

Path Layer UDP Substrate (PLUS) Technical Considerations

Brian Trammell, PLUS BoF
IETF 96 Berlin - 21 July 2016

segue

- (this slide is a placeholder for a clean segue from Ted's and Natasha's talks, if necessary)

P A R E N T A L

A D V I S O R Y

E X P L I C I T

C O O P E R A T I O N

Explicit Cooperation

- “Implicit cooperation” between endpoints and middleboxes already widespread in the Internet,
 - where “cooperation” may be the wrong term: some hacks and workarounds are quite hostile.
- We present making this cooperation explicit, and handing control over it to the endpoints, as a way to reduce tension in the end-to-end tussle.
- We declare that everything devices on path don’t need to see (including transport headers) should be encrypted to prevent future unauthorized “implicit cooperation”.

P A R E N T A L

A D V I S O R Y

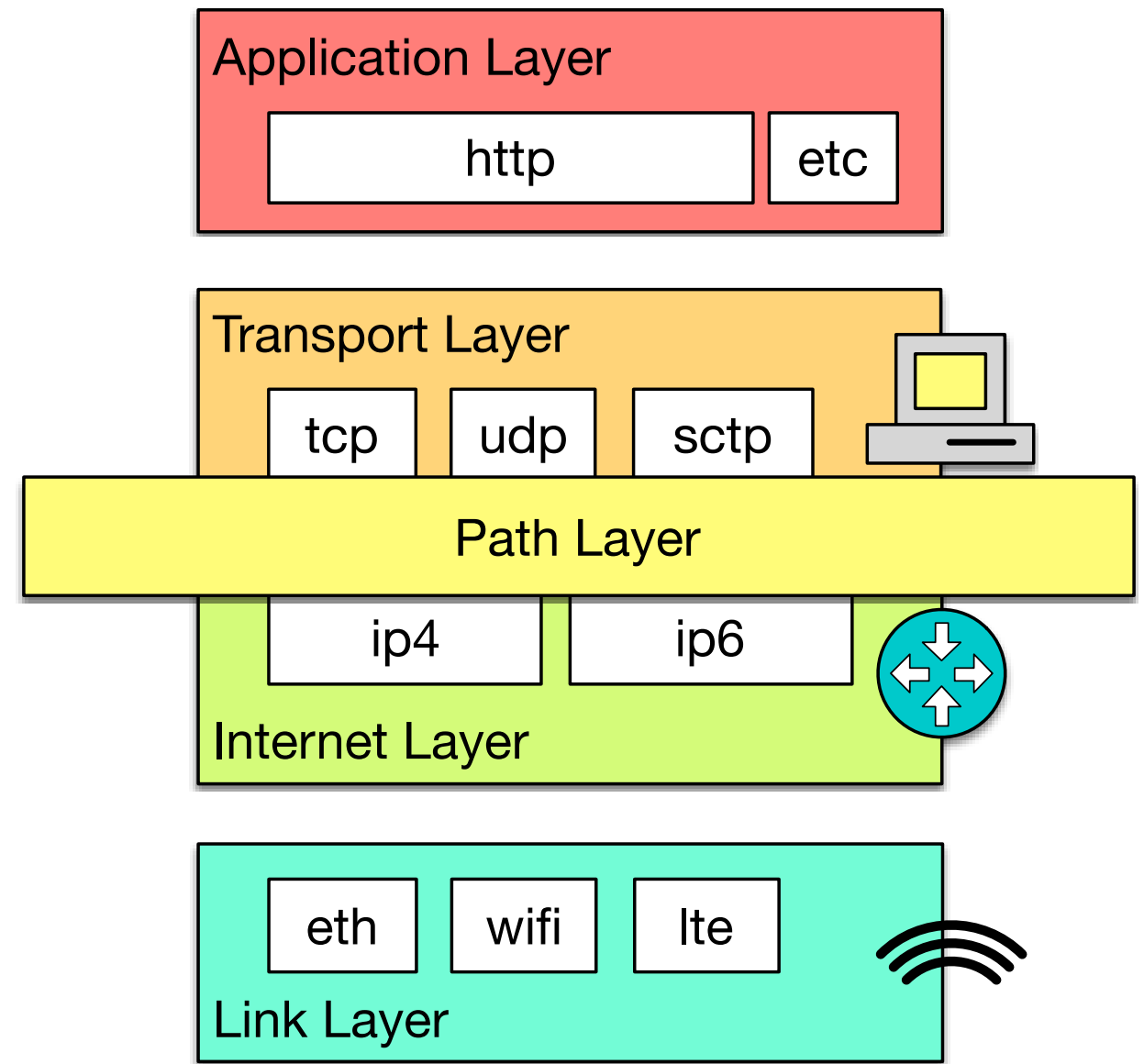
E X P L I C I T S I G N A L

Explicit Signaling

- “Implicit signaling” between endpoints and middleboxes already widespread in the Internet,
- We present making this signaling explicit, and handing control over it to the endpoints, as a way to reduce tension in the end-to-end tussle.
- We declare that everything devices on path don’t need to see (including transport headers) should be encrypted to prevent future unauthorized implicitness.

