussion in QUIC WG, 09:30-12:00 Thursday

An example explicit signal: draft-trammell-quic-spin

- Proposed bit in QUIC short header, visible/integrity-protected.
- Algorithm ensures bit changes 0→1 or 1→0 once per RTT.
- Allows on-path determination of RTT on a per-flow basis.
- Replaces TCP seq/ack and tsval/tescr for this purpose.



- Discussion in QUIC WG, 09:30-12:00 Thursday

The Way Forward

- Header encryption is necessary for transport layer evolution.
  - Every visible bit of information in a protocol's wire image will eventually be used → we now know "everyone can see everything" is the wrong approach.
- Finding a balance between security, privacy, evolvability, and maintenance of current in-network functions is crucial.
  - Explicit signaling to the path must be done on a per-protocol, per-function basis.
  - Signal design must consider *who* needs to see *what* and *why*, and provide the *minimum necessary information* to drive the function.