

PLUS (and SPUD)

Brian Trammell — 30 July 2020

A Path Layer for the Internet

Enabling Network Operations on Encrypted Traffic

Mirja Kühlewind, Tobias Bühler, **Brian Trammell**, ETH Zürich
Stephan Neuhaus, Roman Müntener, Zürich Univ. of Applied Sciences
and Gorrry Fairhurst, Univ. of Aberdeen

IEEE/IFIP Conf. on Network and Service Management, Tokyo, 28 November 2017



measurement and architecture for a middleboxed internet

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..



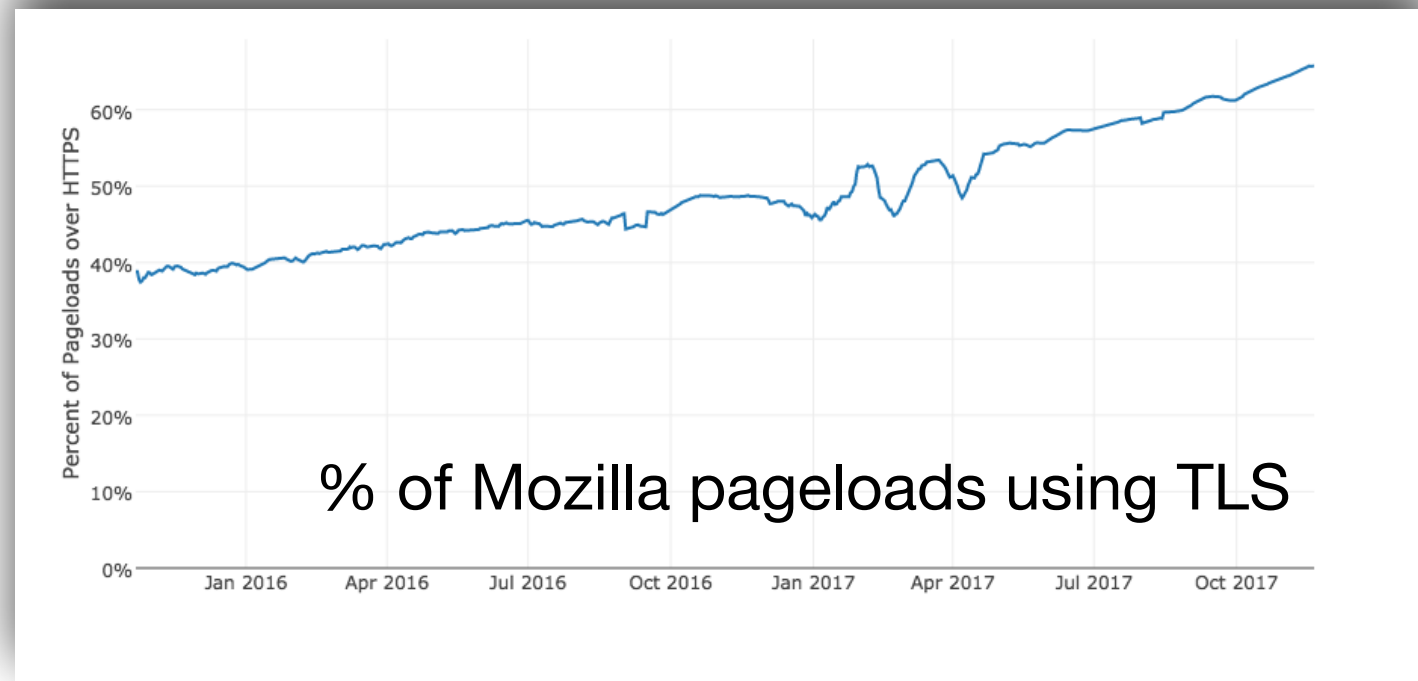
Supported by the Swiss State Secretariat for Education, Research and Innovation under contract number 15.0268. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the Swiss Government.



Increasing Deployment of Encryption

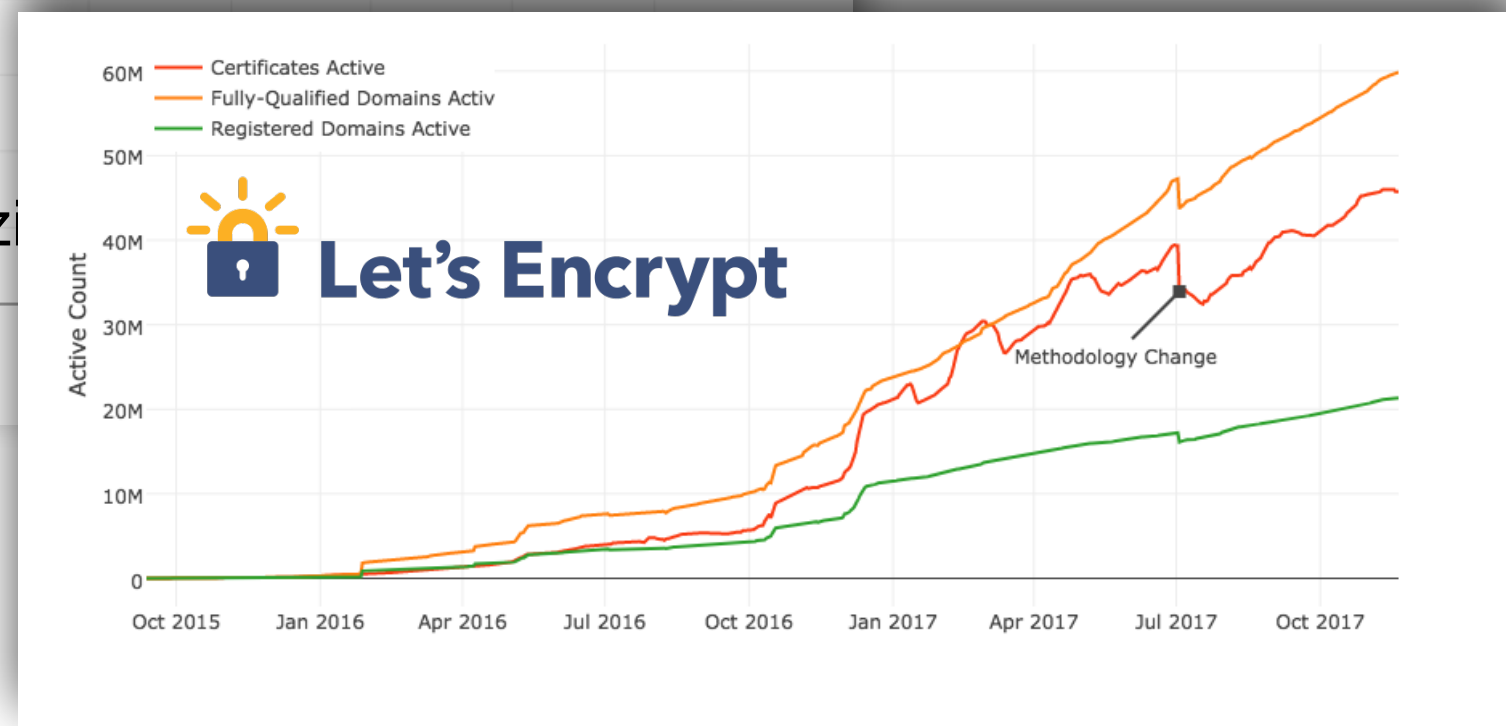
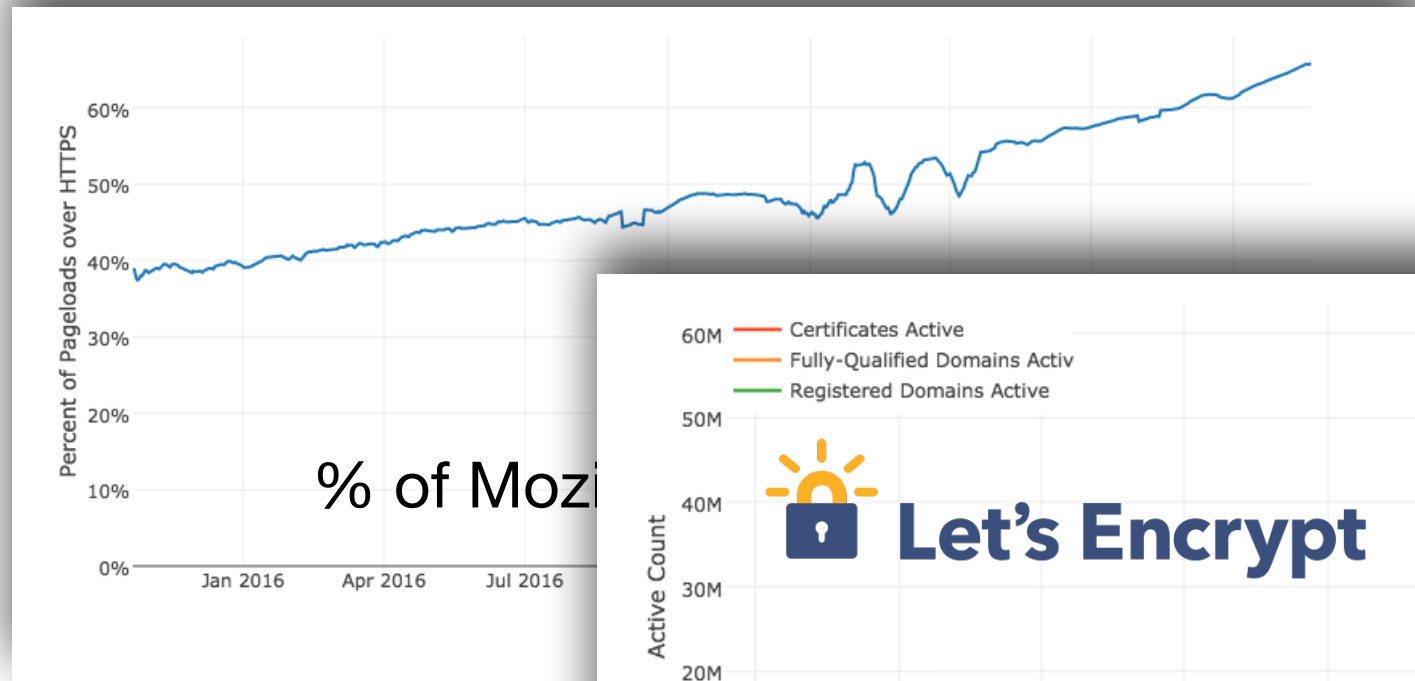


Increasing Deployment of Encryption



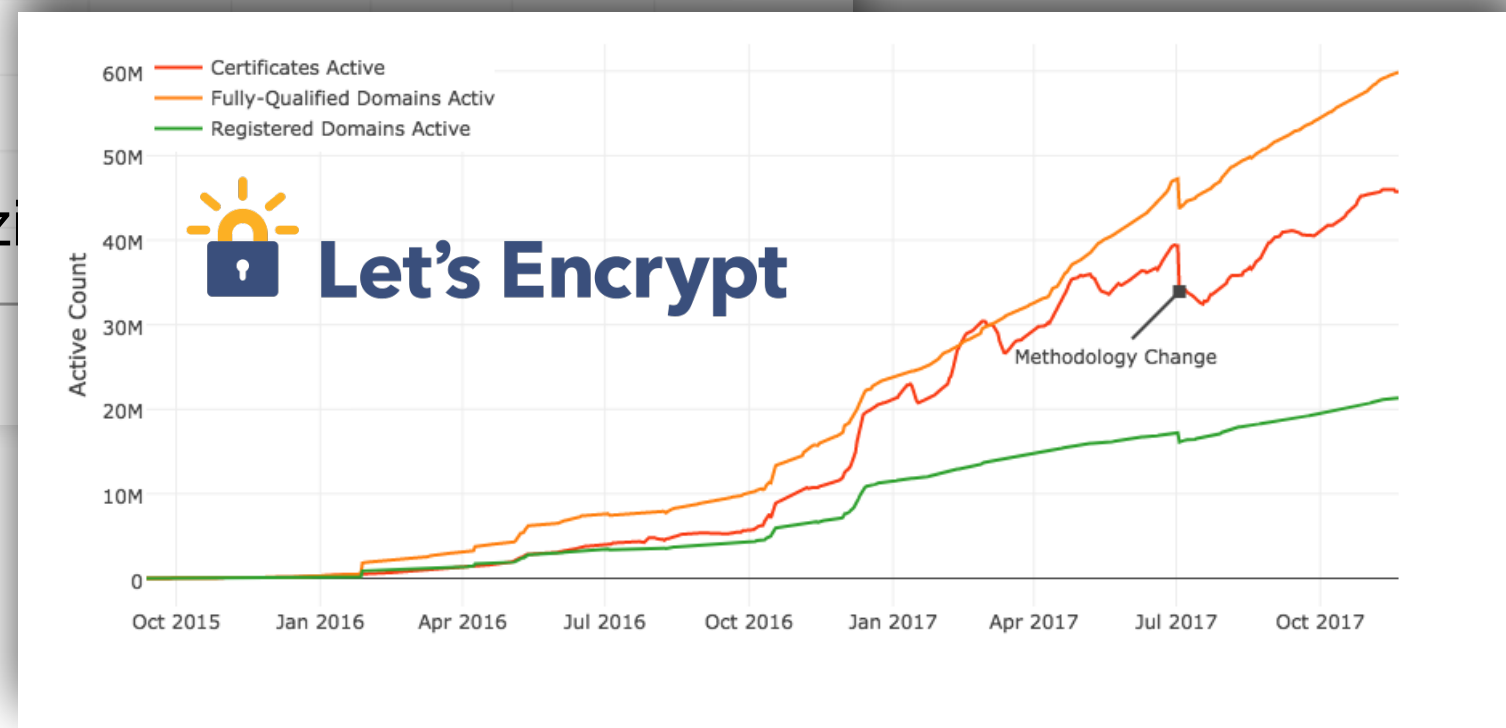
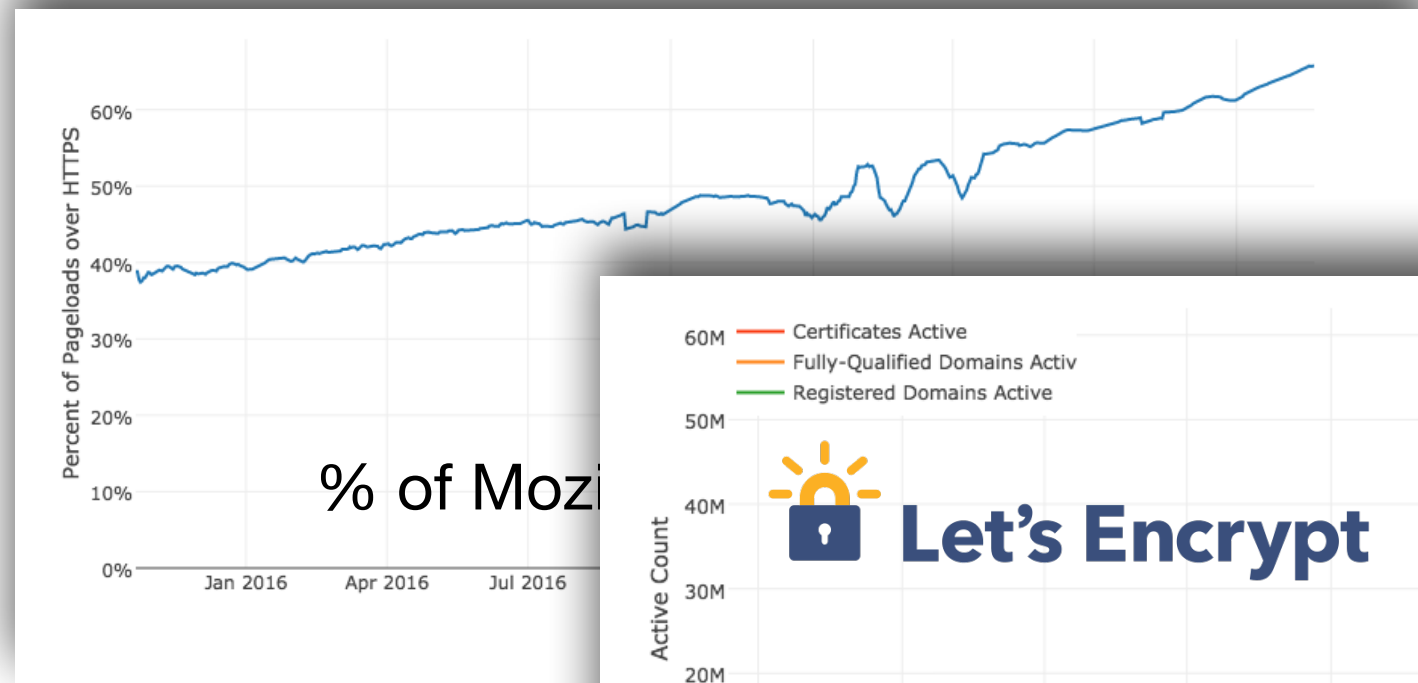


Increasing Deployment of Encryption





Increasing Deployment of Encryption



- → No management function that needs cleartext access to application headers/payload will work on the new Internet.



Explicit Cooperation

- The cleartext party is over, and DPI is dead.
 - Encryption for privacy, security, *and protocol evolvability*.
- A third way: replace use of cleartext by in-network functions with ***endpoint-controlled signaling***.
- Explicit cooperation based on declarative, advisory signals requiring no trust between endpoints and path can reduce disruption driven by increased encryption.



Introducing the Path Layer

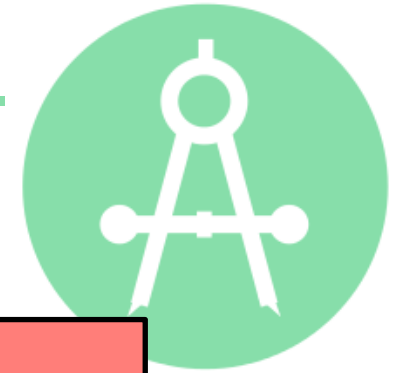
- The boundary between network (hop-by-hop, stateless) and transport (end-to-end, stateful) blurred by in-network state.
- Approach: add a layer to the stack to support these functions and use crypto to reinforce the boundary.

Application
(higher-level semantics)

Transport
(end to end streams/messages)

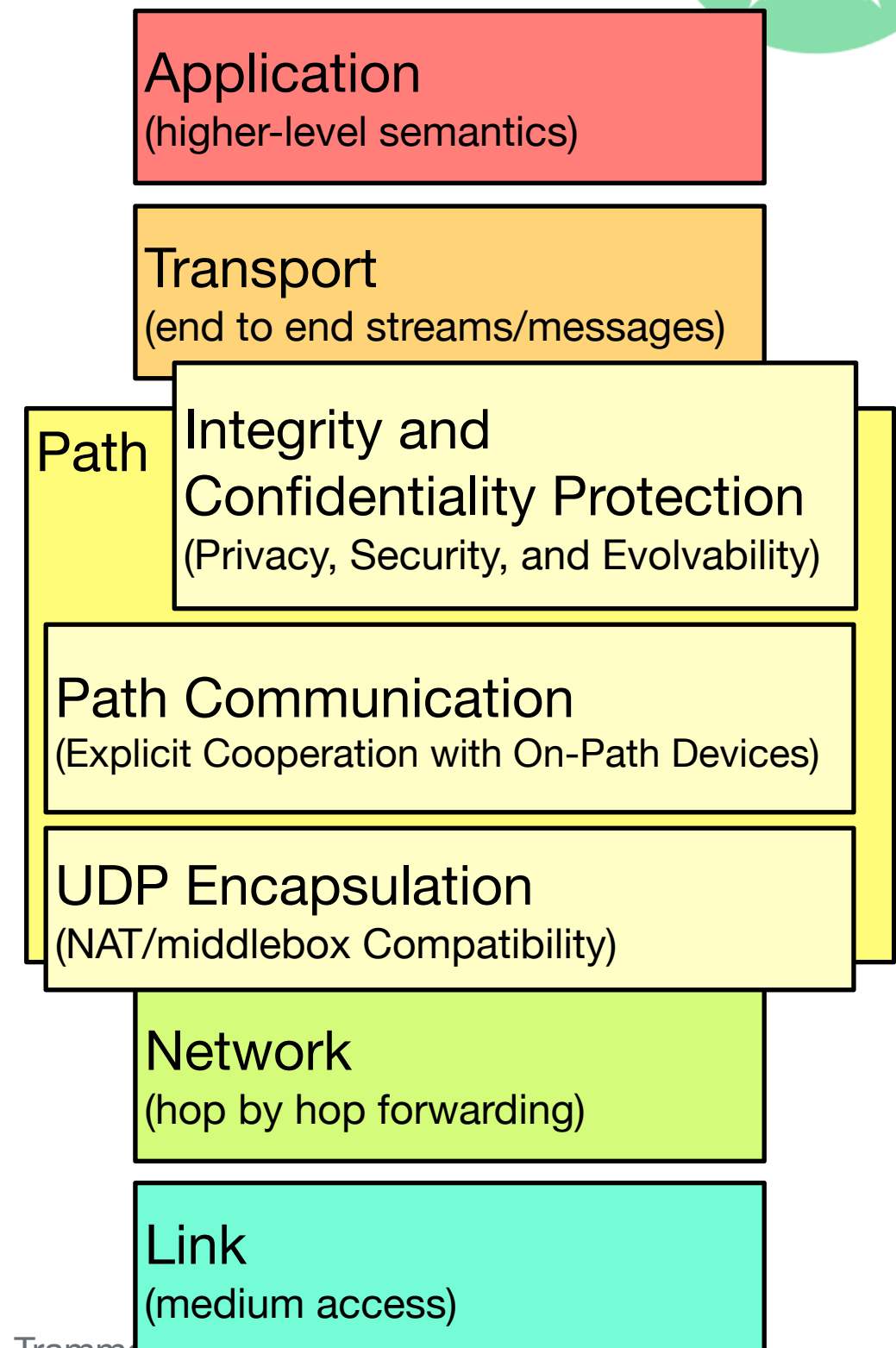
Network
(hop by hop forwarding)

Link
(medium access)



Introducing the Path Layer

- The boundary between network (hop-by-hop, stateless) and transport (end-to-end, stateful) blurred by in-network state.
- Approach: add a layer to the stack to support these functions and use crypto to reinforce the boundary.