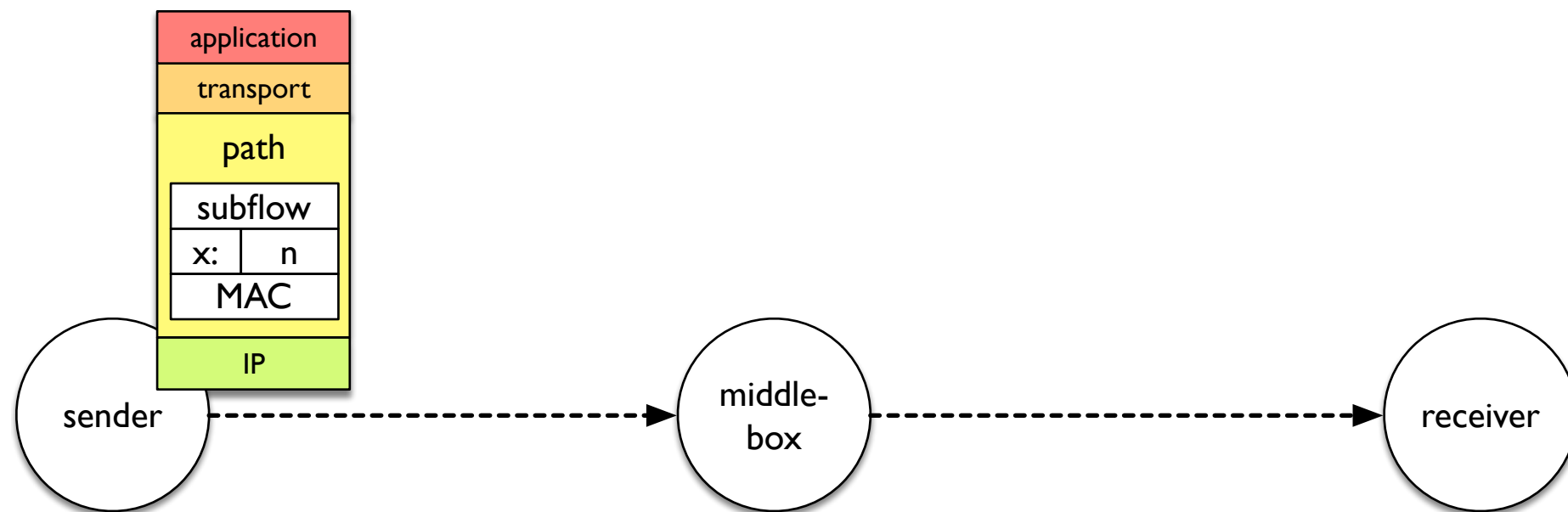
Introducing the Path Layer

- Network: hop-by-hop, no data-plane state.
- Transport: end-to-end, stateful.
- Implicit layer in between where all the state in the network lives.
- PLUS makes this explicit.