Path Layer Principles

- An endpoint should be able to **explicitly expose signals** to be used by on-path devices. Everything not intended for use by the path should be encrypted.
- An endpoint should be able to **request signals** from devices on the path.
- An on-path device **should not be able to forge, change, or remove** a signal sent by an endpoint.
- The **endpoint should control signaling** between endpoints and the path, or from one on-path device to another.
- It should be possible for an endpoint to request and receive signals from a **previously unknown on-path device**.
- The mechanism should present no significant surface for amplification attacks.



Applications of the Path Layer

- Transport-Independent On-Path State
 - Latency Measurement
 - Loss and Congestion Measurement
 - Path Trace Accumulation
 - Loss/Latency Tradeoff
 - Path MTU Discovery
-
- Generic mechanism to allow for future extensibility

What happened?

Coordinated Deployment is Difficult

- PLUS requires endpoint protocol changes as well as in-network changes to see the majority of its benefit.
- Lack of first-mover advantage reduces deployment incentive.
- (See section 2.1.2 of RFC 5218, and basically all of RFC 8170).

Advisory Signals are Weird (and all path signals are advisory)

- PLUS was intended to make the notional path layer explicit by separating end-to-end control from on-path signaling.
 - These are blended in TCP. See RFC 8546 for why this is interesting.
- Difficult to make the argument why an encrypted transport would expose signals without clear benefit to the endpoints (see previous slide on coordinated deployment).
- Difficult to make the argument why a network device would trust advisory signals from an endpoint pair it knows has an encrypted side channel to coordinate lying.
 - See also RFC 8558.

Compulsion is Unfriendly

- A concern was raised that an adversarial network operator could compel endpoint-to-path signaling not in the end user's interest.
 - (e.g. "you must expose a cryptographic token proving you viewed some ad, or are authorized by an external authority, to get reasonable QoS.")
- Building a vector into the protocol to allow explicit exposure may add to the risk that such exposure will become mandatory; this runs counter to the philosophy of the Internet as a network of networks.

Experiments are Scary

- SPUD was defined as a concept and an experimental implementation of that concept.
 - It wasn't a fully defined protocol, and not a WG-forming BoF.
 - People expected a fully-defined protocol, and a WG-forming BoF.
 - The mismatch was a key ingredient in the failure of SPUD and its rebirth as PLUS.

Comparing APN and PLUS

Deployment incentives

- PLUS envisioned an end-to-end path layer, while APN provides path-layer like functionality with traffic encapsulation in restricted domains (a la IOAM).
 - Controller and APN domain are managed by a single entity, but...
 - ...questions about first mover advantage for application and access devices remain: why build APN into your app/platform?
-

Signal trustworthiness

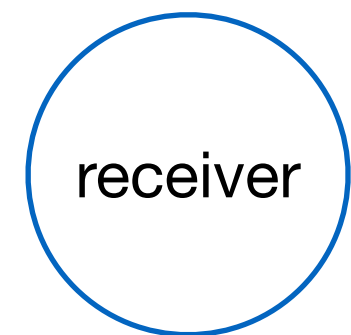
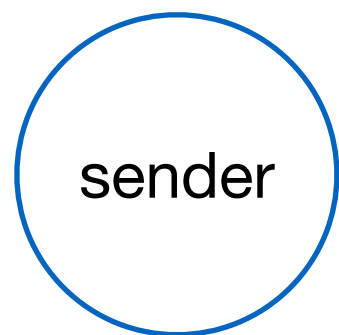
- PLUS signaling is network-focused but application- and protocol-agnostic (LoLa, reordering tolerance, bandwidth expectation, measurability).
- APN signaling has application IDs with more complicated semantics in addition to implementation- and protocol-specific network-focused signals
 - The advisory signaling questions apply: why should the path should trust the app? Why should the app should tell the truth?



Backup: PLUS in Detail (also from the 2017 CNSM talk)

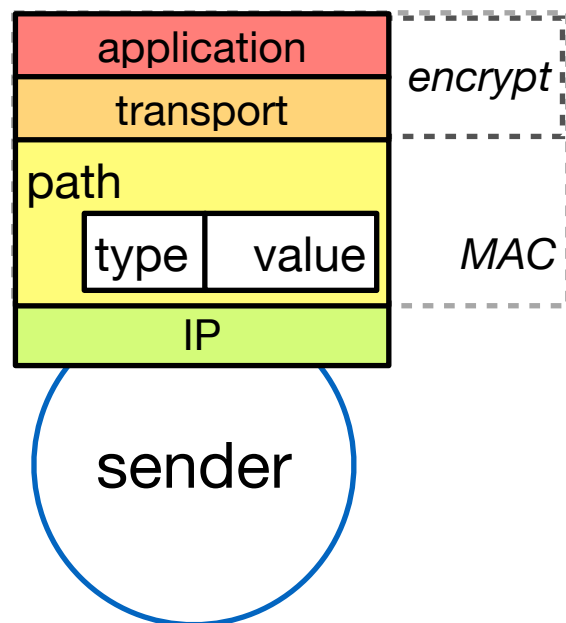


Sender to Path Signaling



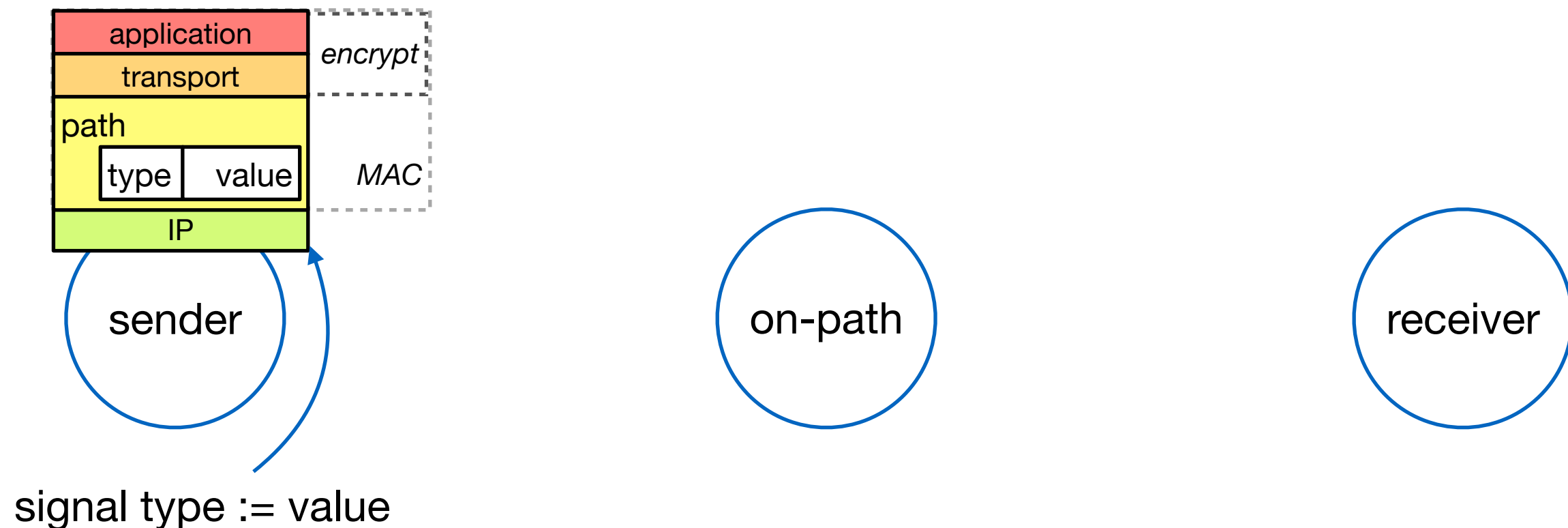


Sender to Path Signaling



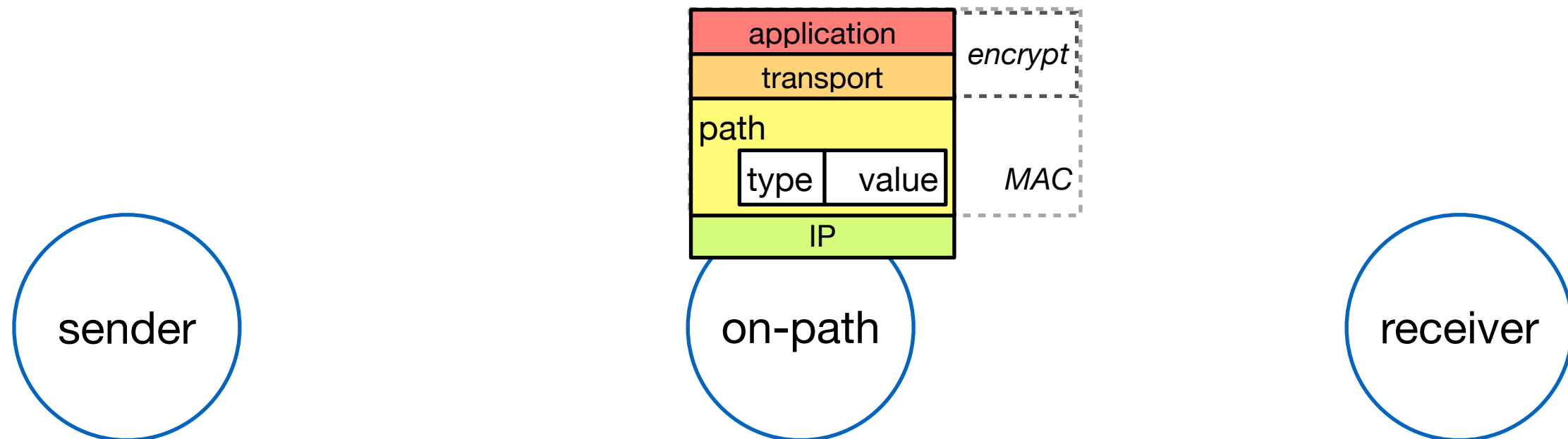


Sender to Path Signaling



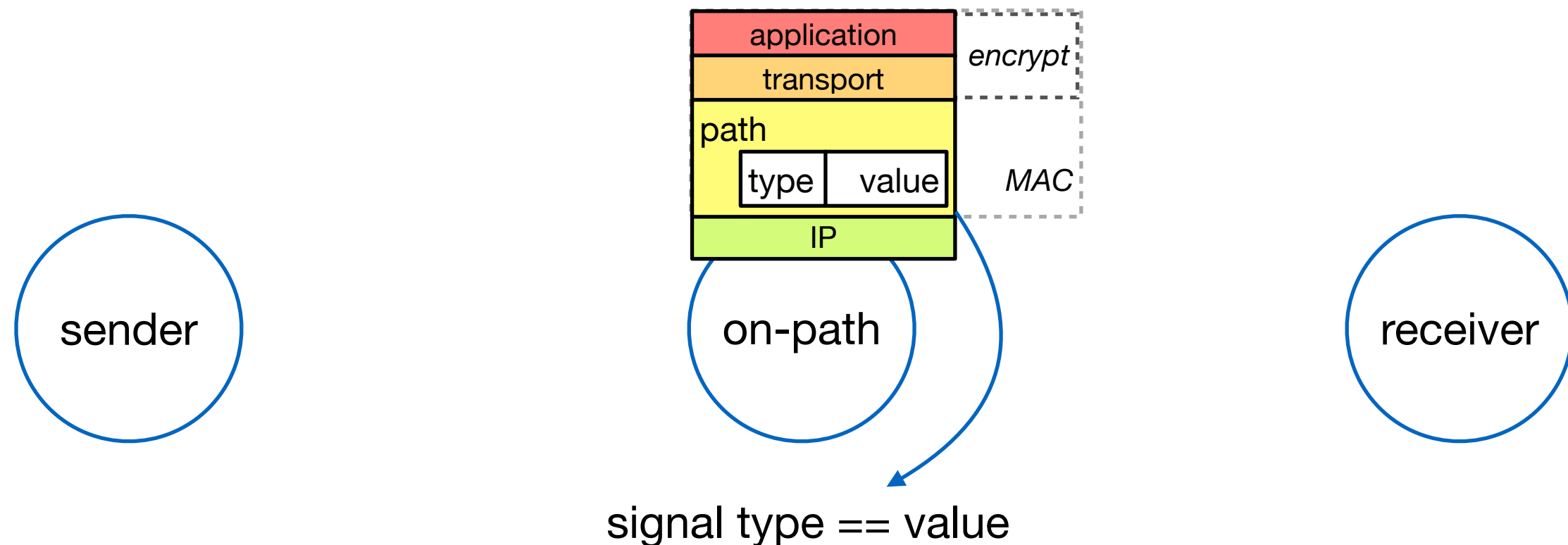


Sender to Path Signaling



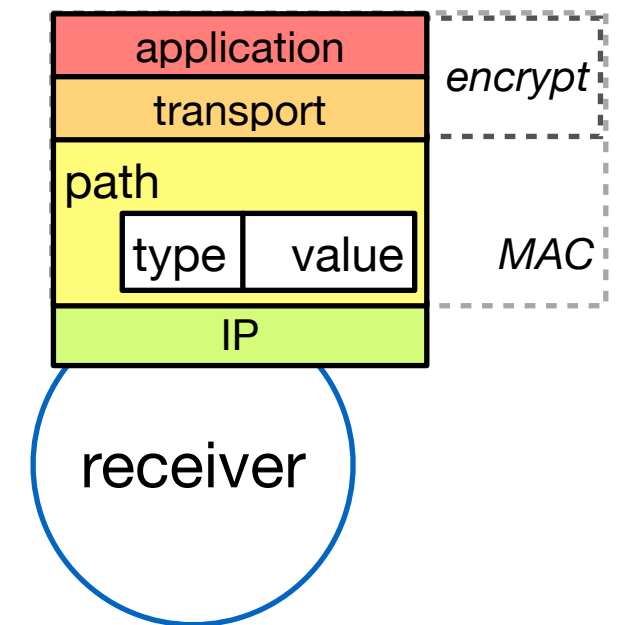
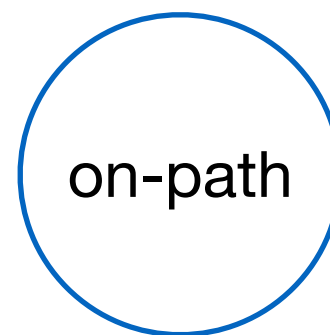
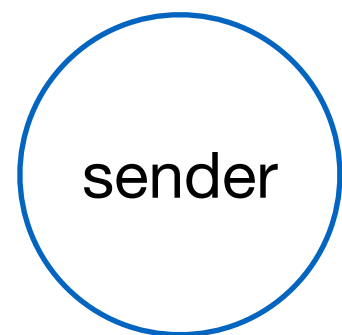


Sender to Path Signaling



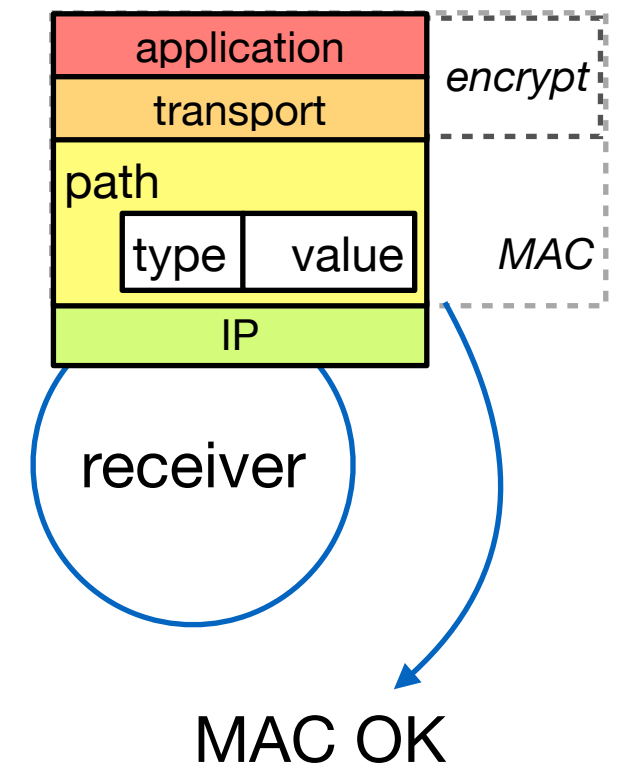
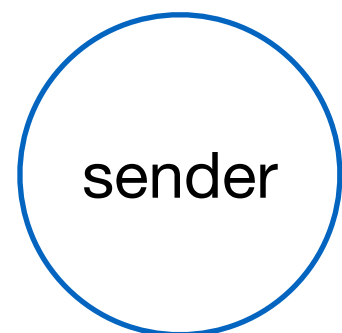


Sender to Path Signaling



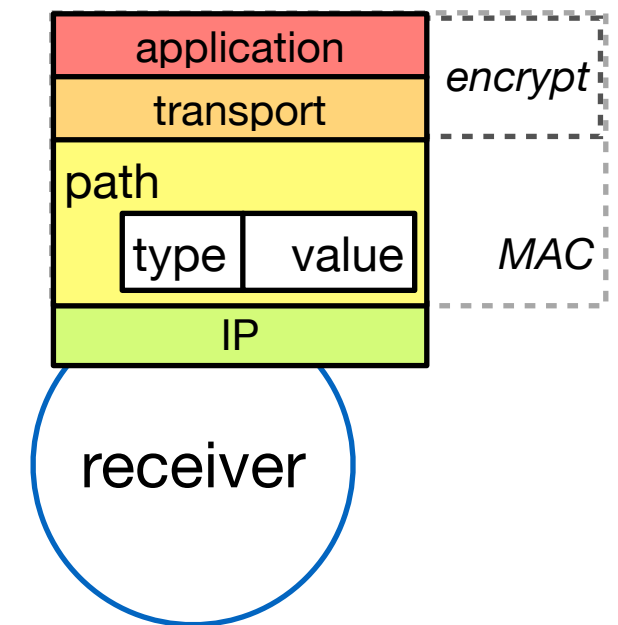
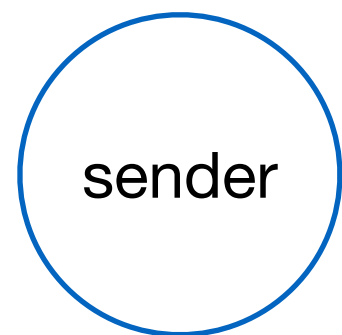


Sender to Path Signaling





Sender to Path Signaling





Basic PLUS Header

31	16	15	0
0	UDP Source Port		UDP Destination Port
4	UDP Length		UDP Checksum
8	PLUS Magic 0xd8007ff		L R S 0
12	Connection and Association Token (CAT)		
20	Packet Serial Number (PSN)		
24	Packet Serial Echo (PSE)		
28	Encrypted		



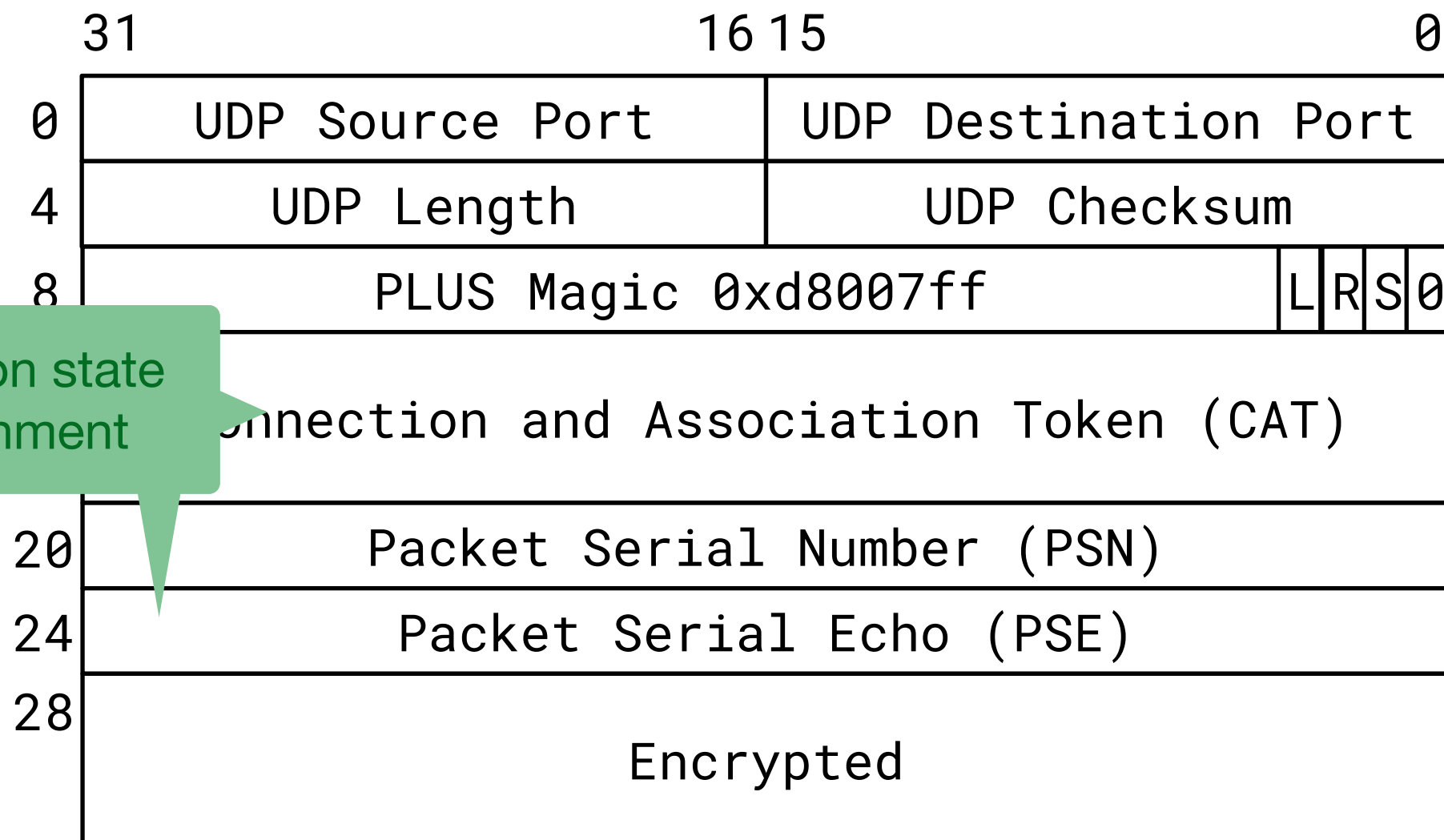
Basic PLUS Header

Recognize PLUS packets on path

		16	15					0			
		Source Port				UDP Destination Port					
4		UDP Length				UDP Checksum					
8		PLUS Magic 0xd8007ff						L	R	S	0
12		Connection and Association Token (CAT)									
20		Packet Serial Number (PSN)									
24		Packet Serial Echo (PSE)									
28		Encrypted									