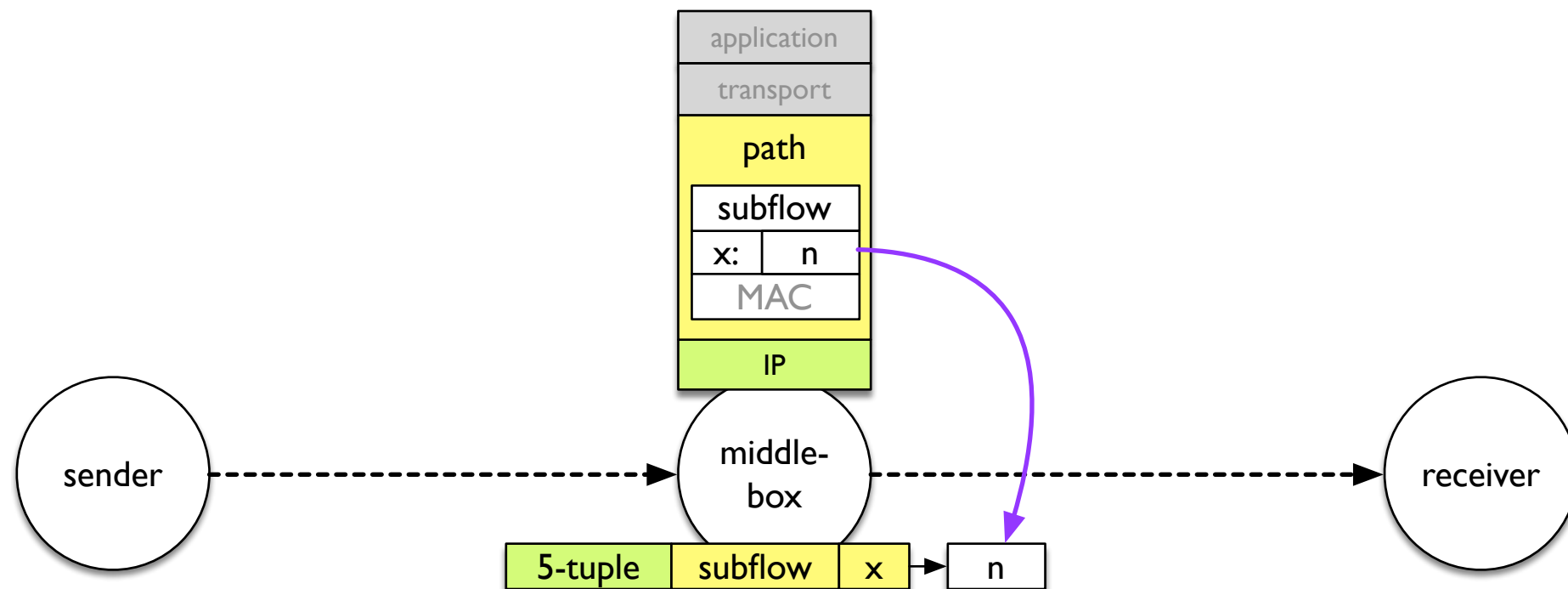


Mechanisms

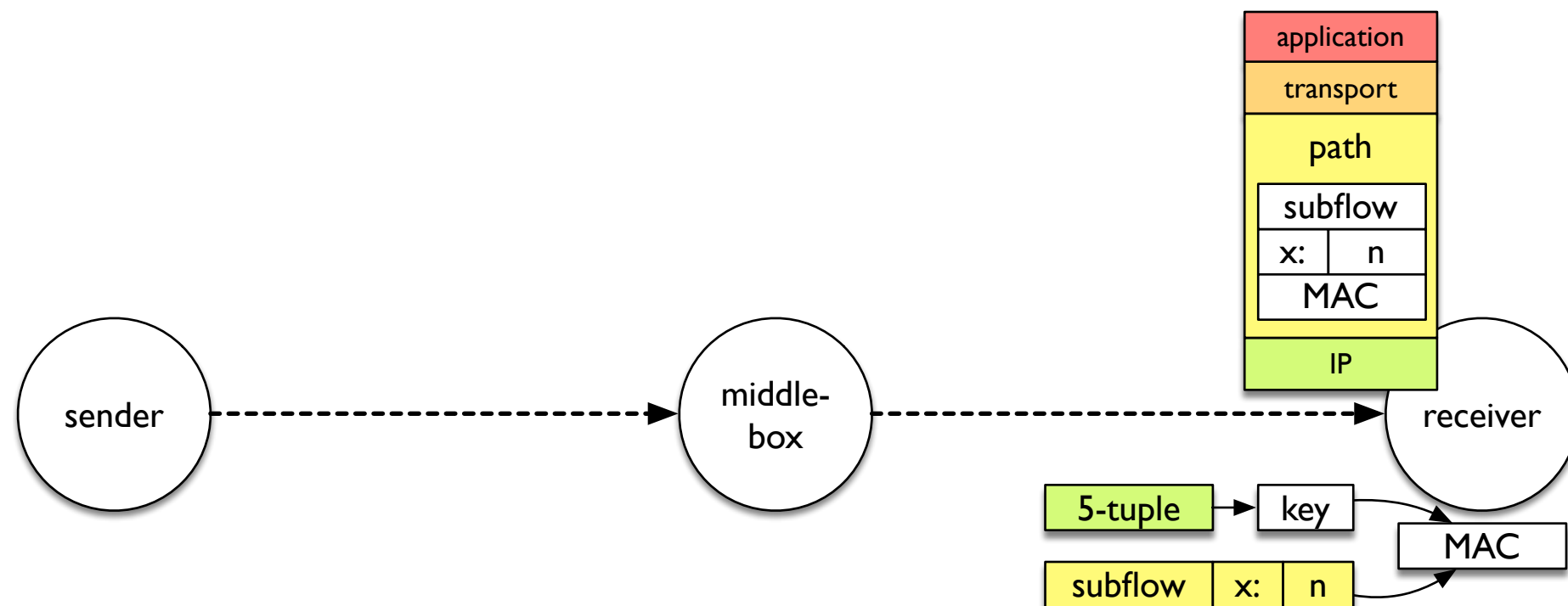
Endpoint to Path (sender-side)