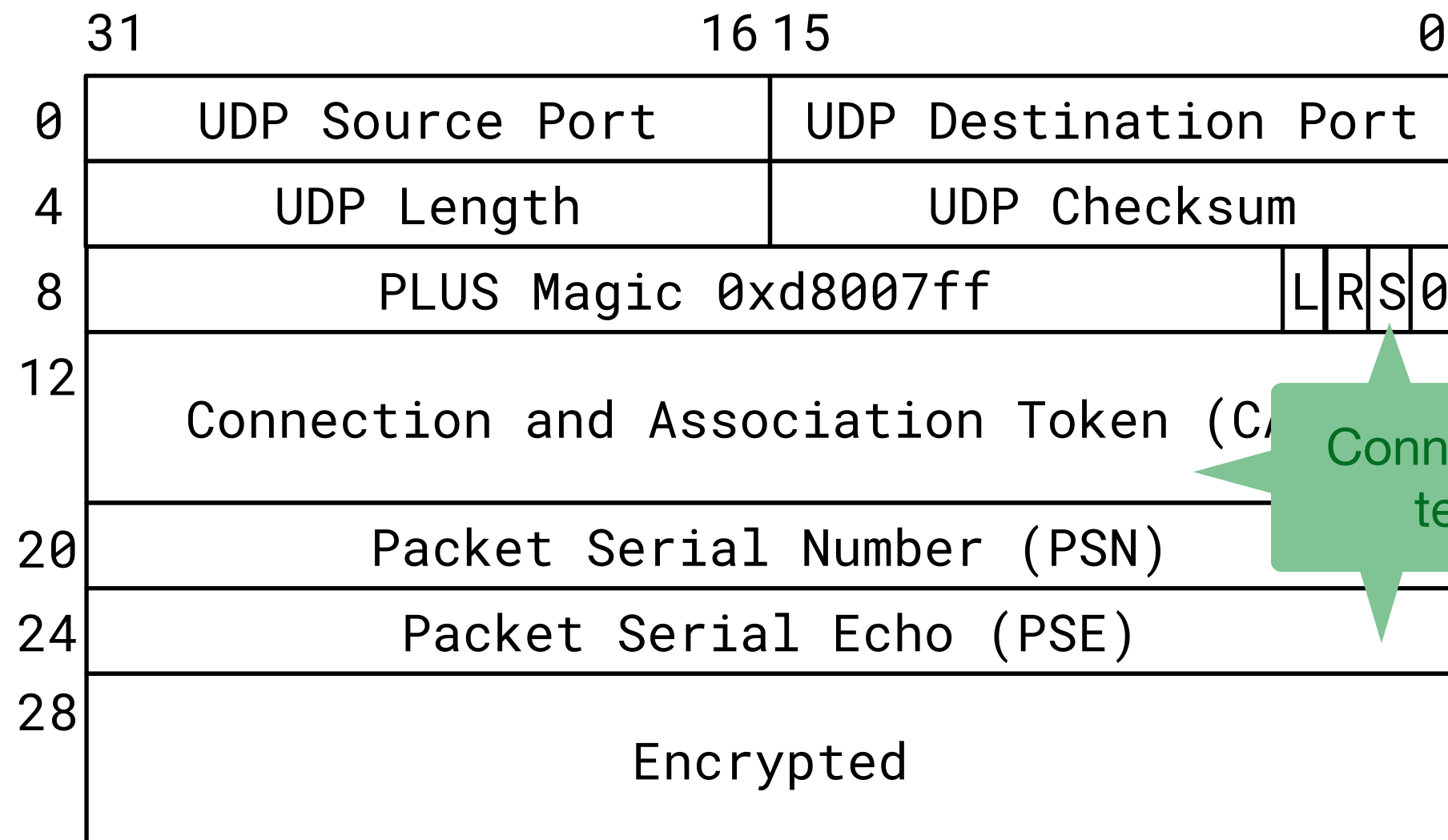
Basic PLUS Header



Connection state
establishment



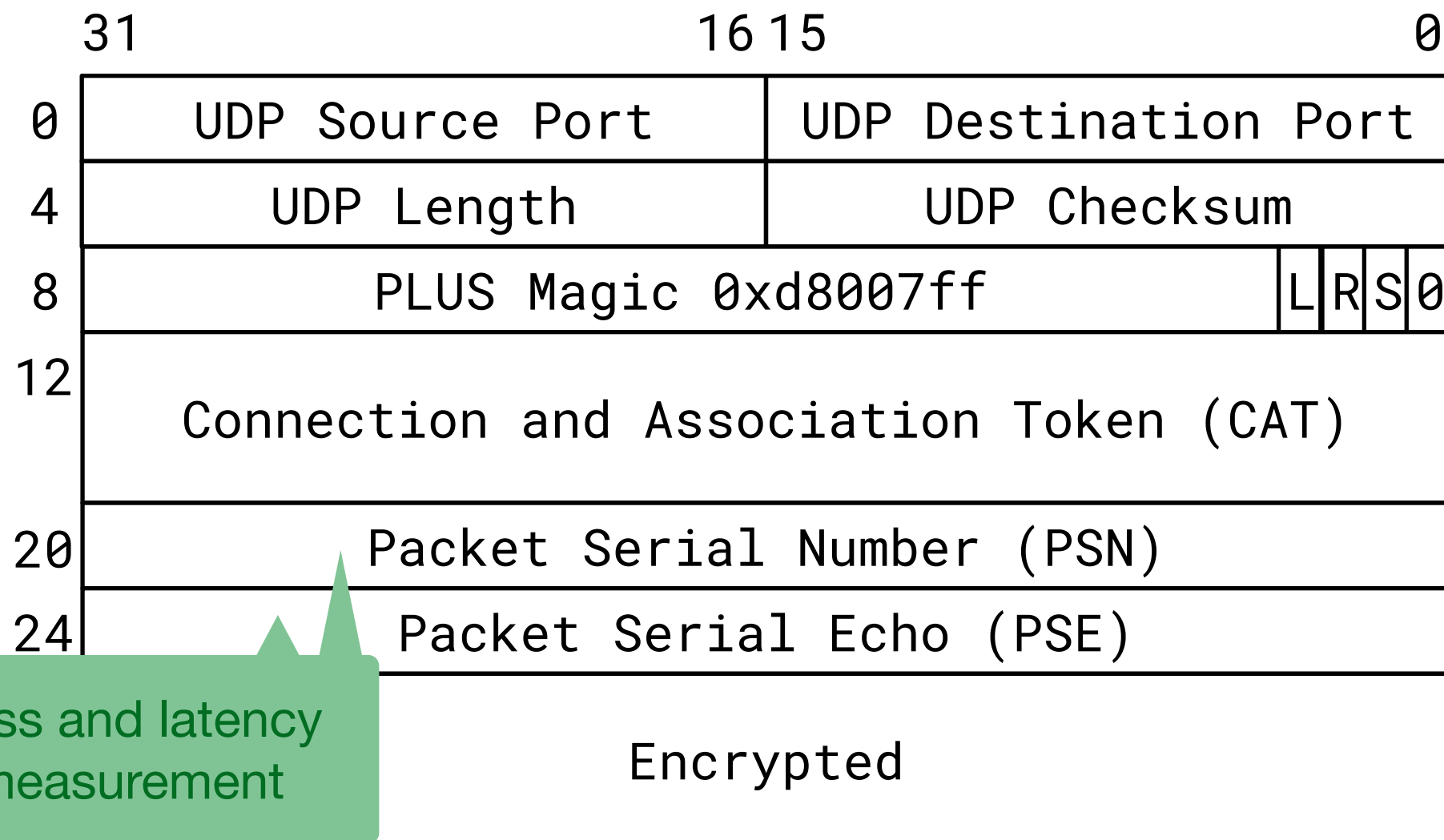
Basic PLUS Header



Connection state
teardown

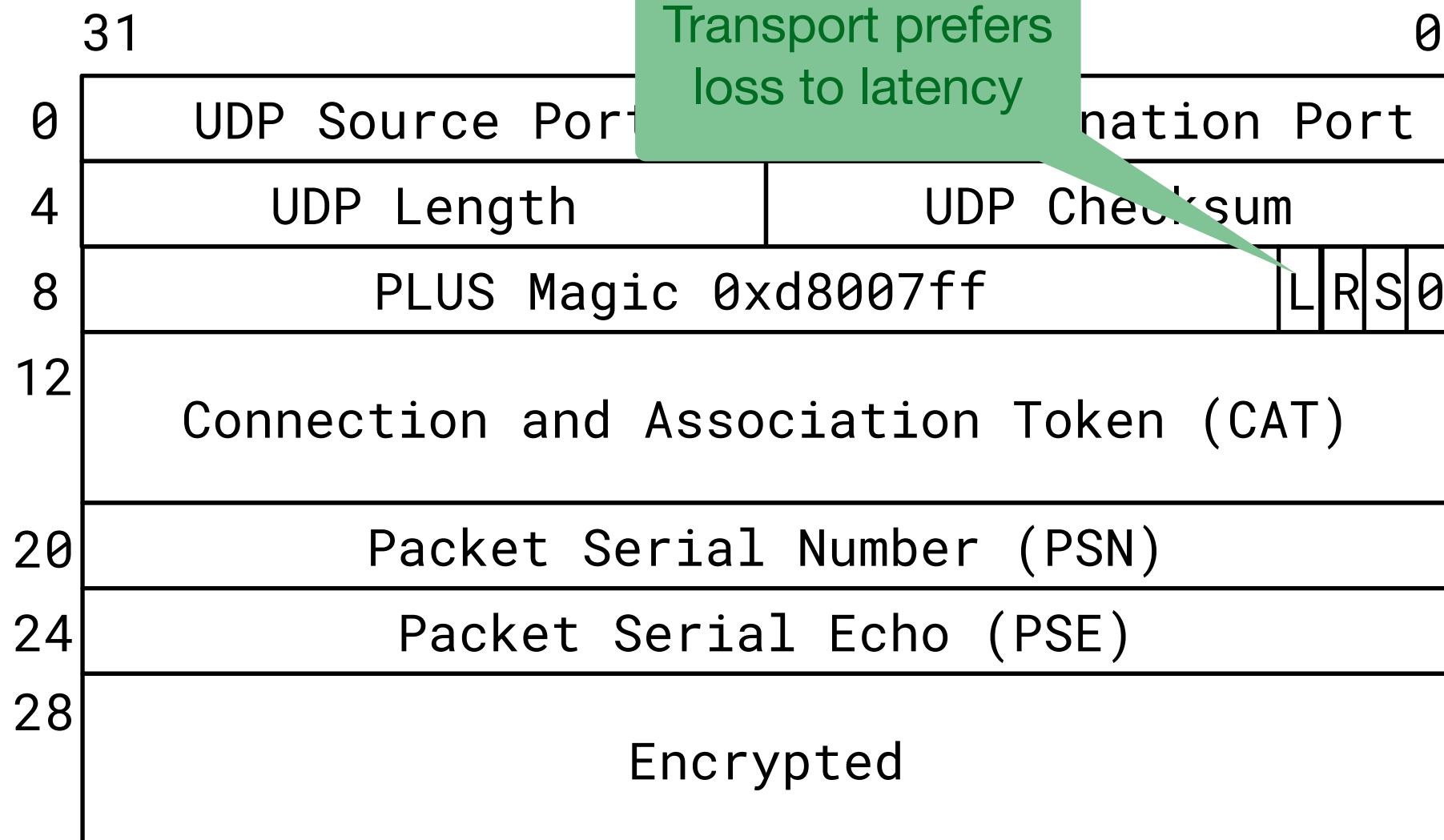


Basic PLUS Header



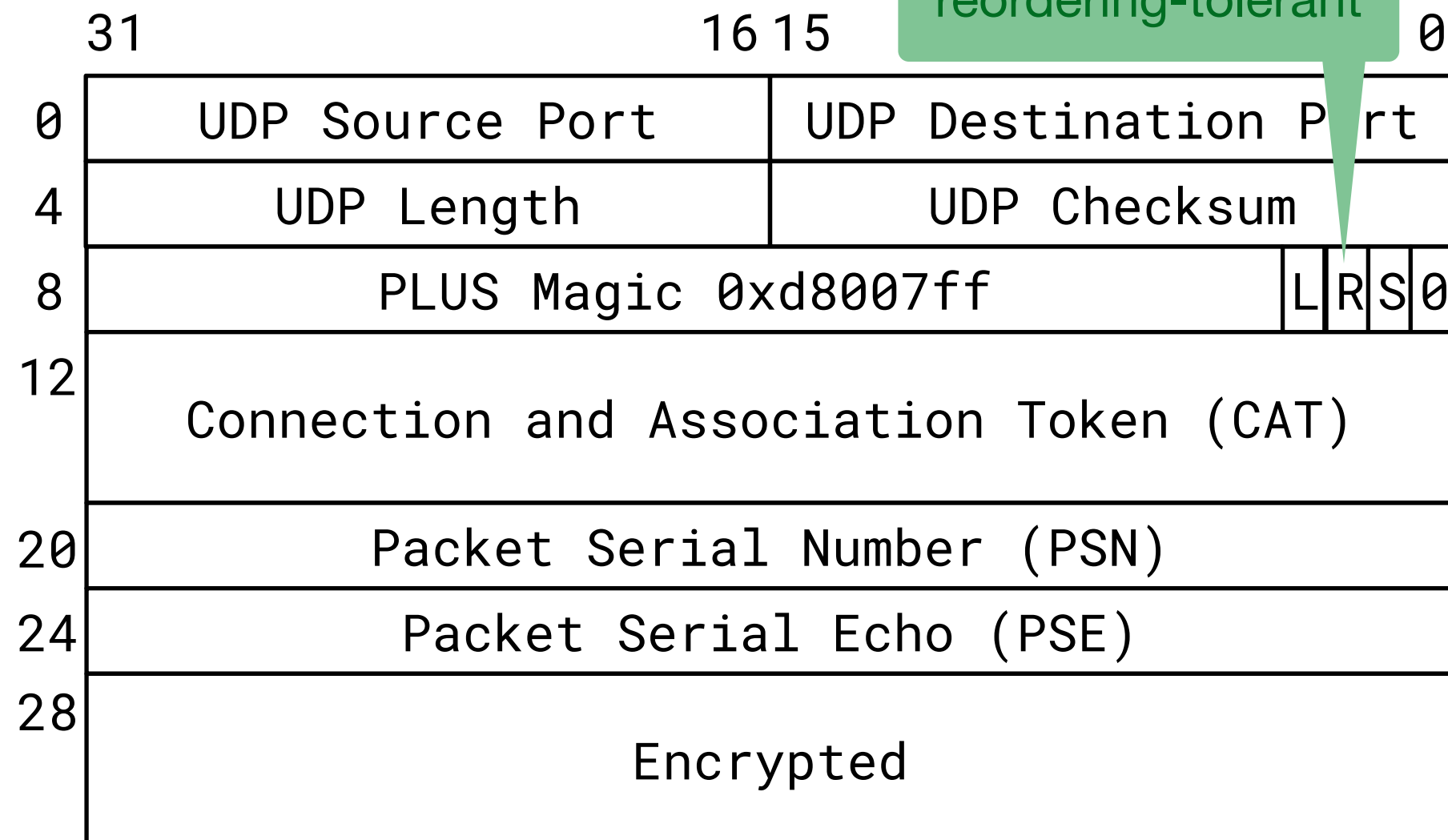


Basic PLUS Header





Basic PLUS Header



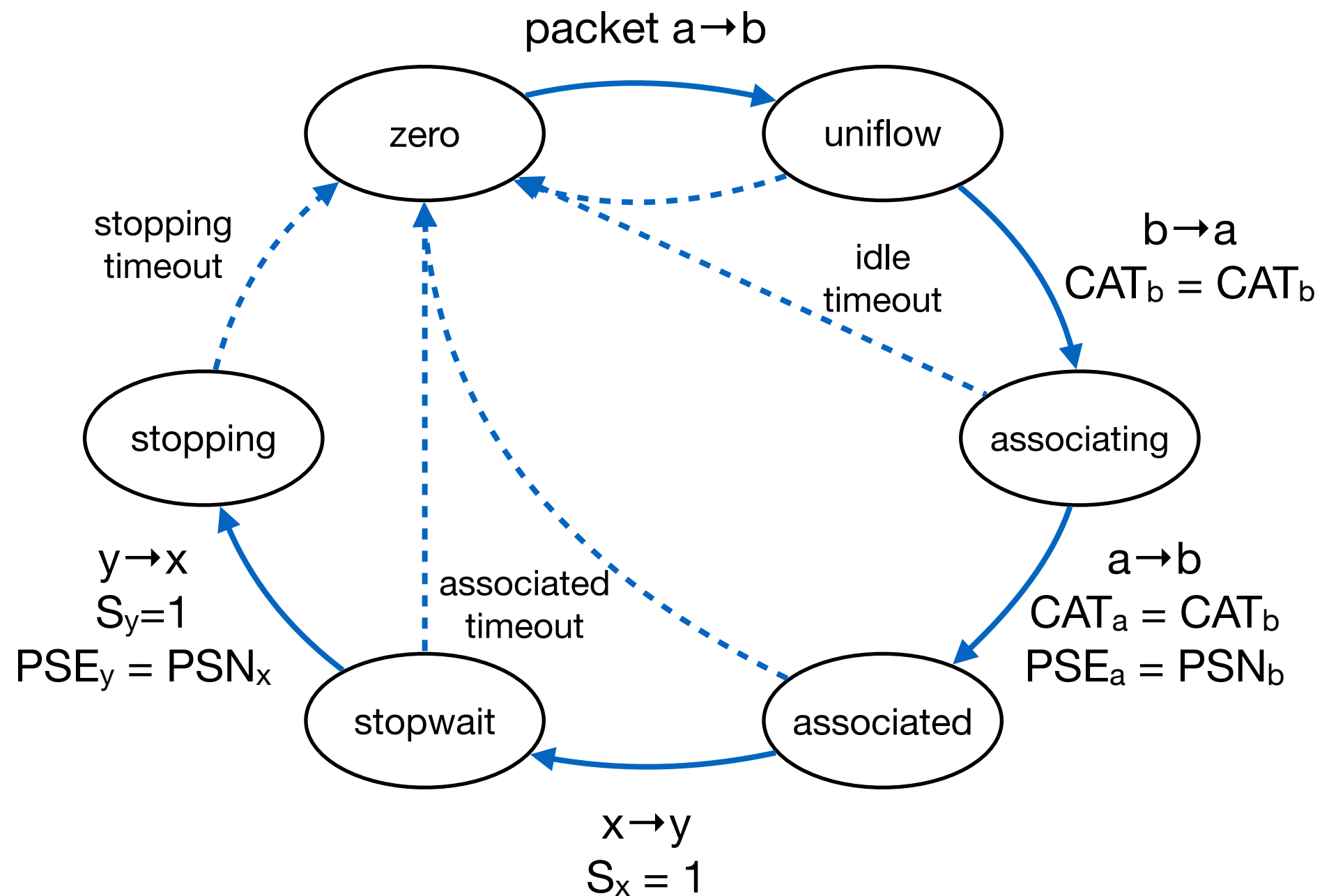


Basic PLUS Header

	31	16	15	0
0	UDP Source Port		UDP Destination Port	
4	UDP Length		UDP Checksum	
8	PLUS Magic 0xd8007ff			L R S 0
12	Connection and Association Token (CAT)			
20	Packet Serial Number (PSN)			
24	Packet Serial Echo (PSE)			
28	Encrypted			

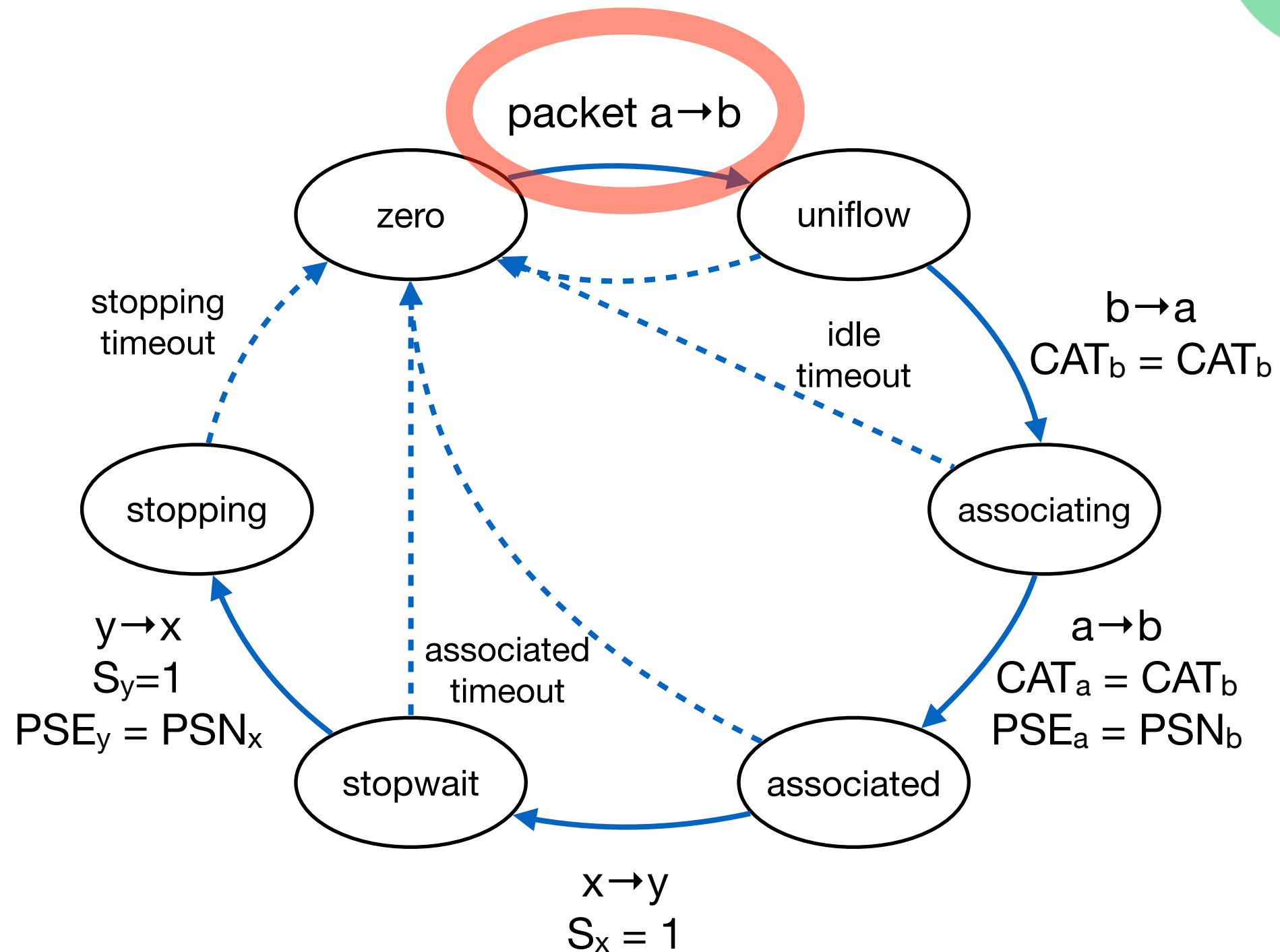


Transport-Independent On-Path State



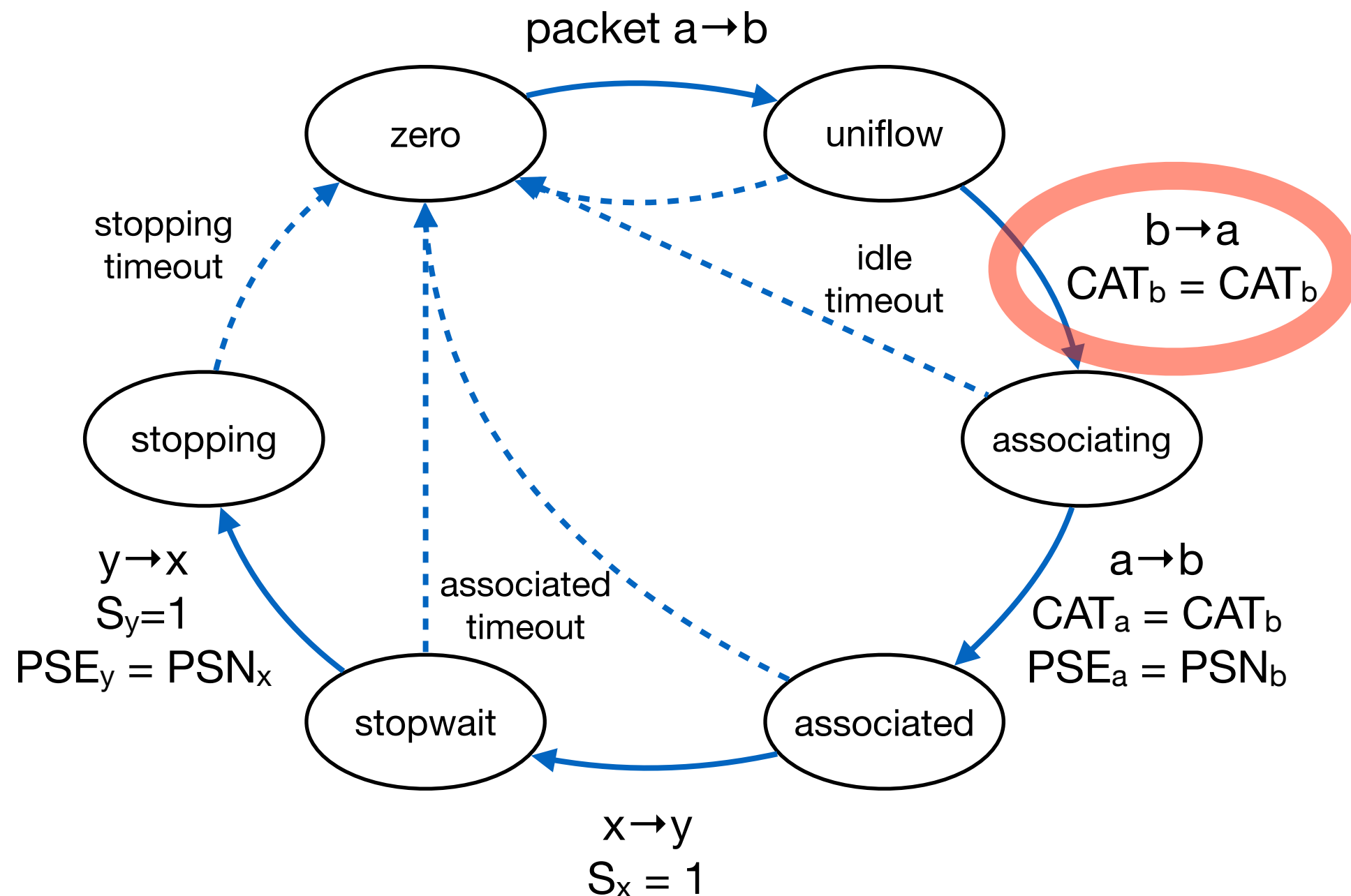


Transport-Independent On-Path State



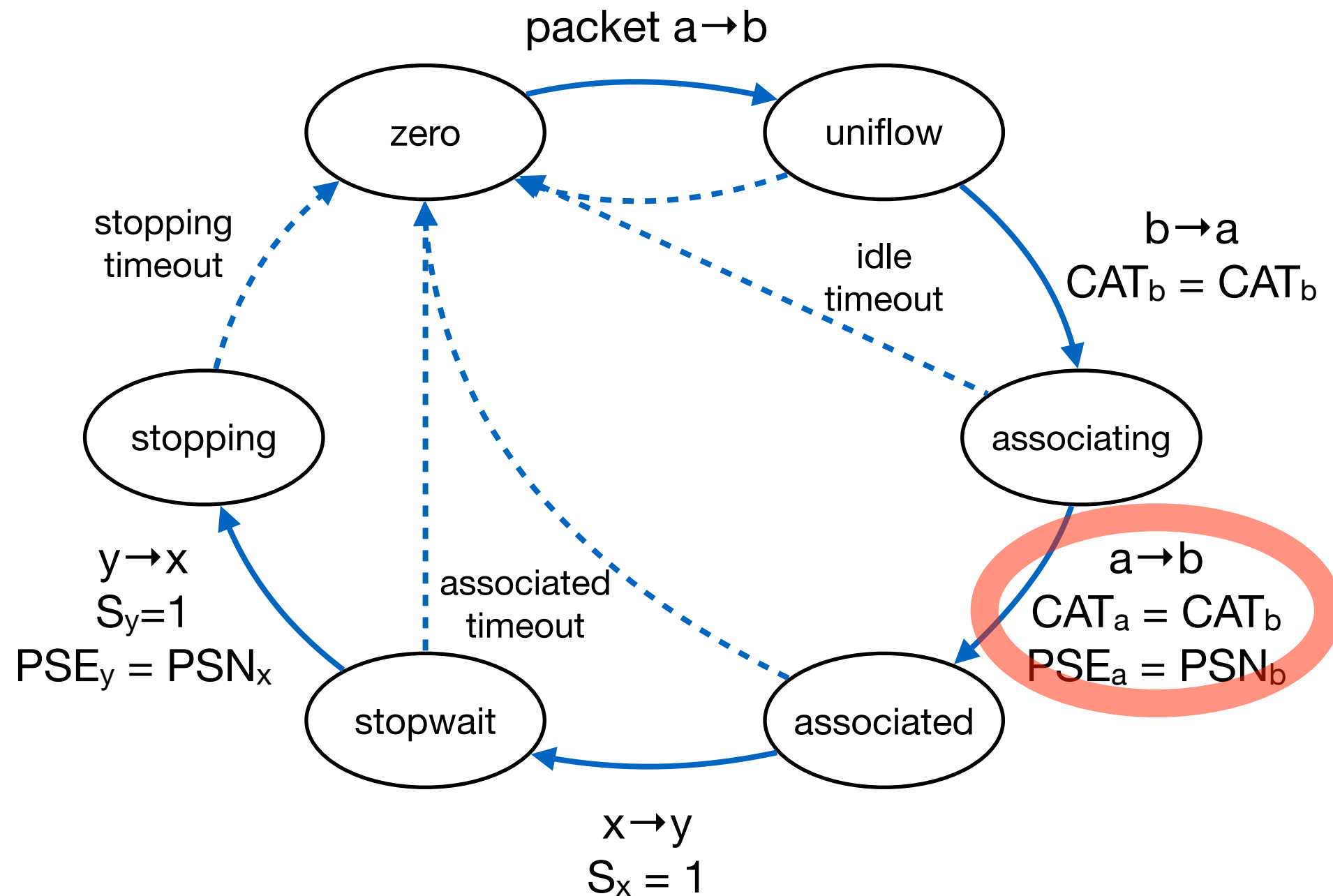


Transport-Independent On-Path State



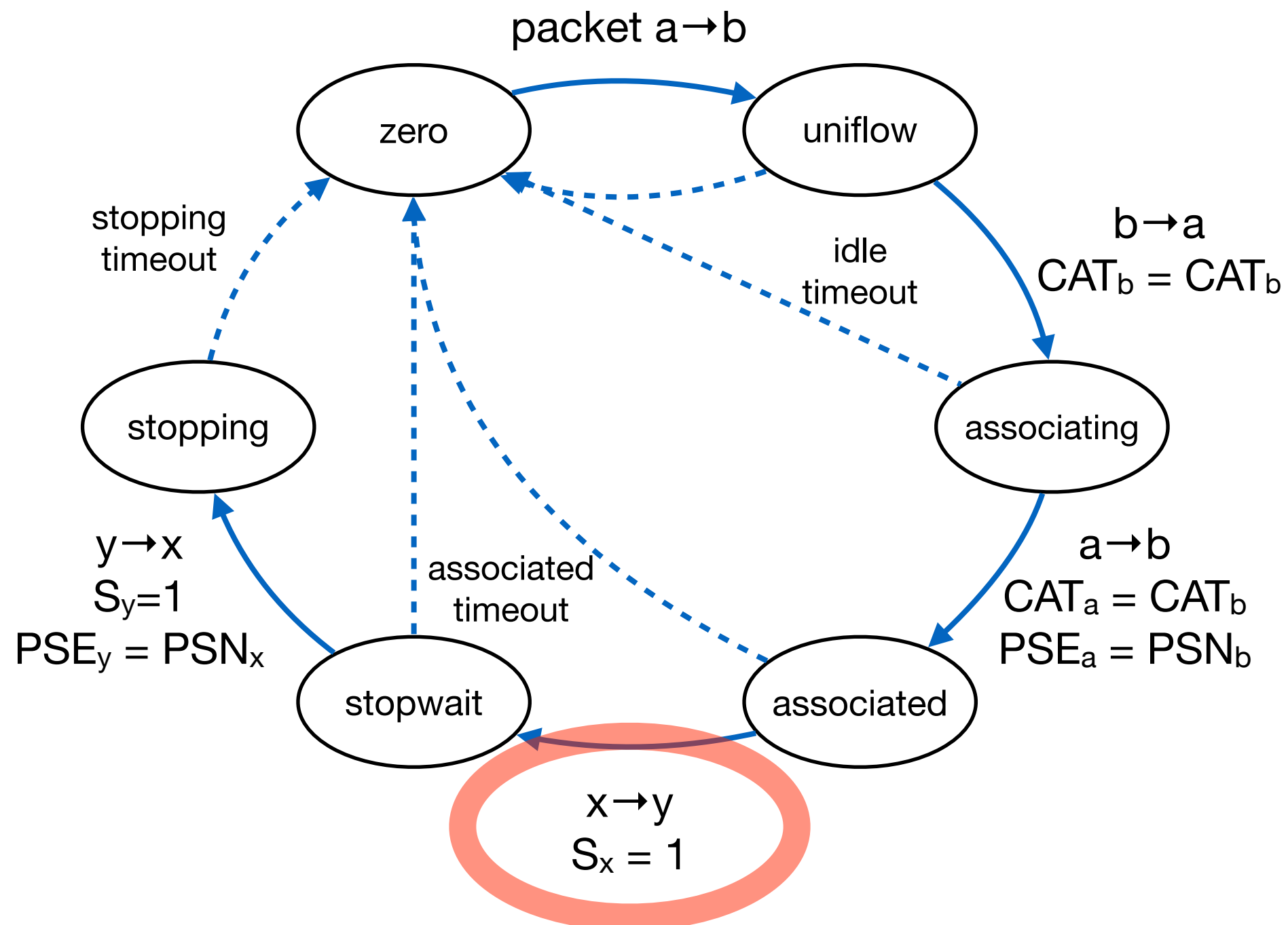


Transport-Independent On-Path State



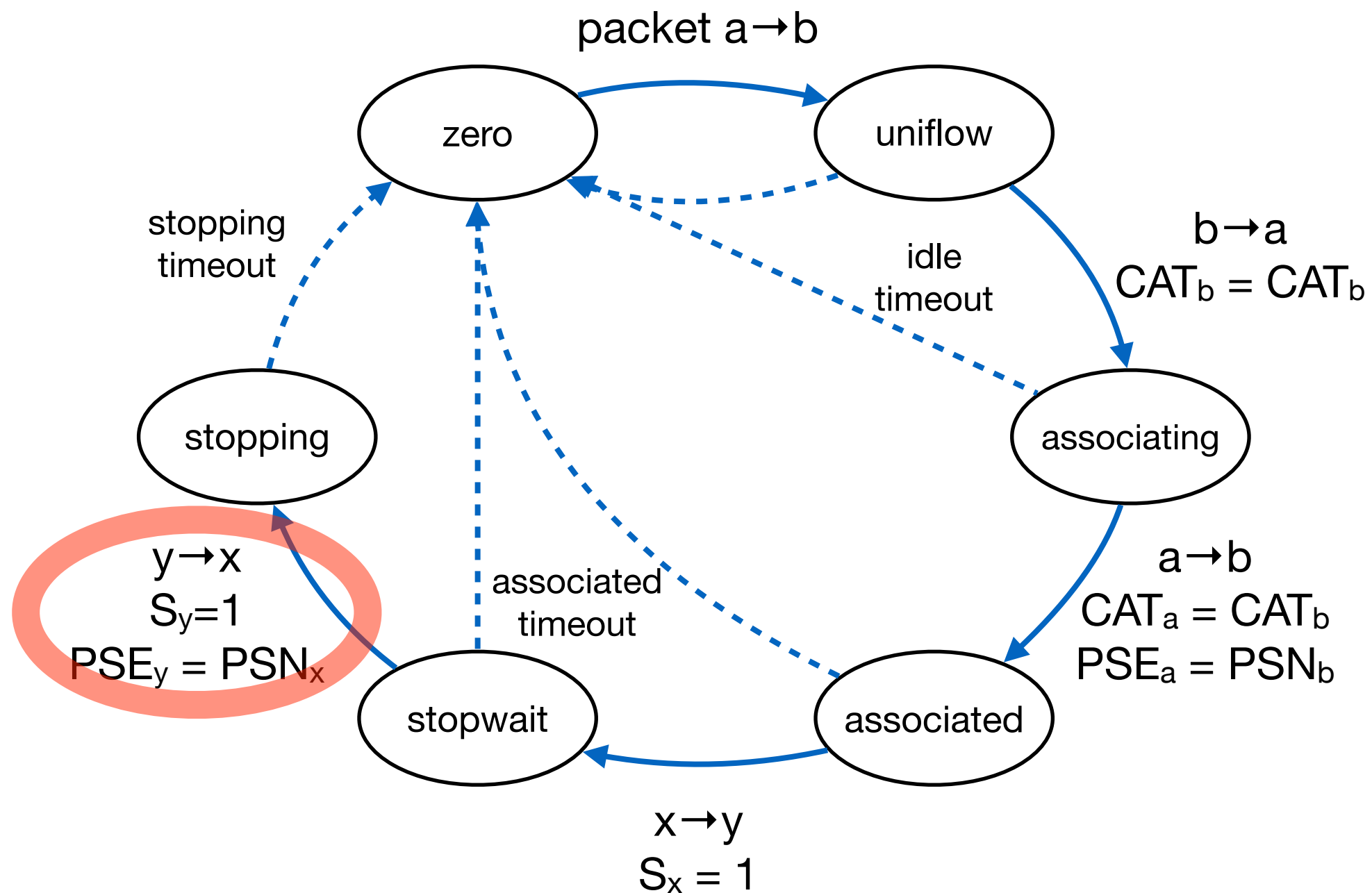


Transport-Independent On-Path State



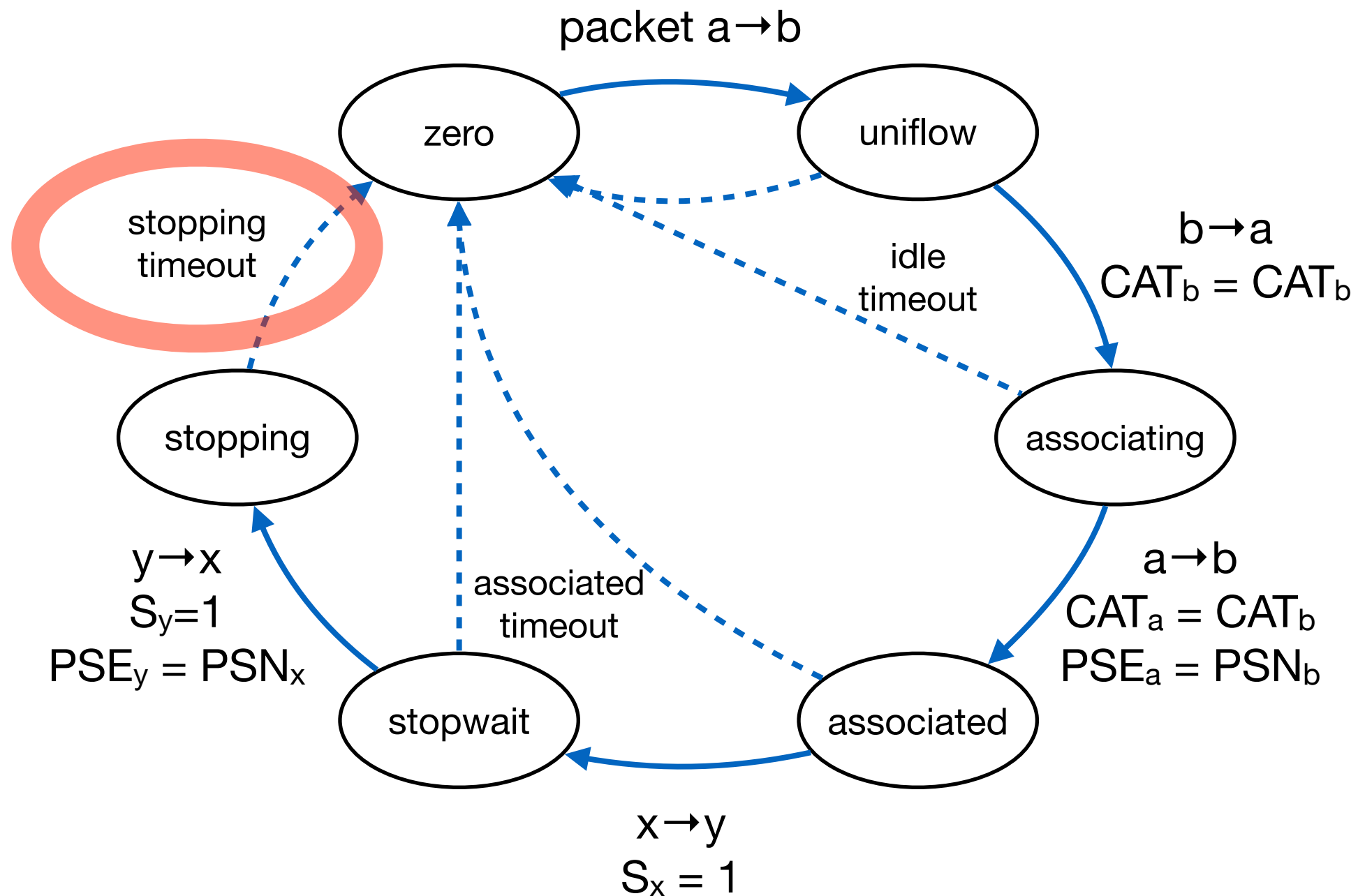


Transport-Independent On-Path State



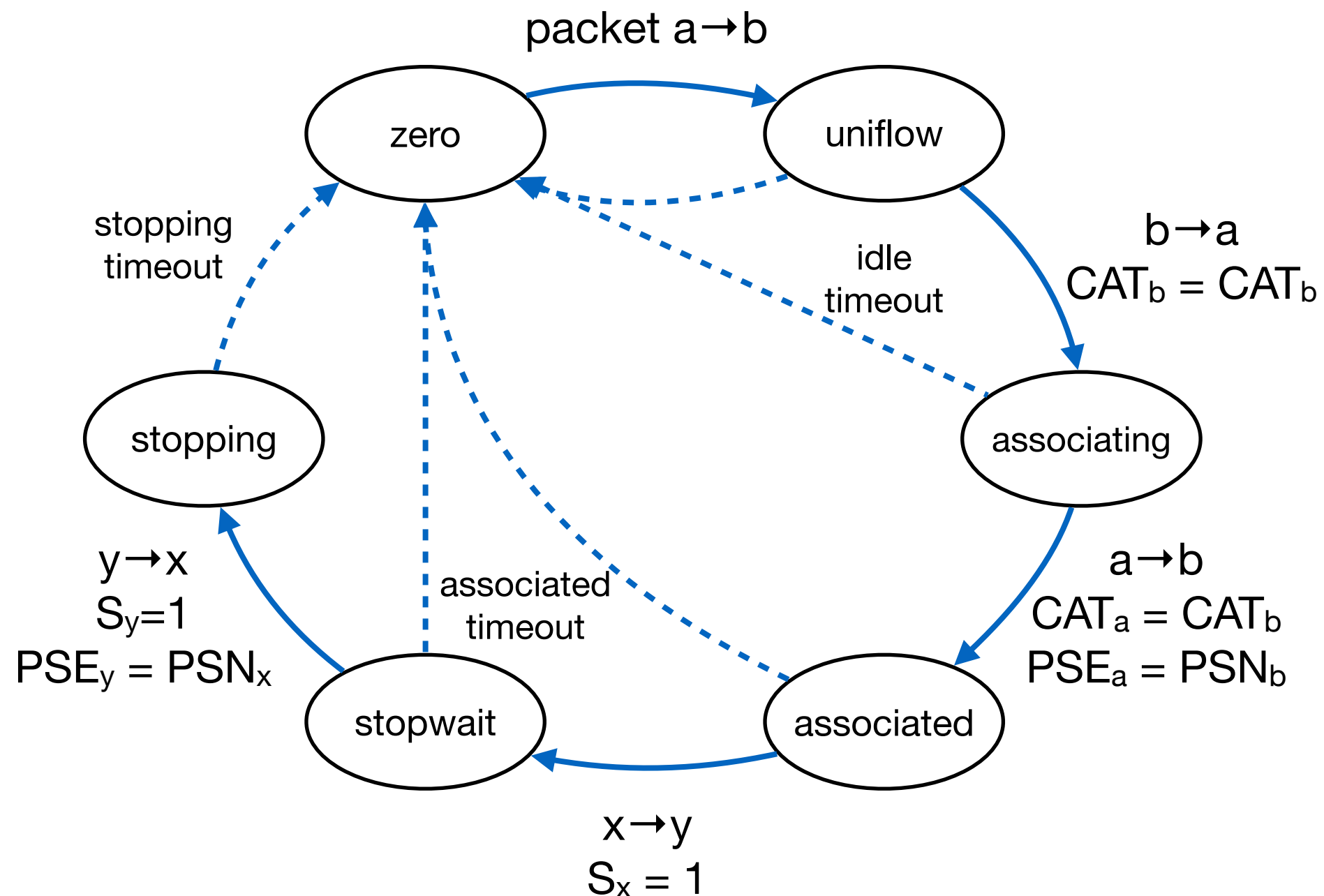


Transport-Independent On-Path State



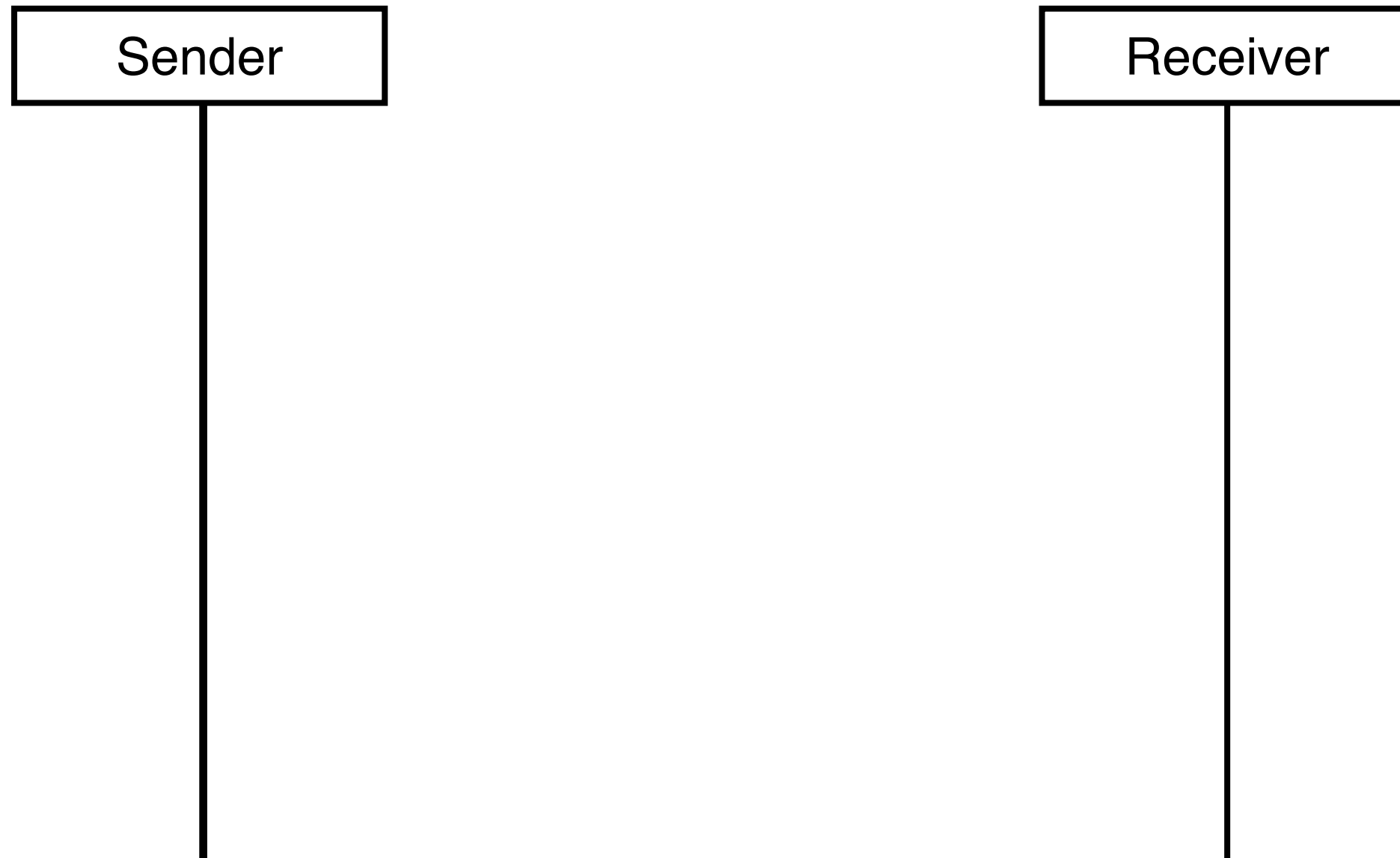


Transport-Independent On-Path State





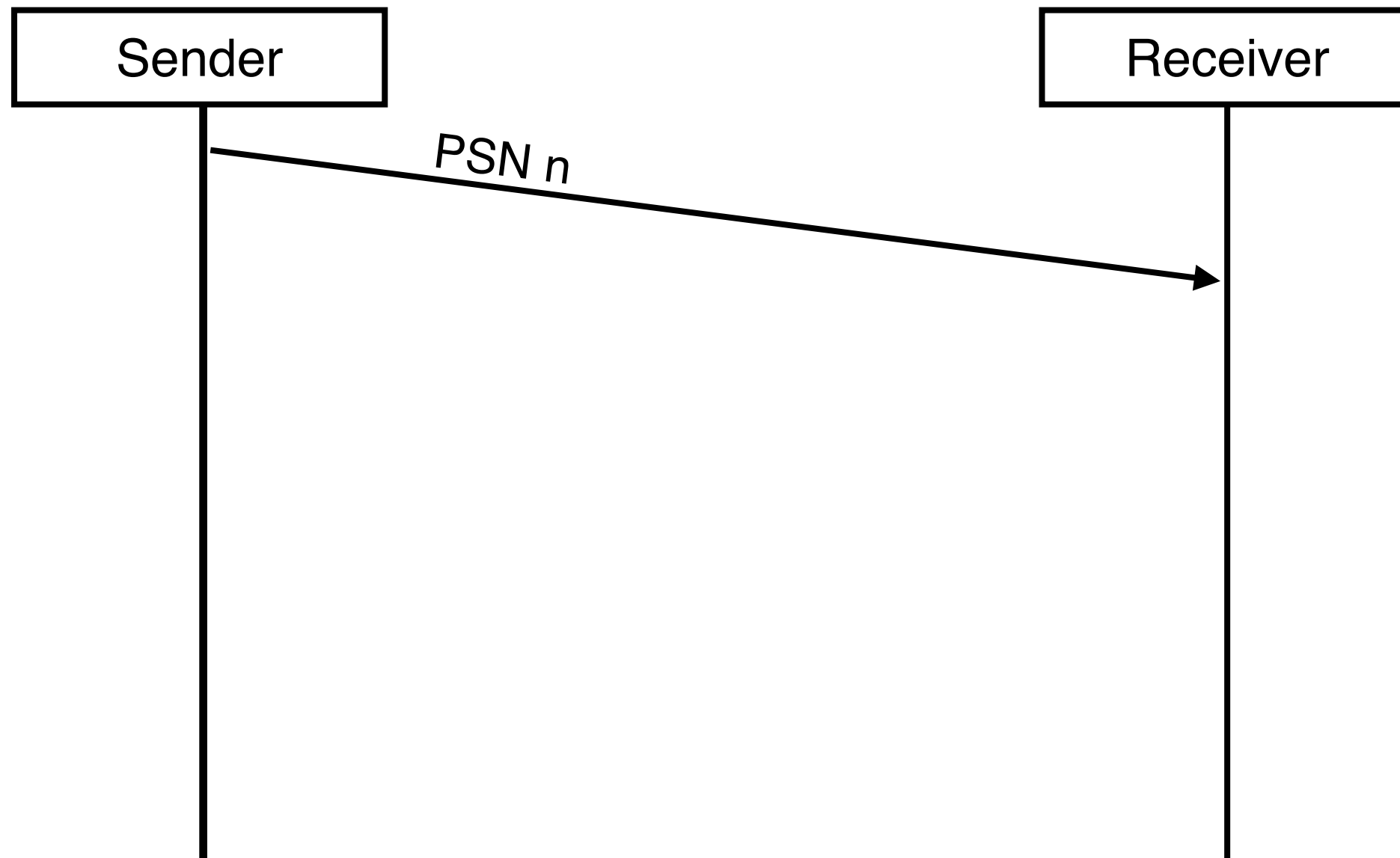
Latency Measurement