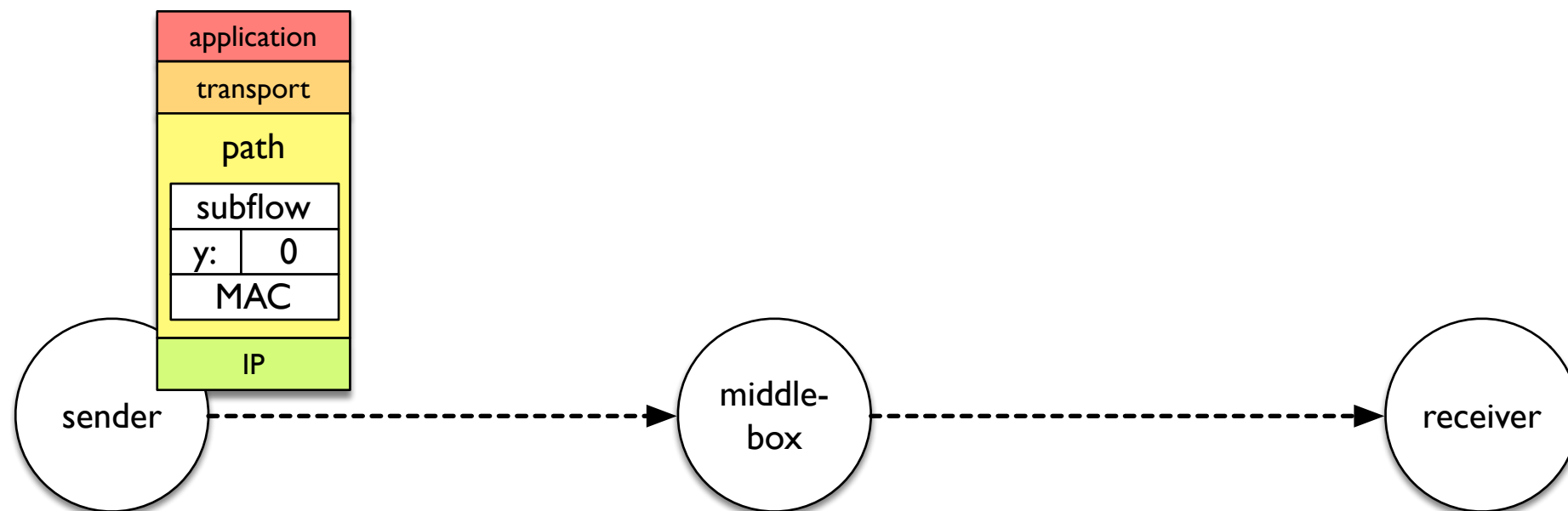
Endpoint to Path (on-path)



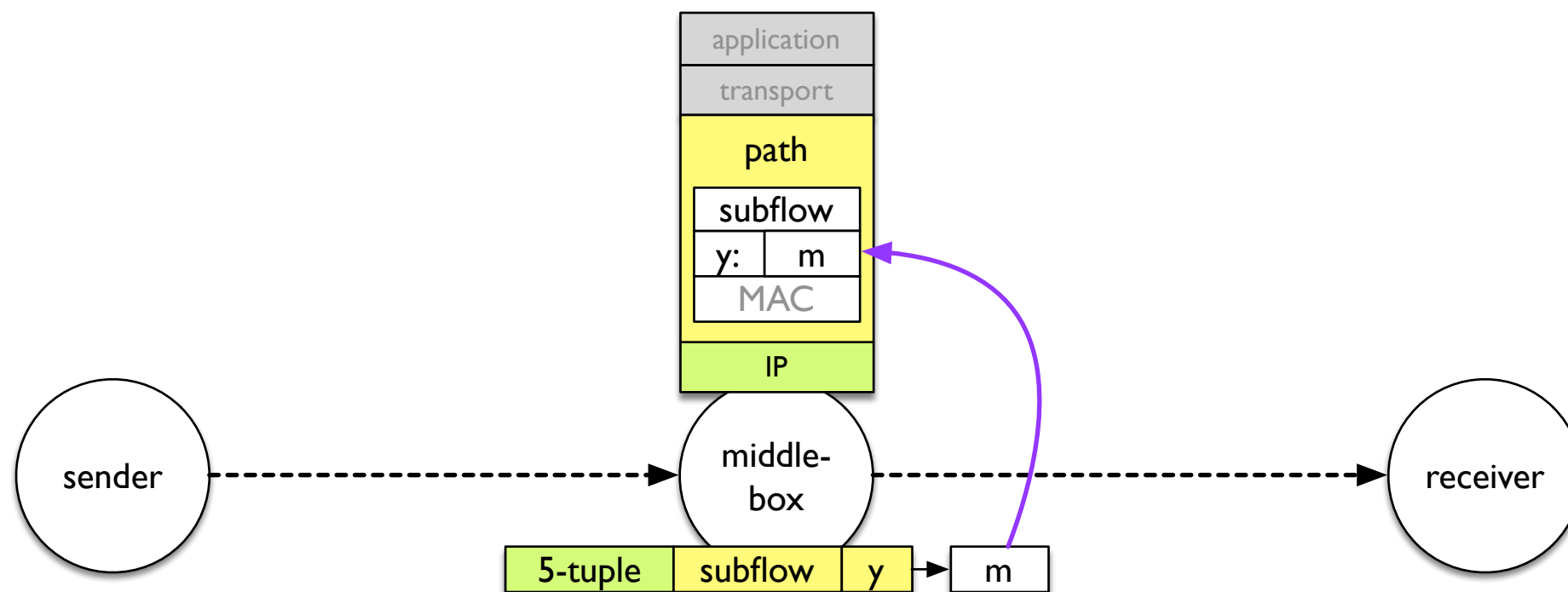
Endpoint to Path (receiver-side)