- PSN/PSE are explicit measurement signals replacing TCP SEQ/ACK + TSOPT



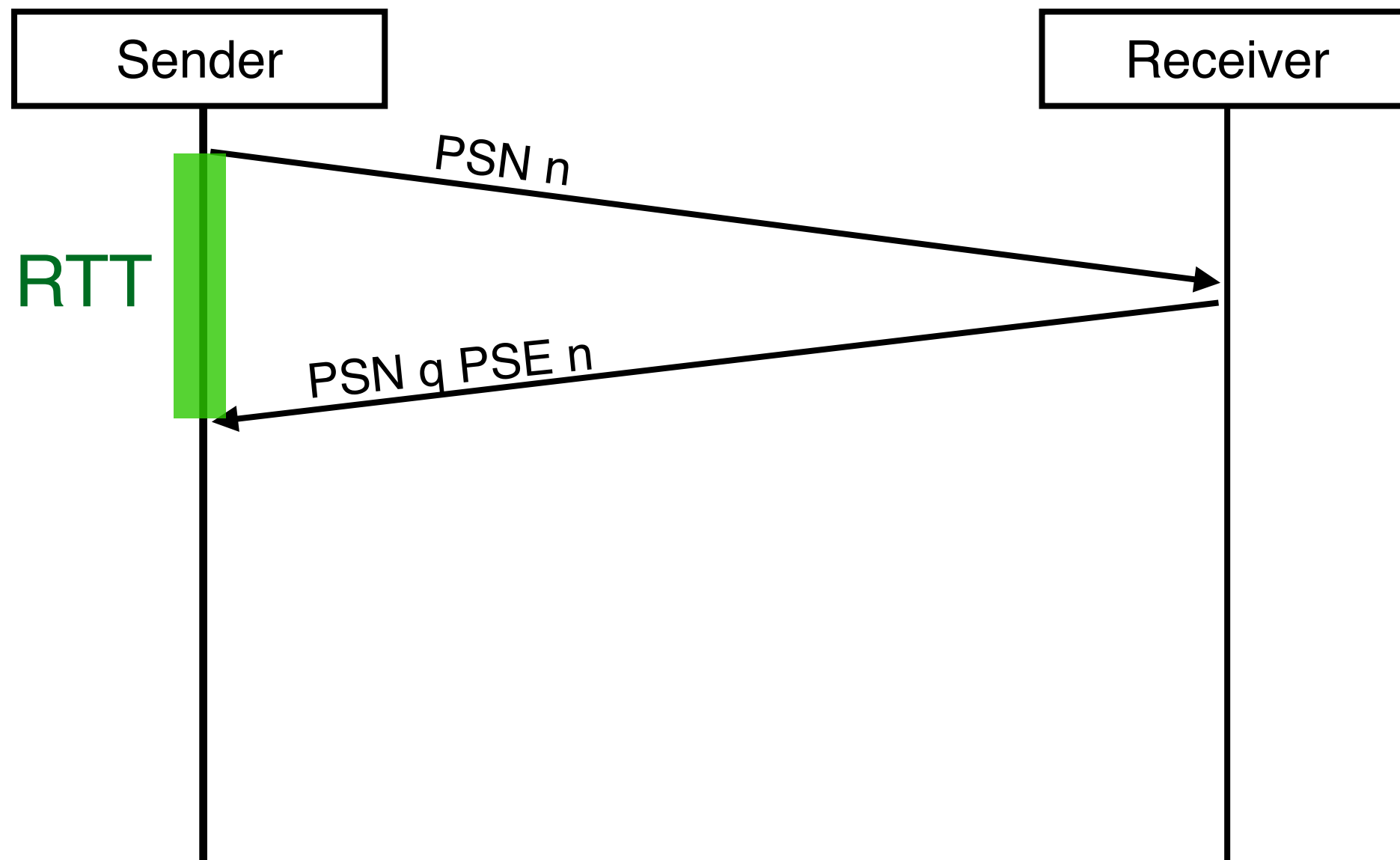
Latency Measurement



- PSN/PSE are explicit measurement signals replacing TCP SEQ/ACK + TSOPT



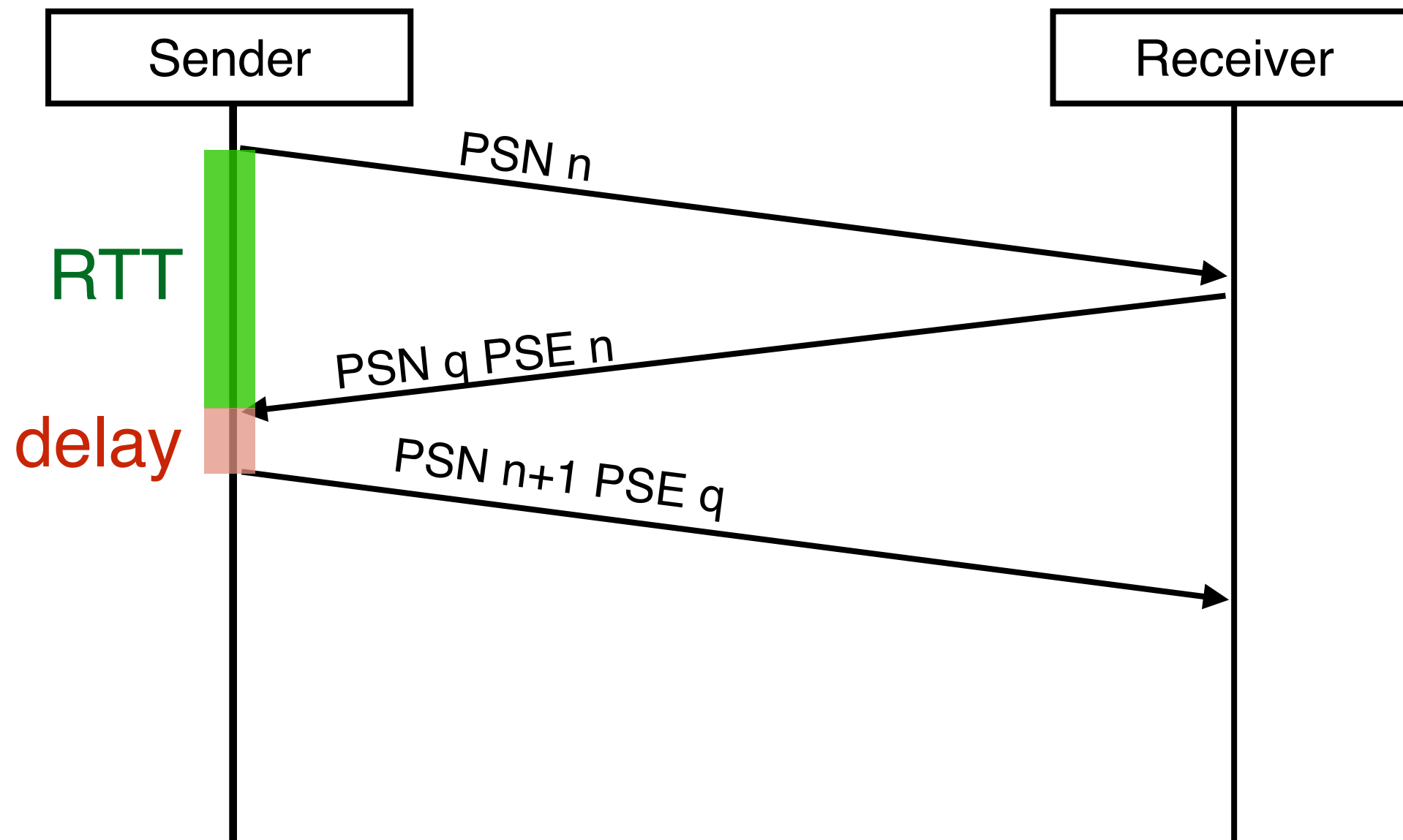
Latency Measurement



- PSN/PSE are explicit measurement signals replacing TCP SEQ/ACK + TSOPT



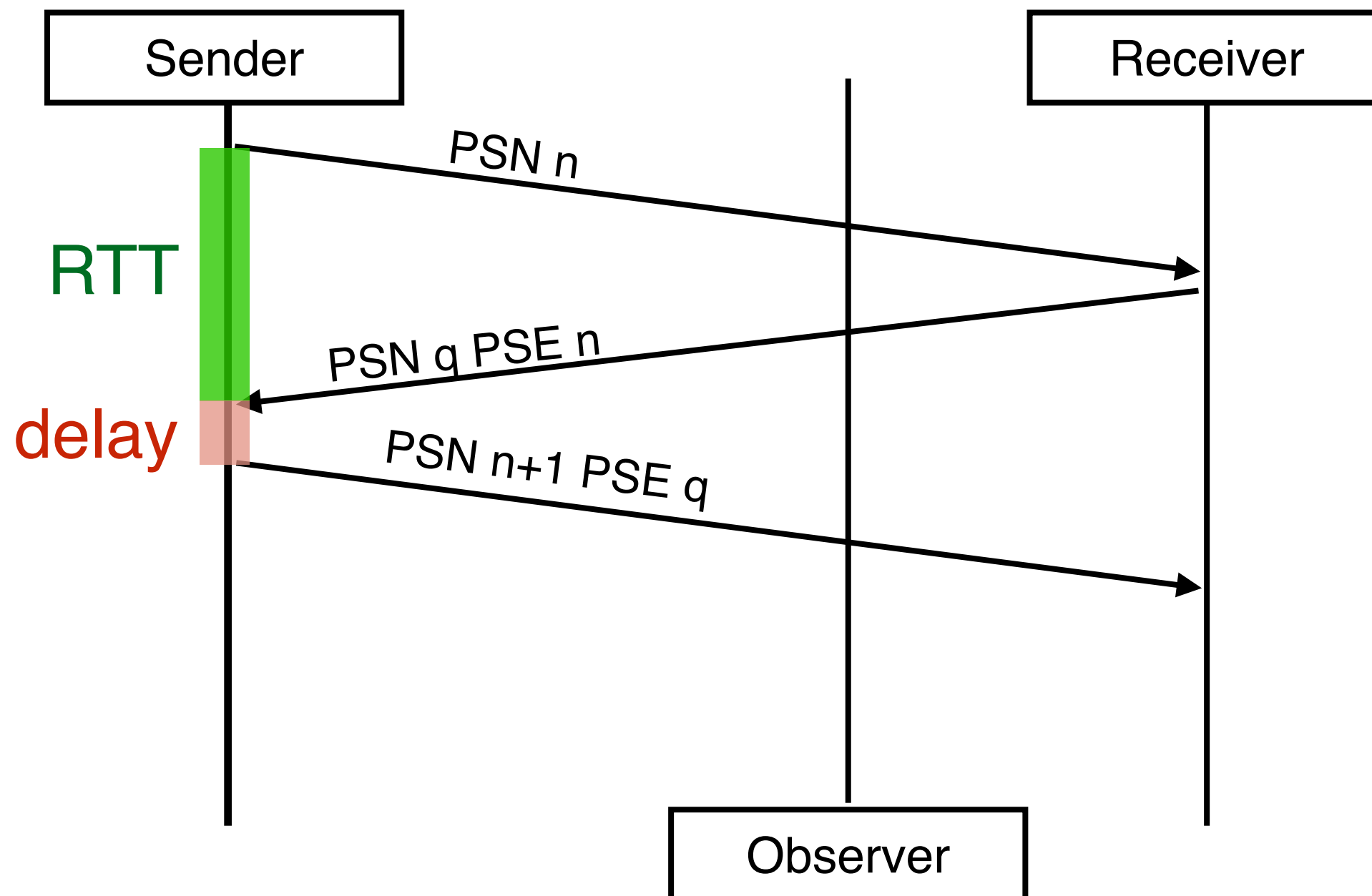
Latency Measurement



- PSN/PSE are explicit measurement signals replacing TCP SEQ/ACK + TSOPT



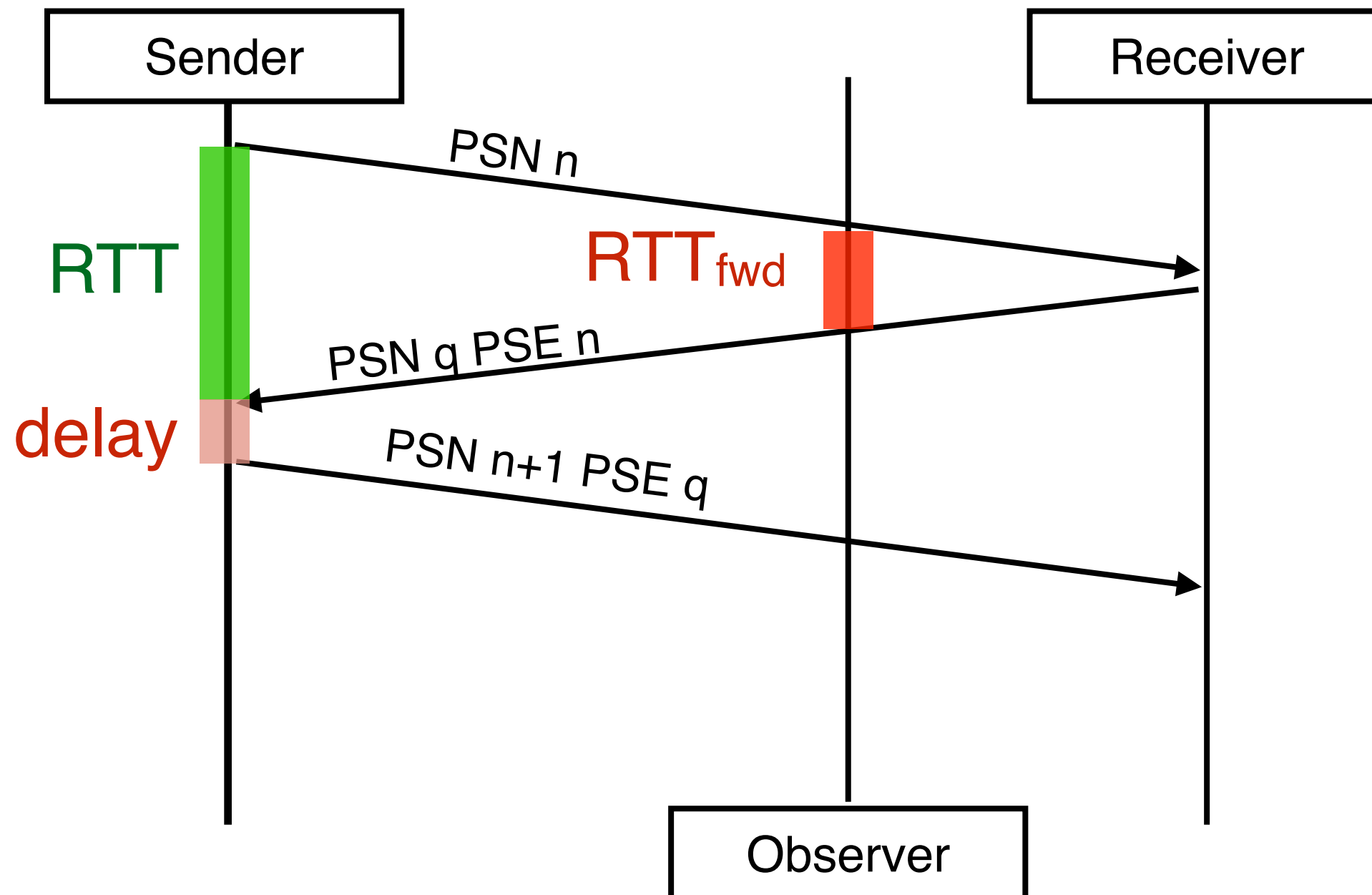
Latency Measurement



- PSN/PSE are explicit measurement signals replacing TCP SEQ/ACK + TSOPT



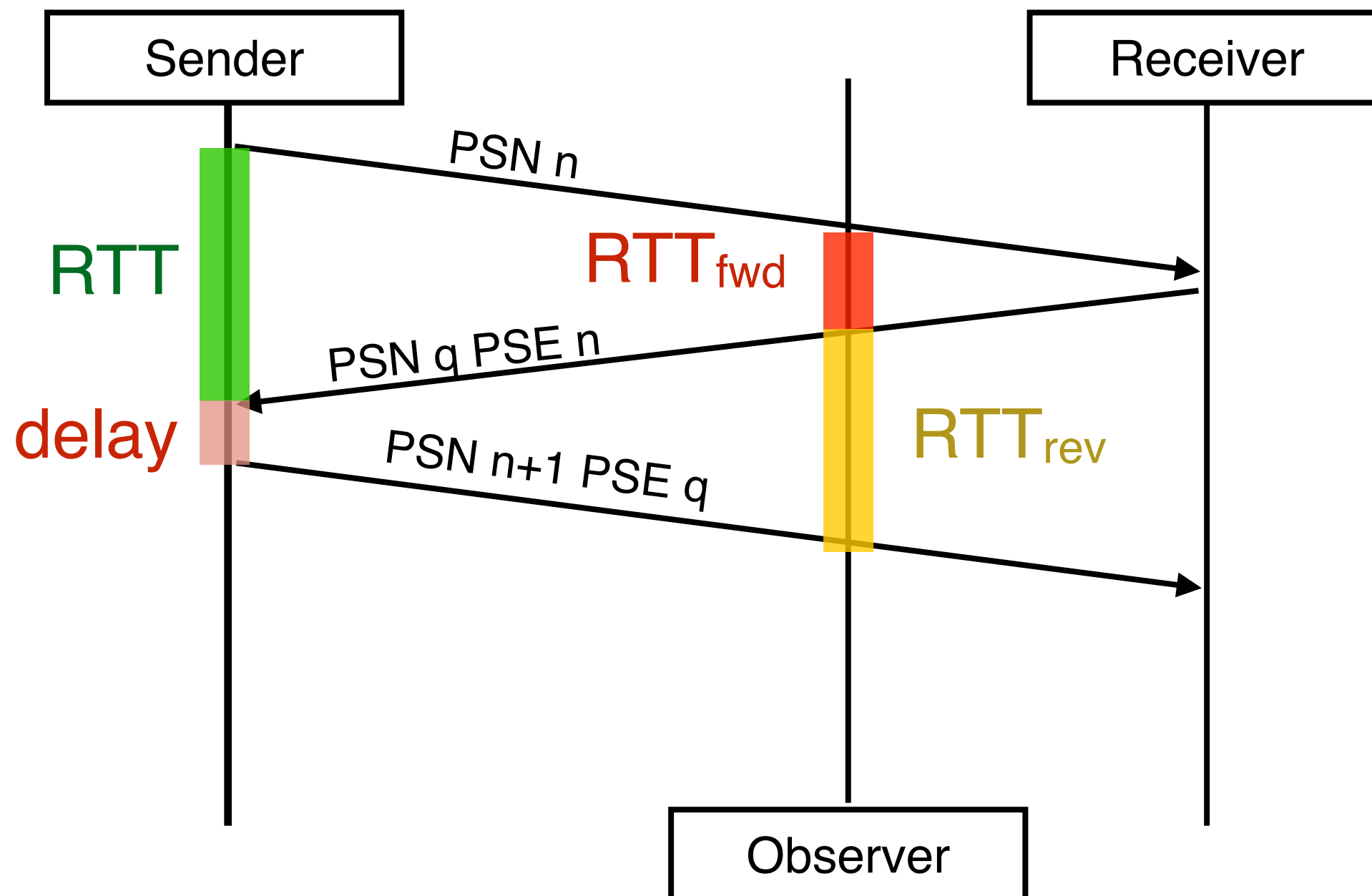
Latency Measurement



- PSN/PSE are explicit measurement signals replacing TCP SEQ/ACK + TSOPT