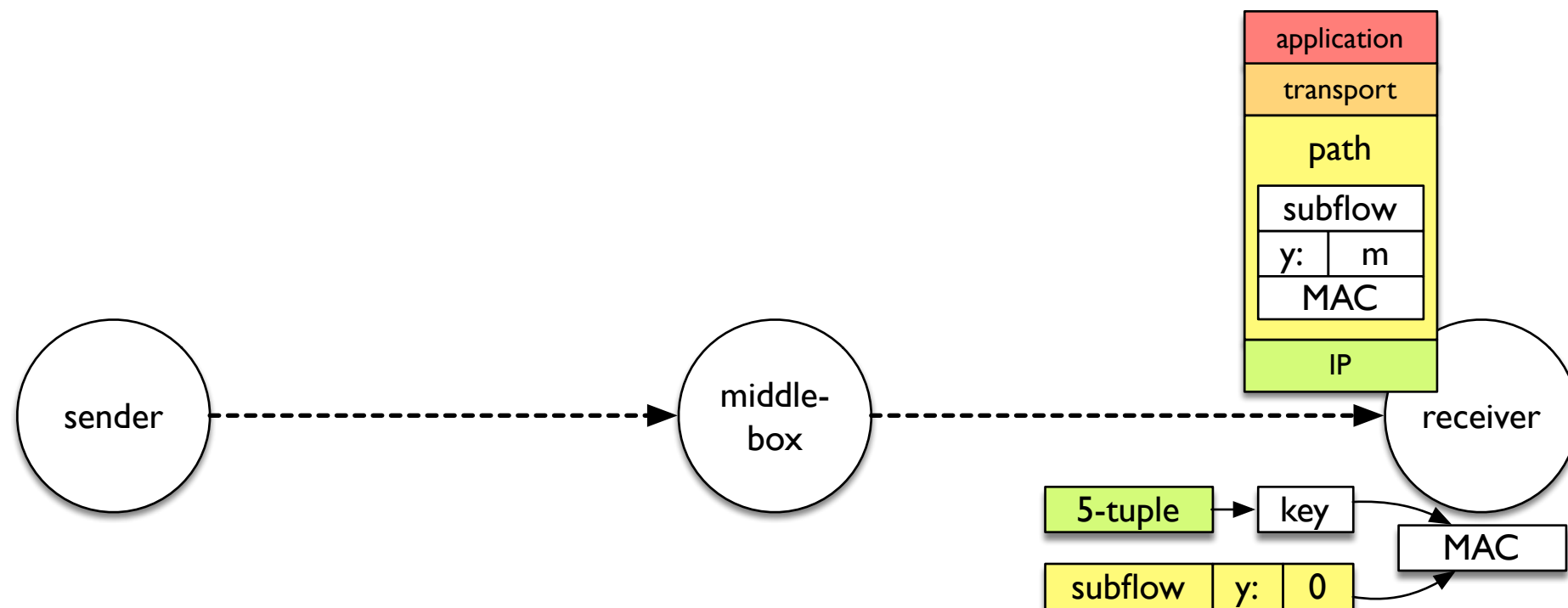
Path to Receiver (sender-side)



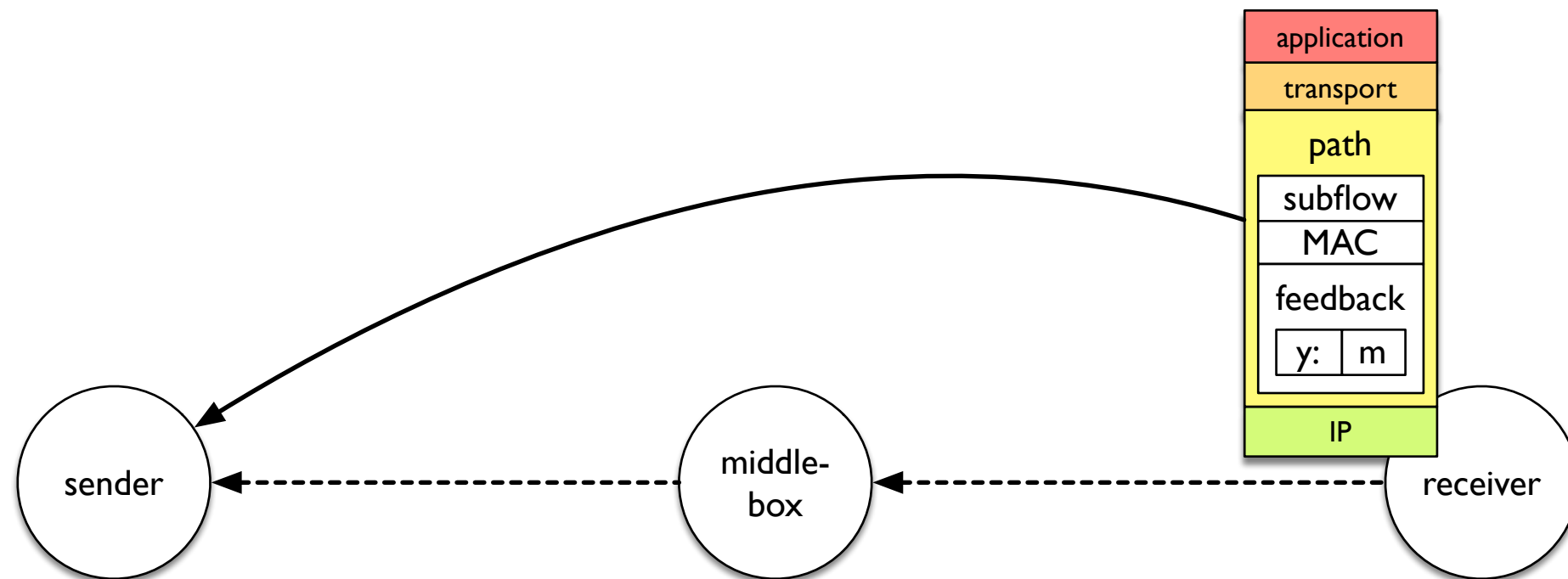
Path to Receiver (on-path)