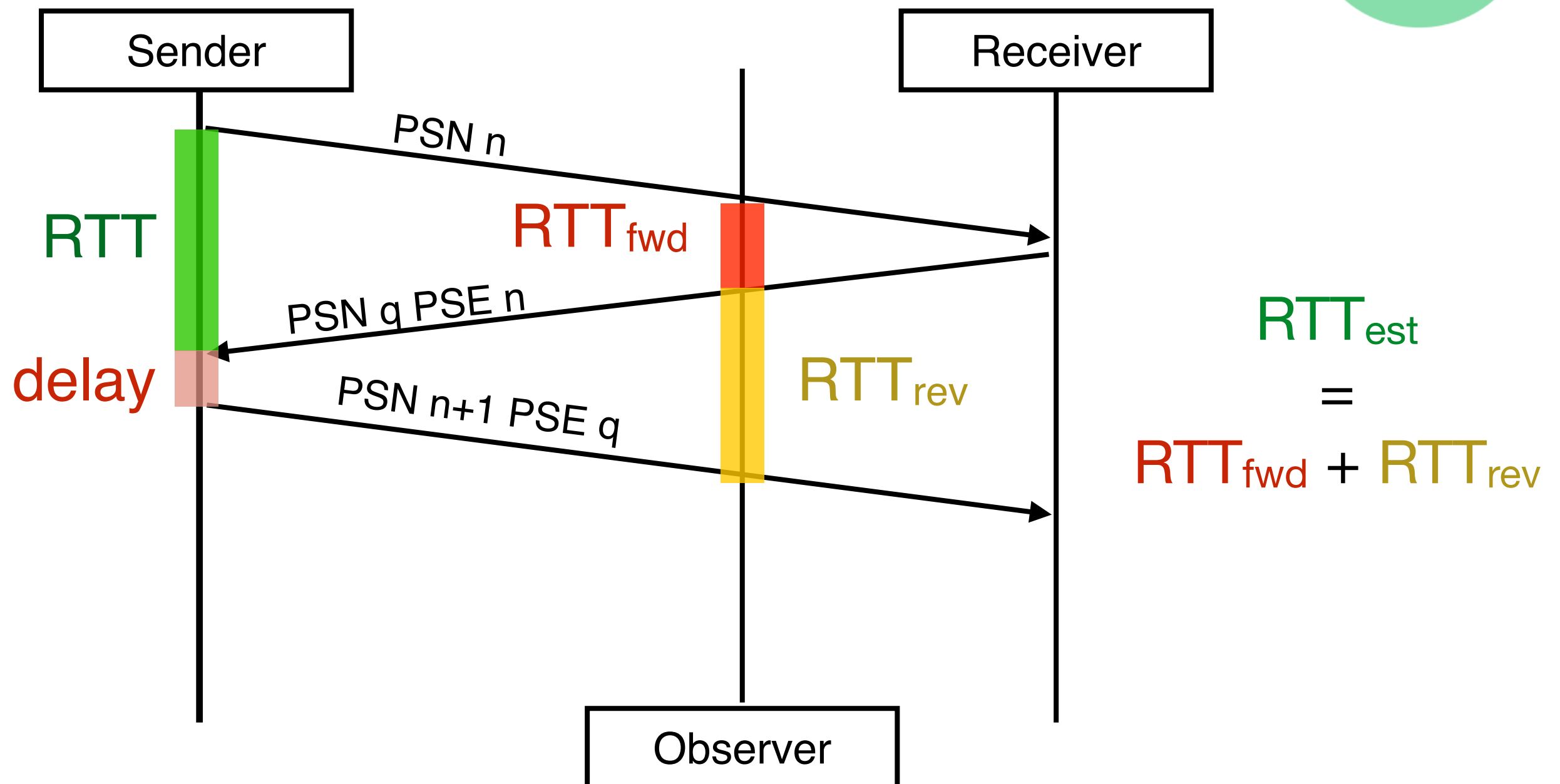
Latency Measurement



- PSN/PSE are explicit measurement signals replacing TCP SEQ/ACK + TSOPT



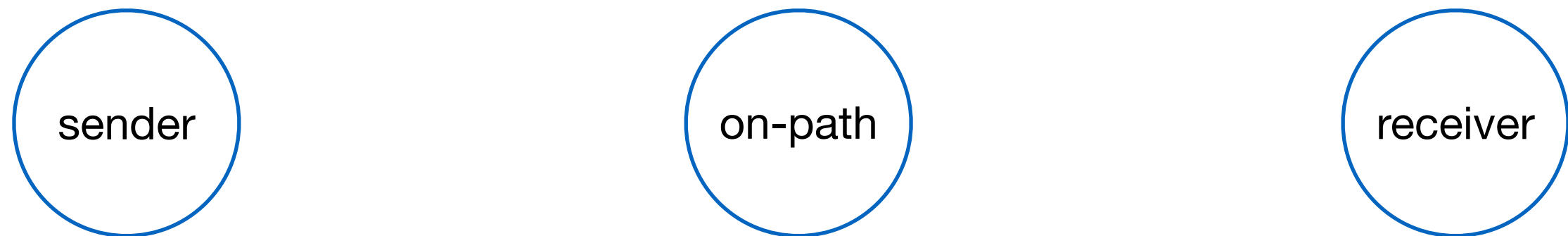
Latency Measurement



- PSN/PSE are explicit measurement signals replacing TCP SEQ/ACK + TSOPT

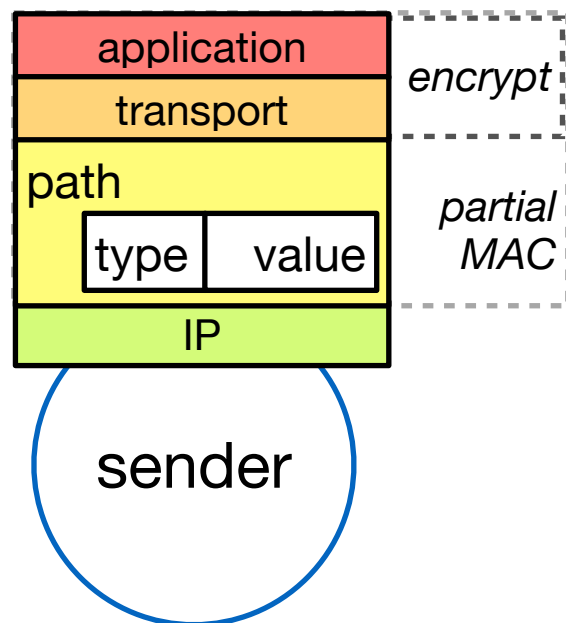


Path to Receiver Signaling with Feedback



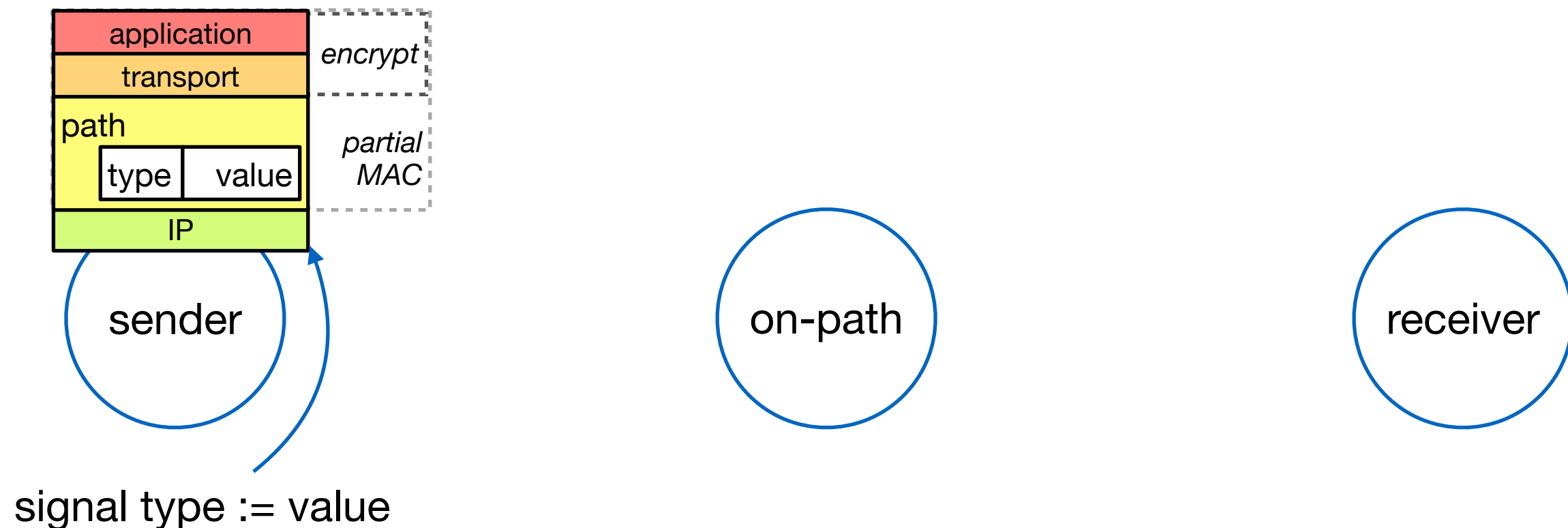


Path to Receiver Signaling with Feedback



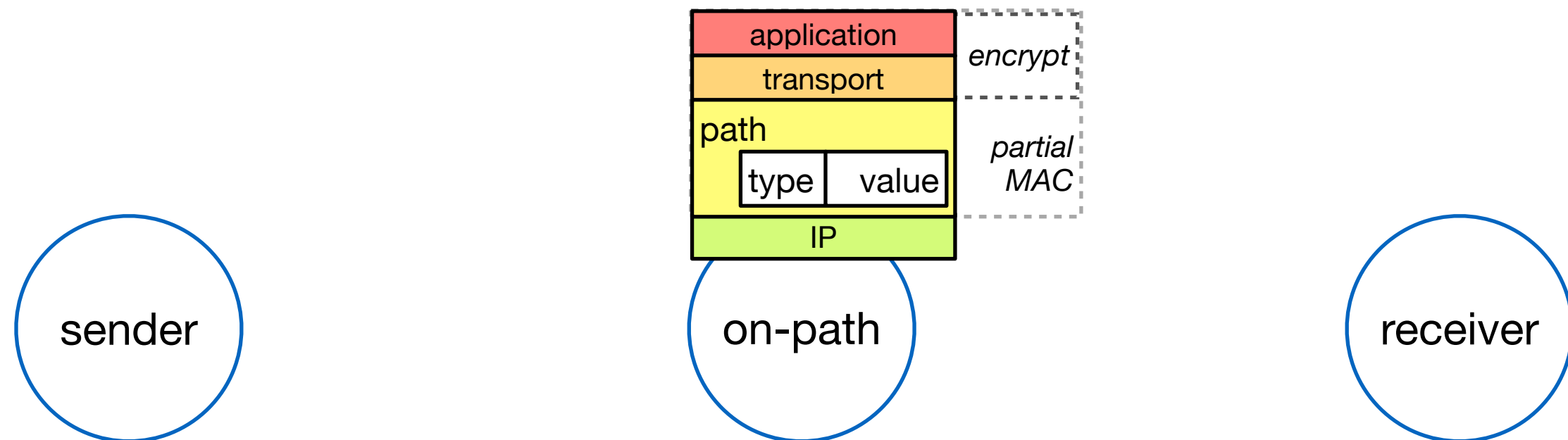


Path to Receiver Signaling with Feedback



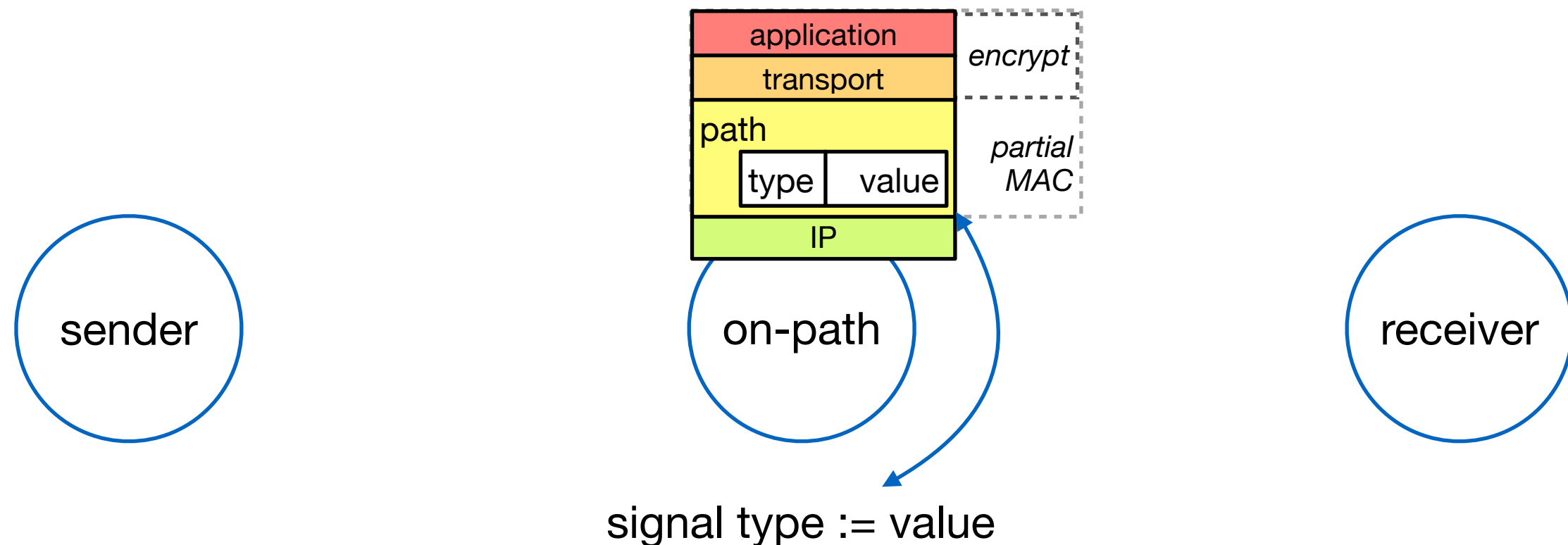


Path to Receiver Signaling with Feedback



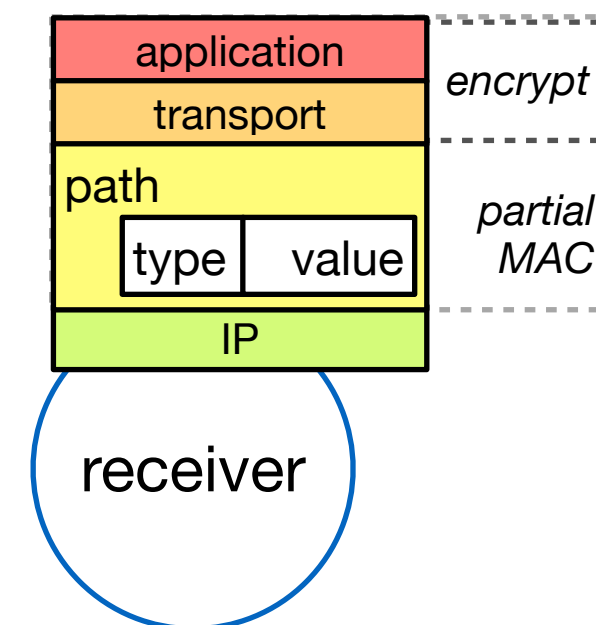
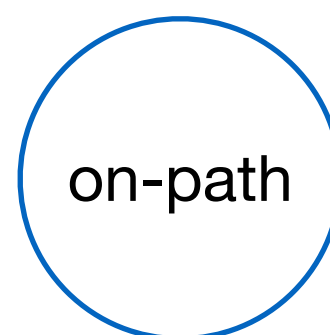
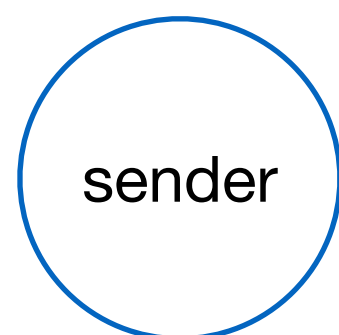