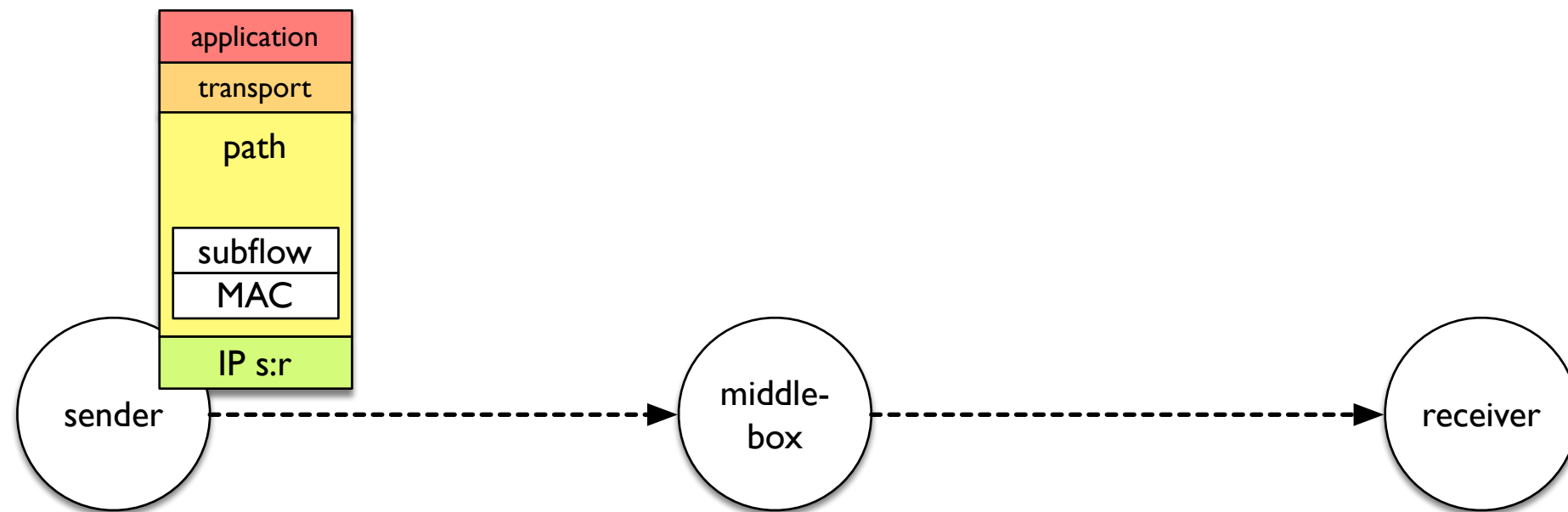
Path to Receiver (receiver-side)