Path to Receiver Signaling with Feedback



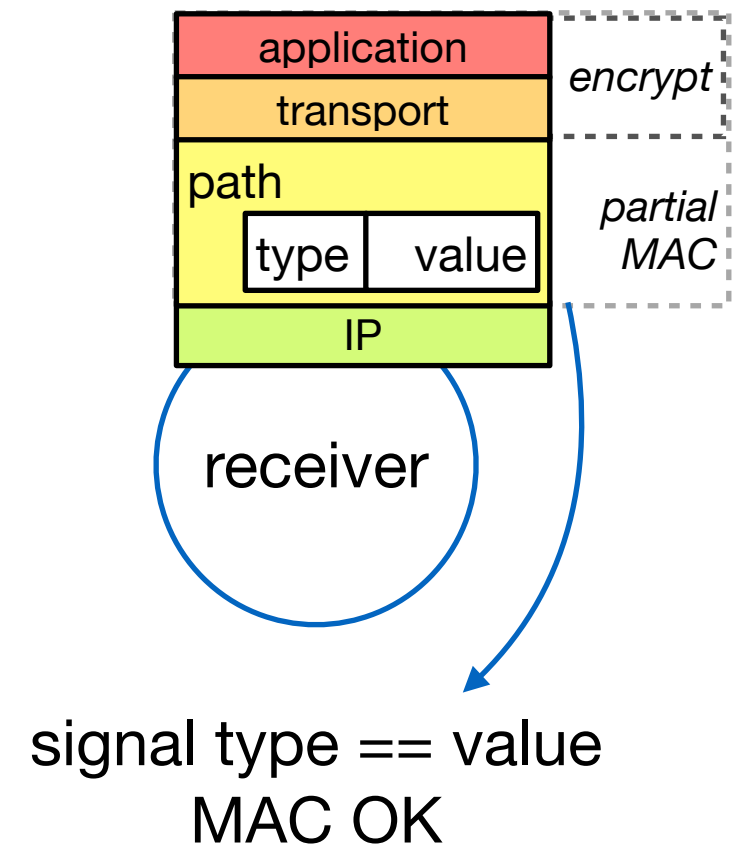
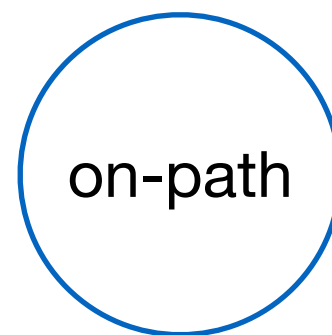
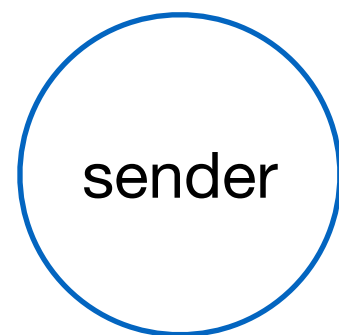


Path to Receiver Signaling with Feedback



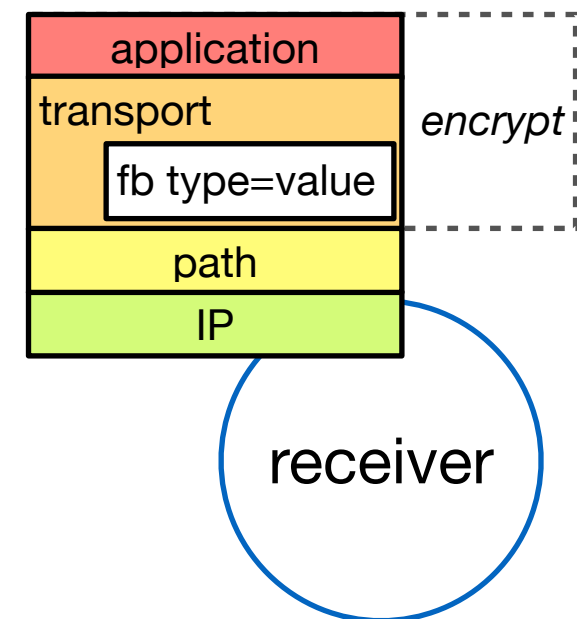
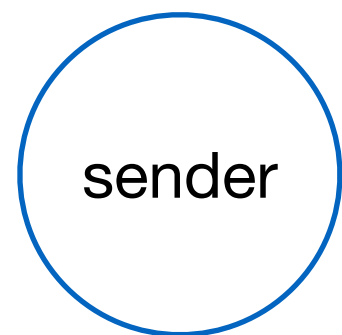