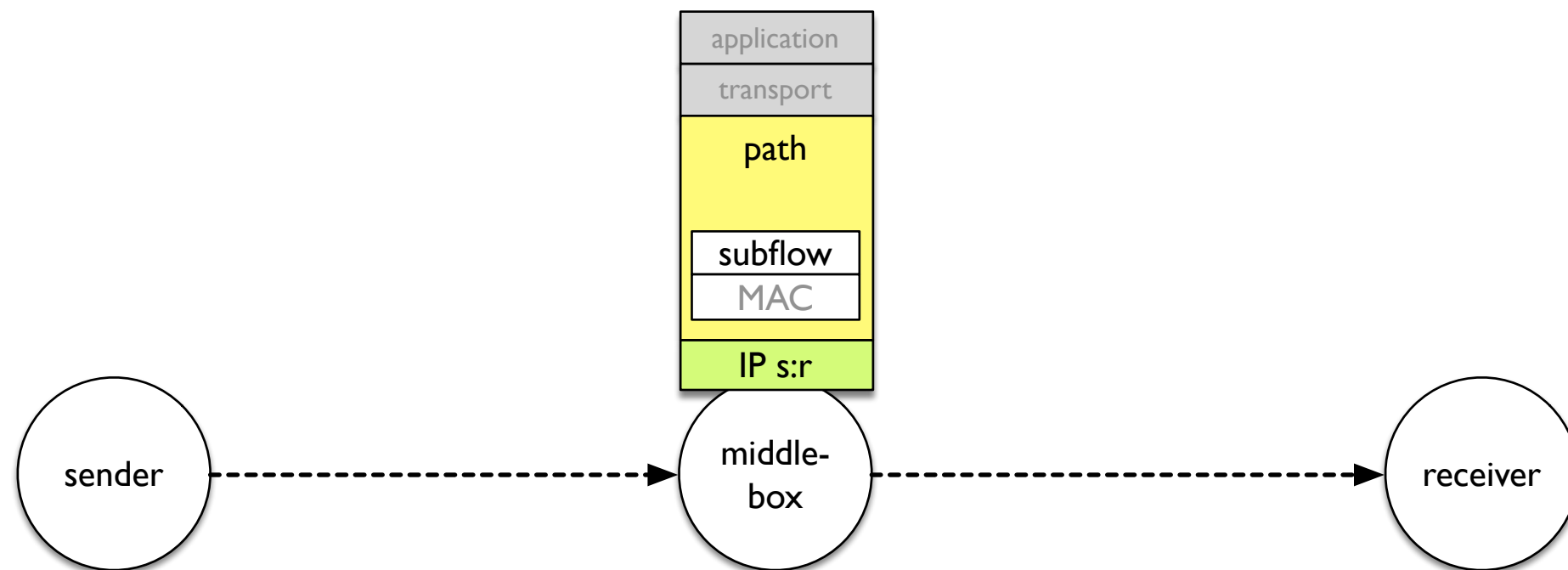
Receiver Feedback



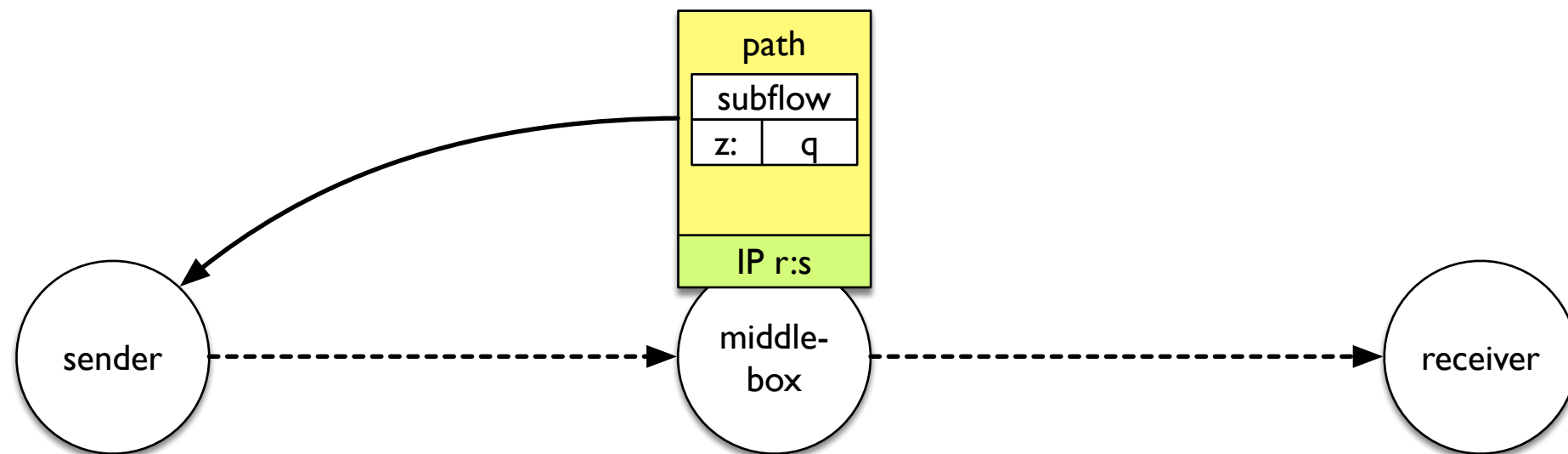
Path Direct to Sender (sender-side)



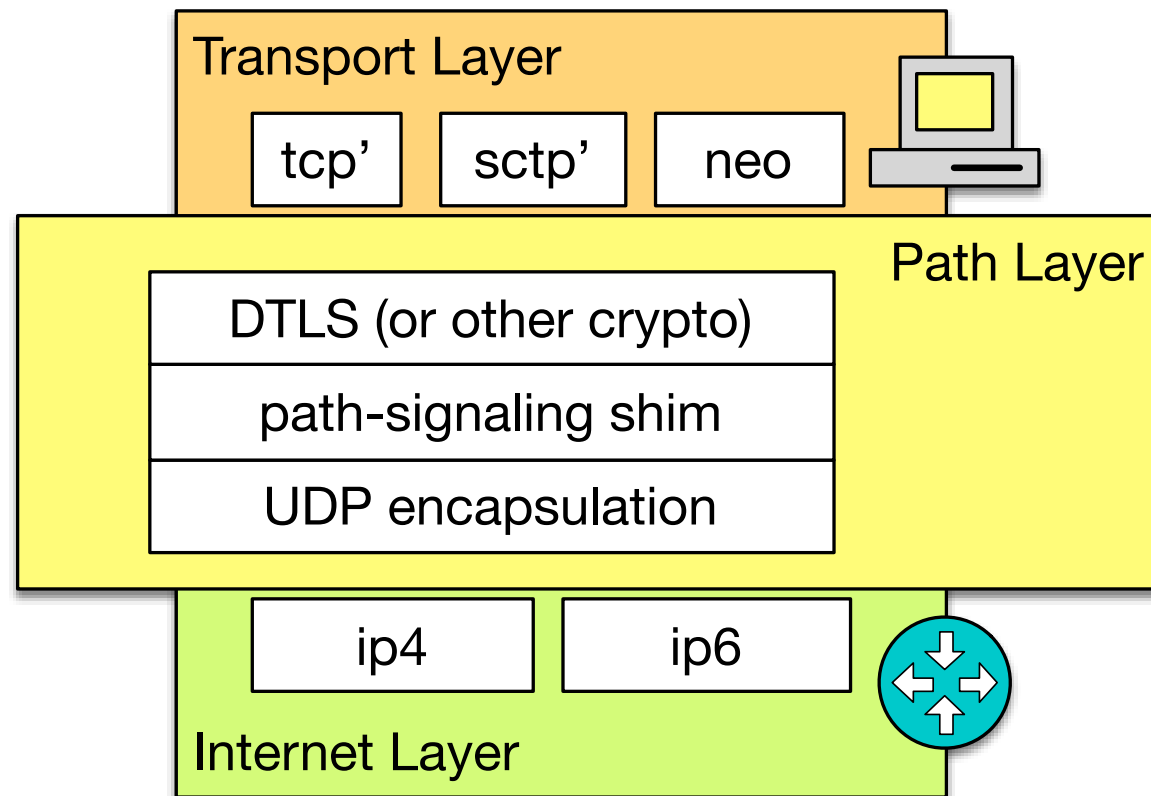
Path Direct to Sender (on-path)



Path Direct to Sender (feedback)



Anatomy of the Path Layer



- UDP encapsulation
 - userspace implementation
 - ports for NAT
 - ~95% deployable today
- encoding for signaling mechanisms
- crypto to protect transport headers and above

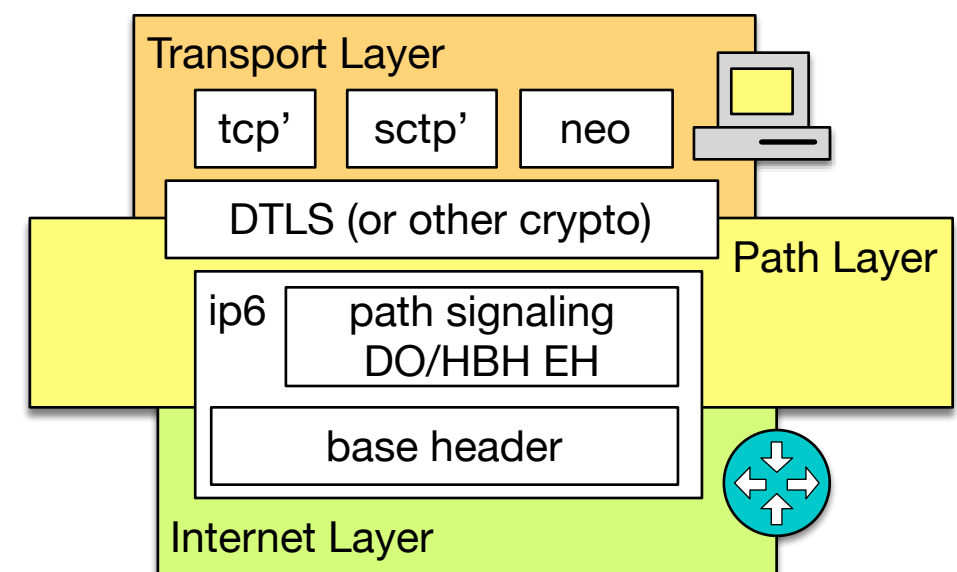
meanwhile, on the
spud@ietf.org list...

Is this a user tracking and network neutrality violation machine?

- Will it be possible for a middlebox to use PLUS to insert user identifiers in the server-bound stream of a client-server protocol?
 - **No**, unless the client specifically requests it.
 - (Note: possible without PLUS, out of band, today)
- Will it be possible to use PLUS to require a client to insert a particular kind of metadata into a stream?
 - Bad news: yes; no technical solution exists here.
 - (Worse news: also many ways to do this without PLUS)
 - Good news: PLUS brings **transparency** to this behavior.

Can we use IPv6 extension headers?

- IPv6 extension headers can be used to implement PLUS mechanisms
 - Ignore IPv4 in future deployments
 - DO to expose to path: hack, but more deployable
 - HBH to communicate with path: cleaner, but deployment issues
- DO/HBH already supported in most socket APIs
- But: much more impaired than UDP (draft-ietf-v6ops-ipv6-ehs-in-real-world-02)



Can we make transport innovation work without cooperation?

- **draft-herbert-transport-over-udp**
 - Standardize x over DTLS over UDP stack.
 - Fix transport innovation problem with crypto.
 - Breaks most middleboxes except NATs
 - This is a feature.
- True: sometimes, cooperation is useless.
 - Equivalent to PLUS when neither endpoint decides to expose anything to the path.

Can we use UDP Options?

- **draft-touch-tsvwg-udp-options**
 - add option space to UDP in a “gap” between the UDP and IP lengths of a packet.
 - Allows optional data to be added to existing UDP applications in a backward compatible manner.
- Proposal: use this option space for PLUS
- Are these the same problem at all?
 - No advantage over a UDP-based shim layer.
 - Needs kernel support: no userspace implementation.
 - No fast-path recognition or packet/property association.

and in conclusion...

Things we need

- A mechanism for making widespread cooperation between endpoints and middleboxes explicit
- Endpoint control over explicit cooperation
- A clear boundary between what the path can see and what it cannot, enforced by encryption
- A design for this facility that deploys on the endpoints from day zero