Path to Receiver Signaling with Feedback





Path to Receiver Signaling with Feedback



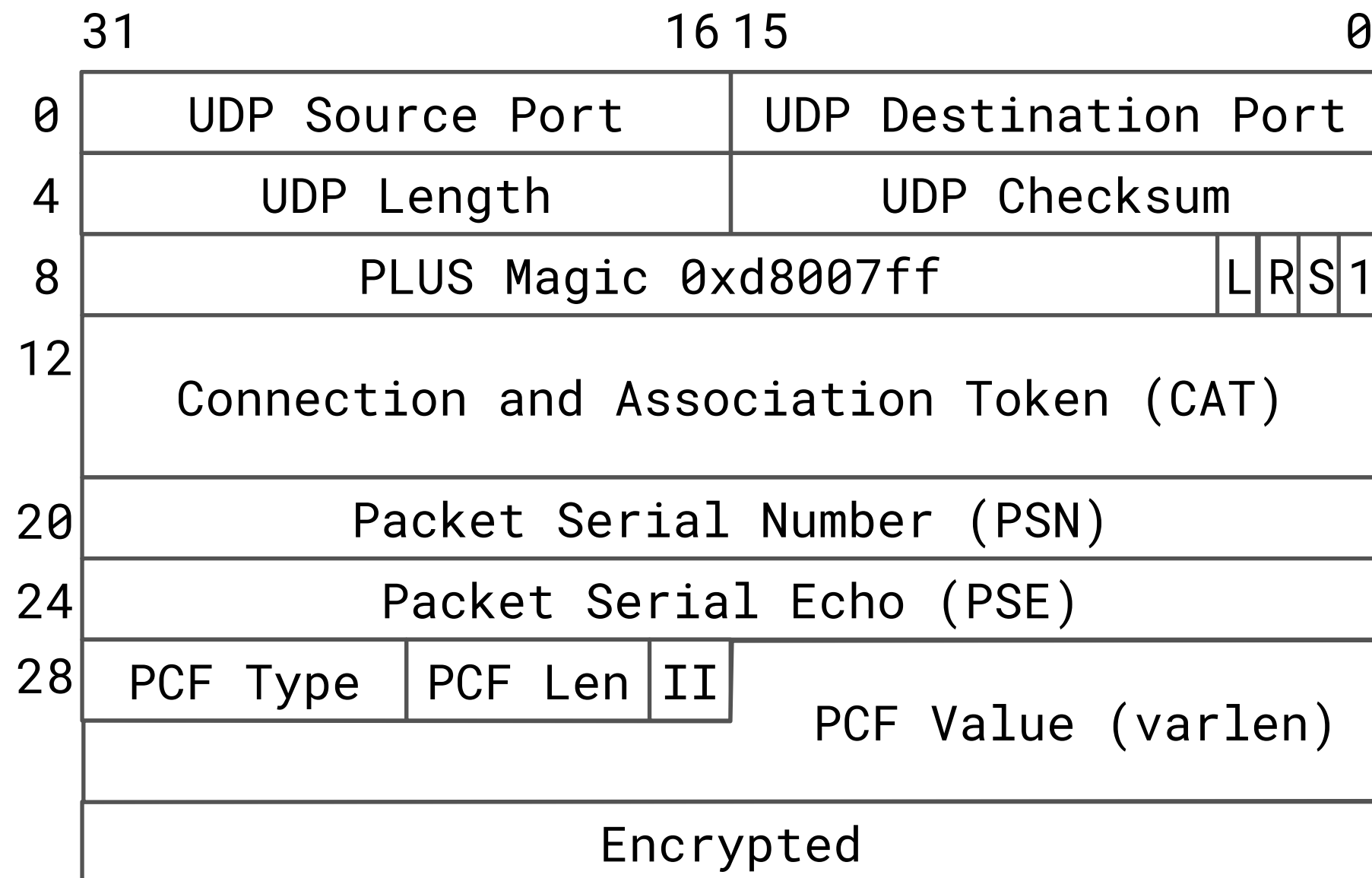


Path to Receiver Signaling with Feedback



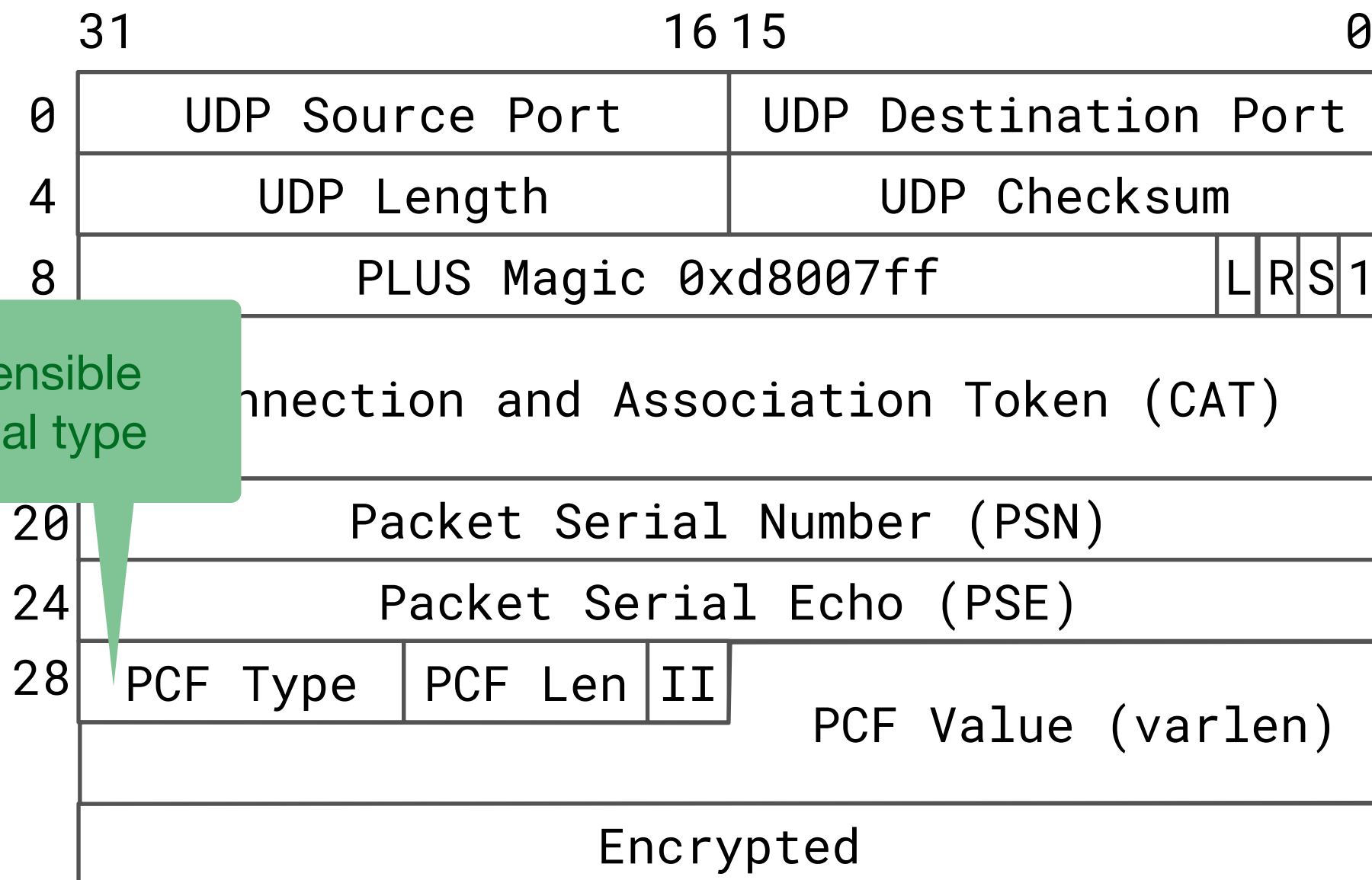


Extended PLUS Header





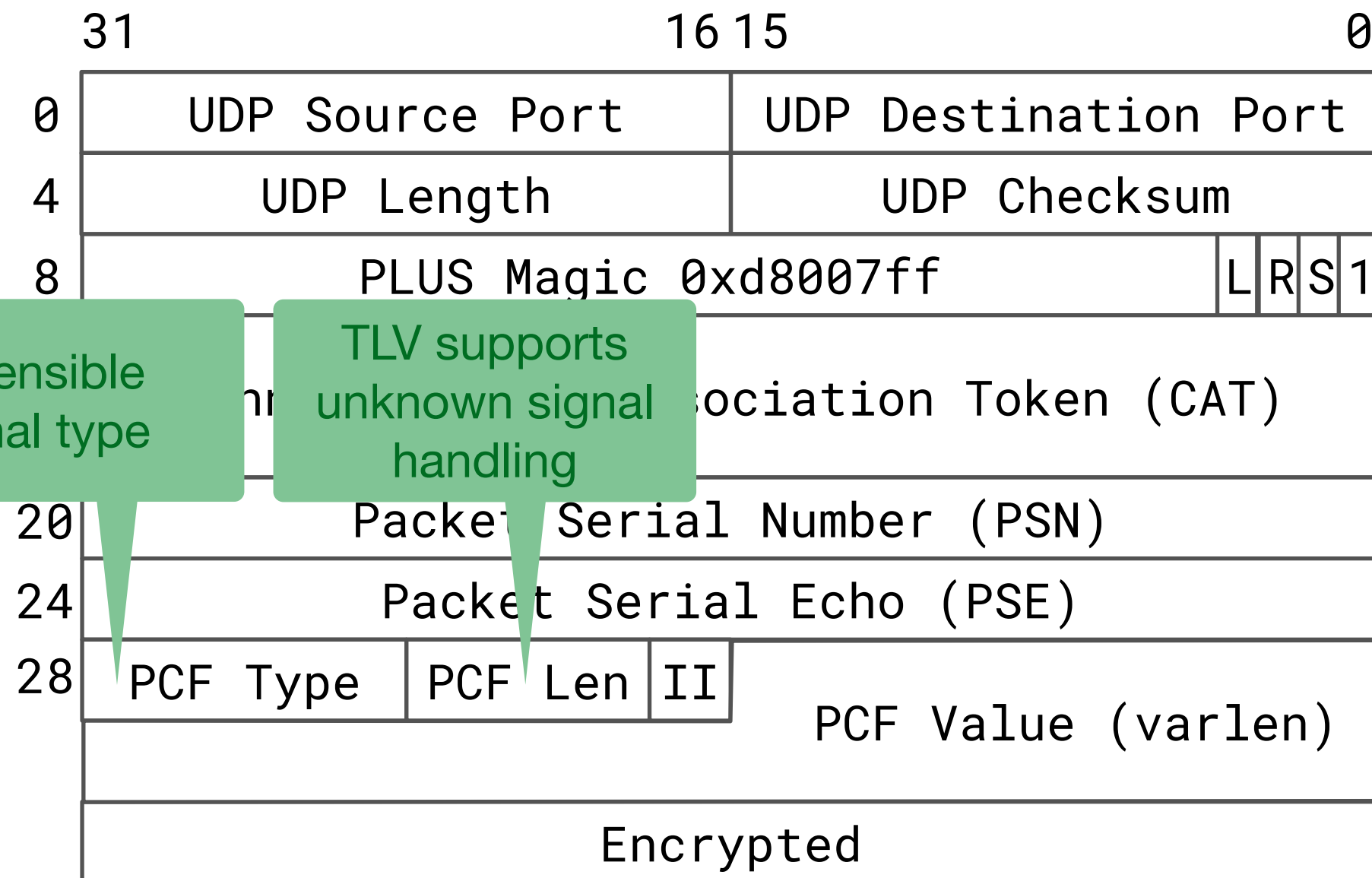
Extended PLUS Header



Extensible
signal type



Extended PLUS Header

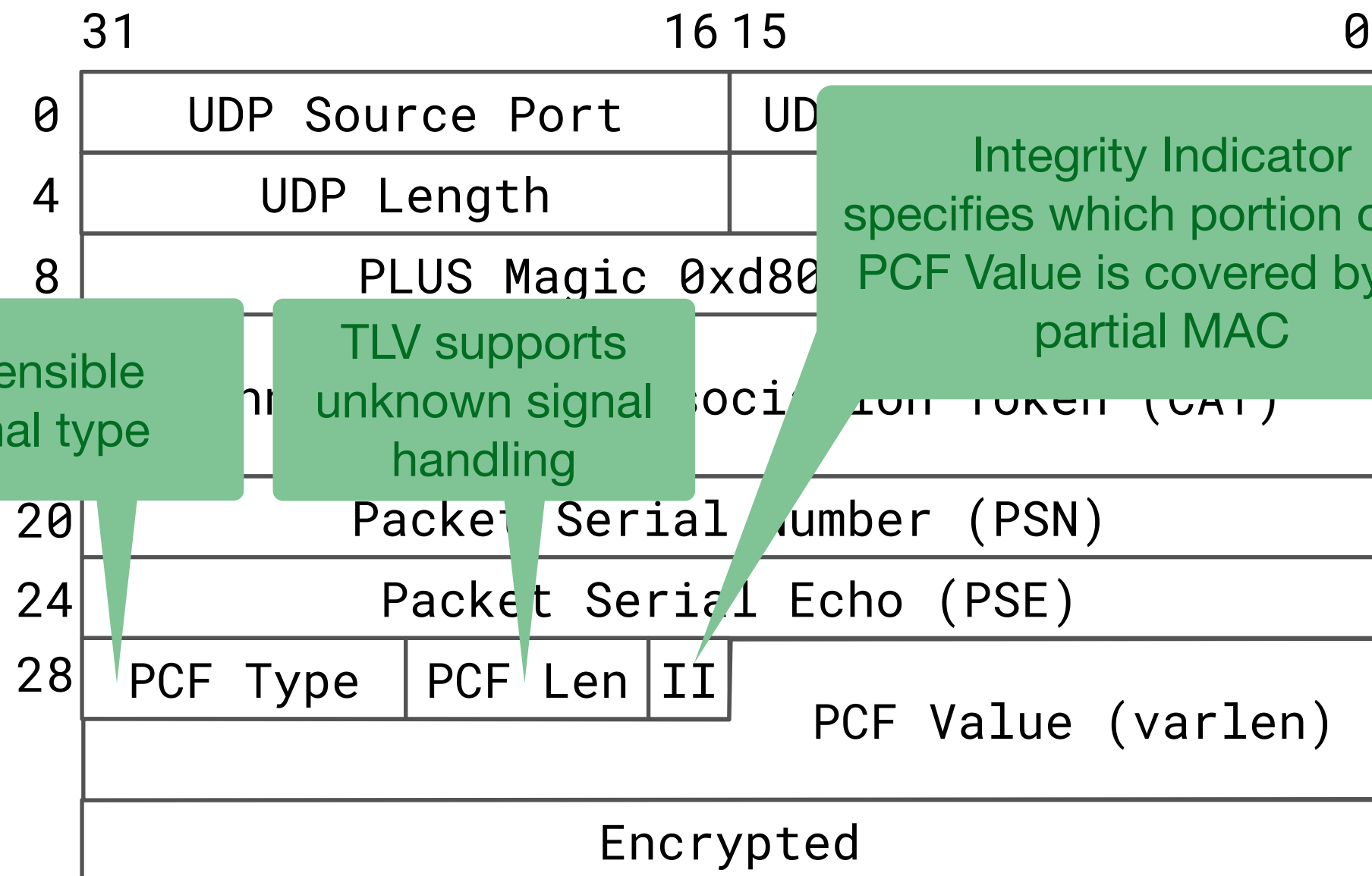


Extensible
signal type

TLV supports
unknown signal
handling



Extended PLUS Header



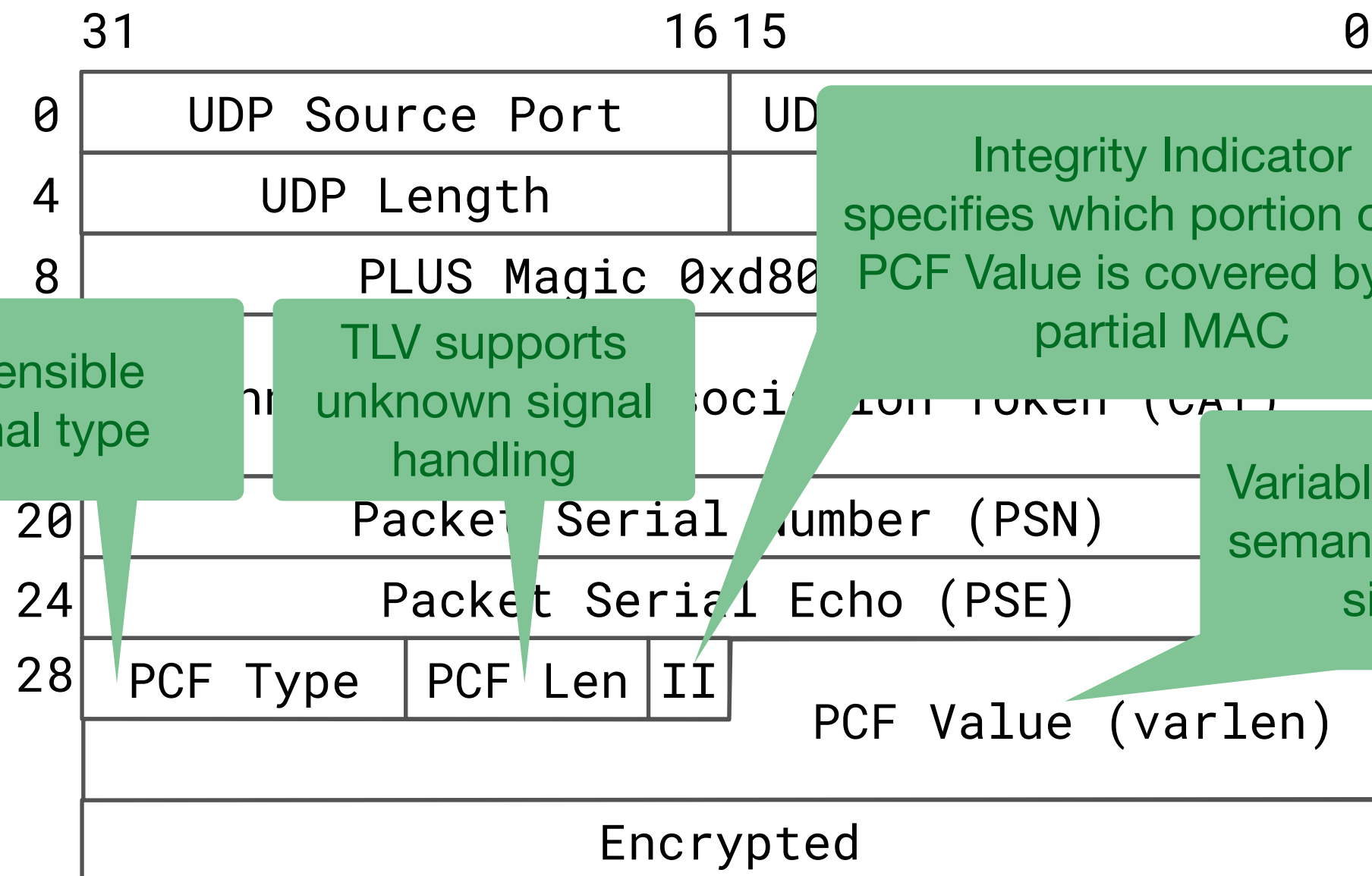
Extensible
signal type

TLV supports
unknown signal
handling

Integrity Indicator
specifies which portion of the
PCF Value is covered by the
partial MAC



Extended PLUS Header



Extensible
signal type

TLV supports
unknown signal
handling

Integrity Indicator
specifies which portion of the
PCF Value is covered by the
partial MAC

Variable-length value,
semantics defined by
signal type



Loss and Congestion Measurement

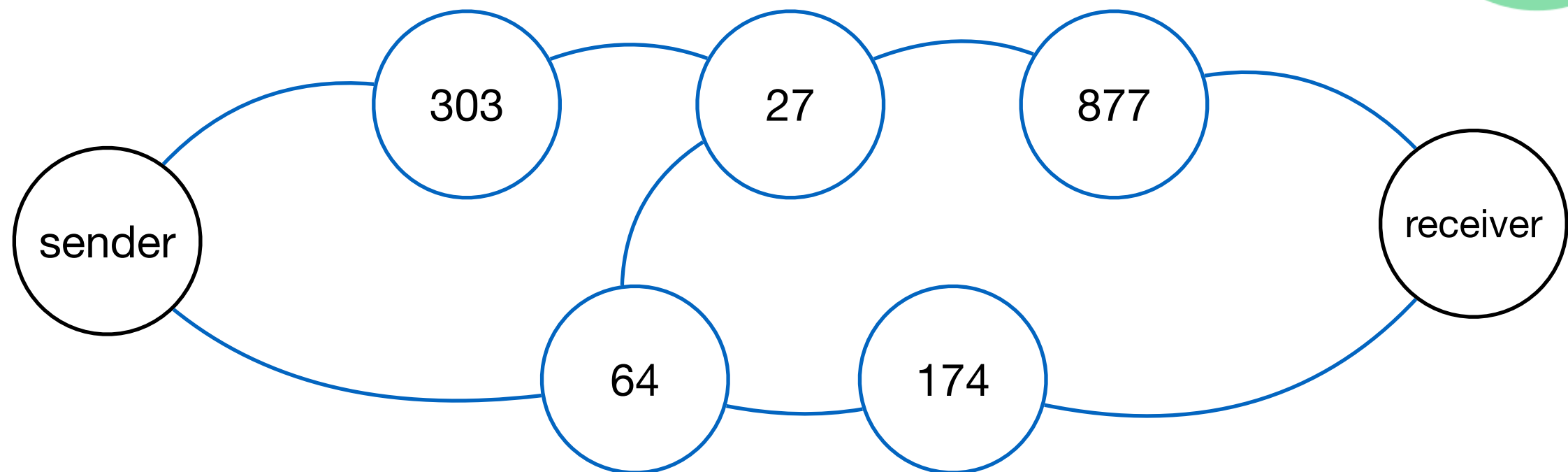
- PSN is serial, so sequence gaps can be used to estimate one-point upstream loss and loss between two points.
- Full-path loss requires signaling using extended header:

PCF type: 1	len:[2,4,8,16]	II:
Cumulative Loss Count (uint[8,16,32,64])		
Cumulative ECE Count (uint[8,16,32,64])		

- Feed-forward of cumulative loss and ECE seen by sender allows accurate counting anywhere along the path.
- Sender-side sampling allows efficiency tradeoff.



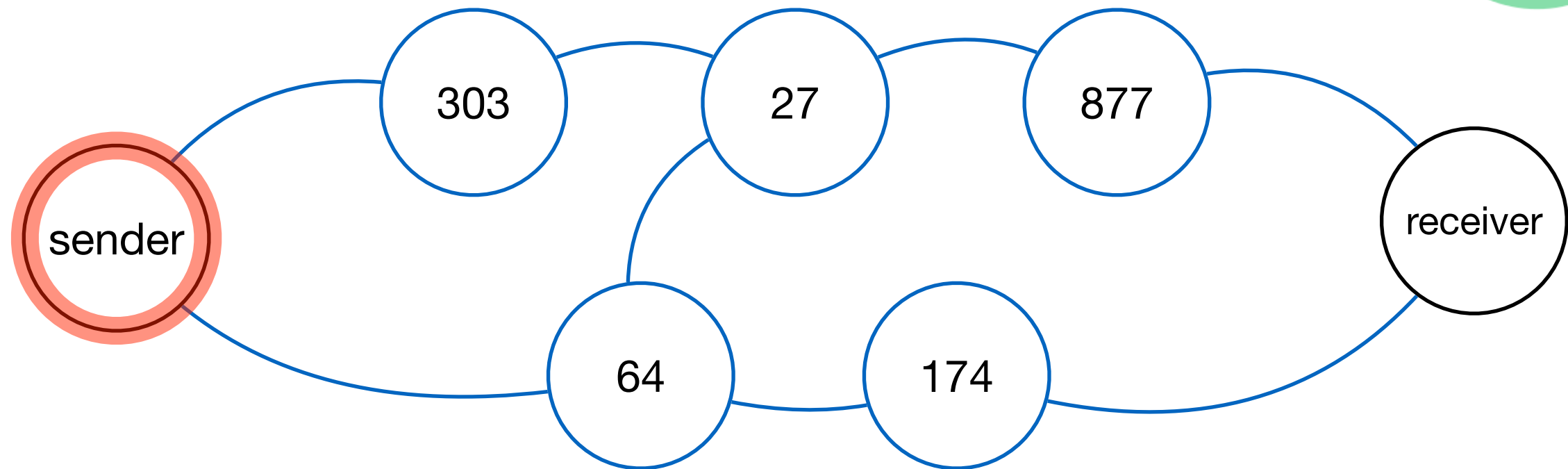
Path Tracing



- Each PLUS-aware hop XORs random value per node to PCF type 4 value.
- Value at receiver indicates which path was taken without identifying path.



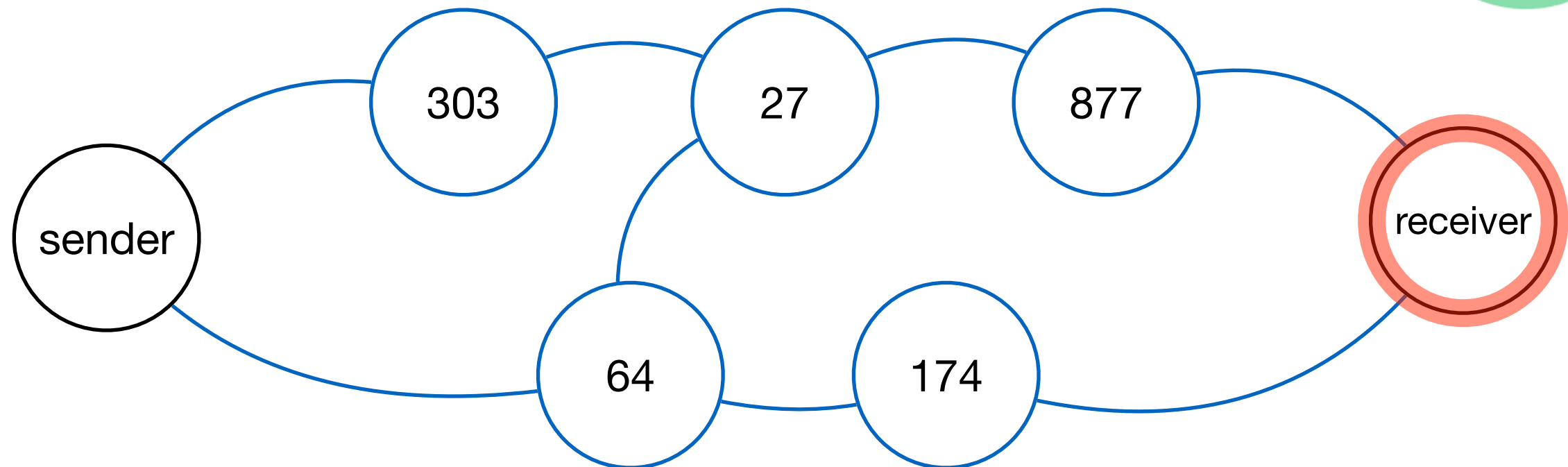
Path Tracing



- Each PLUS-aware hop XORs random value per node to PCF type 4 value.
- Value at receiver indicates which path was taken without identifying path.



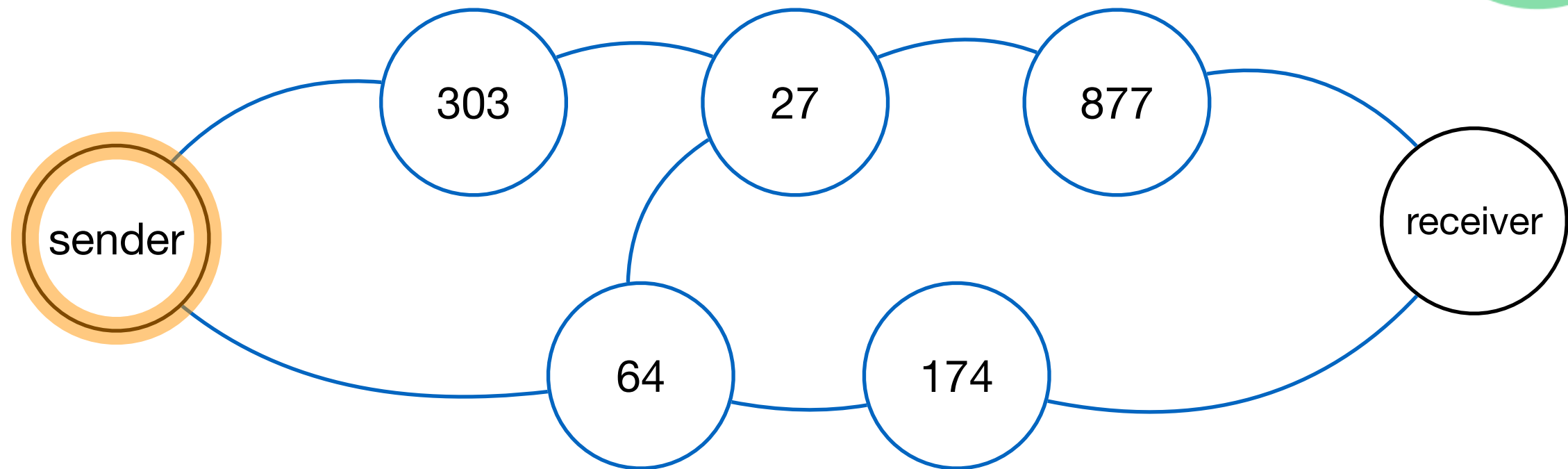
Path Tracing



- Each PLUS-aware hop XORs random value per node to PCF type 4 value.
- Value at receiver indicates which path was taken without identifying path.
- **Red** path: 1207



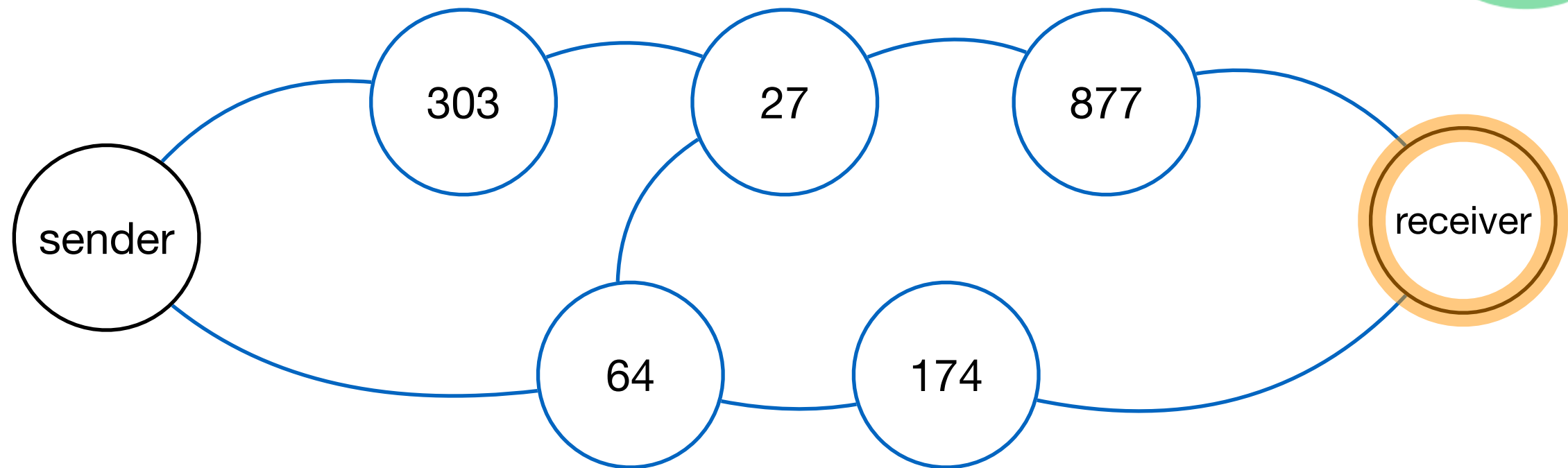
Path Tracing



- Each PLUS-aware hop XORs random value per node to PCF type 4 value.
- Value at receiver indicates which path was taken without identifying path.
- **Red** path: 1207



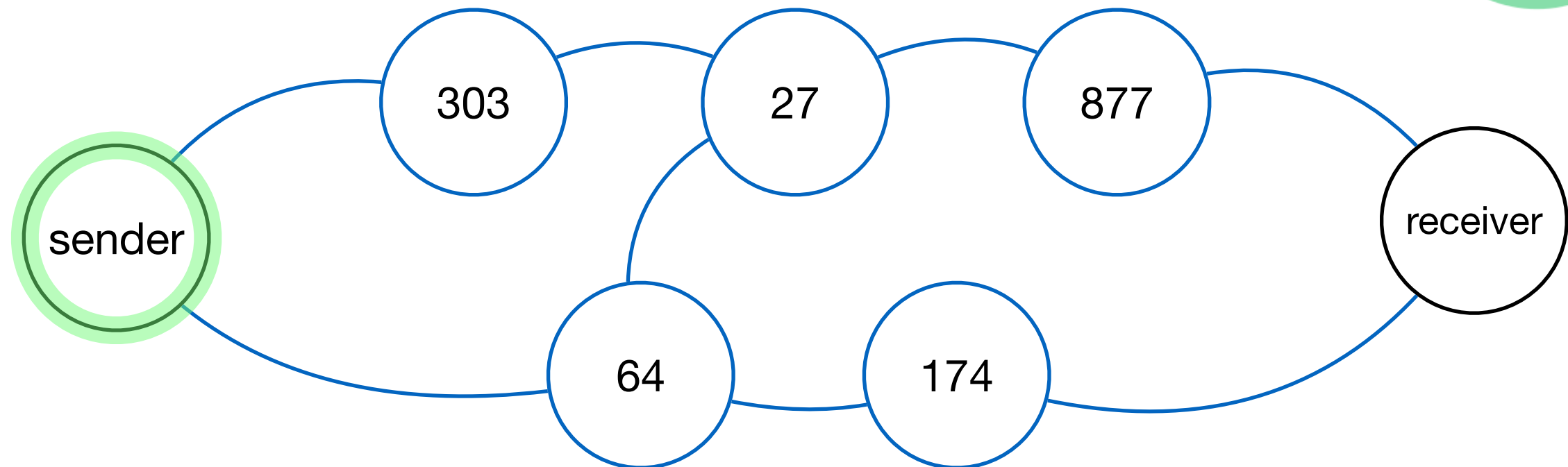
Path Tracing



- Each PLUS-aware hop XORs random value per node to PCF type 4 value.
- Value at receiver indicates which path was taken without identifying path.
- **Red** path: 1207
- **Orange** path: 238



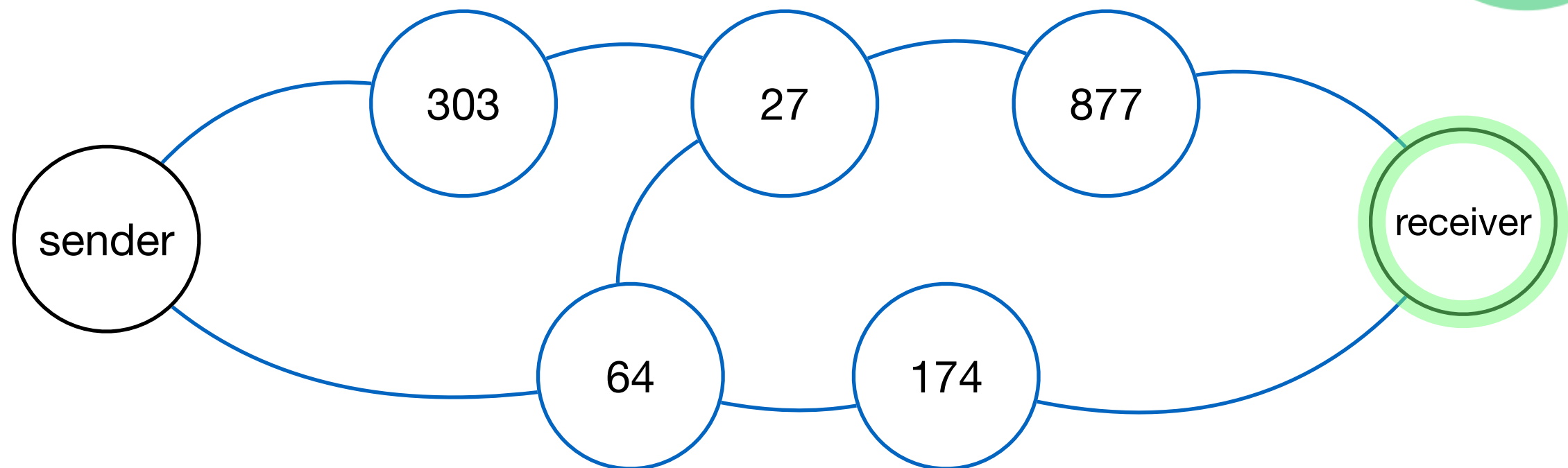
Path Tracing



- Each PLUS-aware hop XORs random value per node to PCF type 4 value.
- Value at receiver indicates which path was taken without identifying path.
- **Red** path: 1207
- **Orange** path: 238



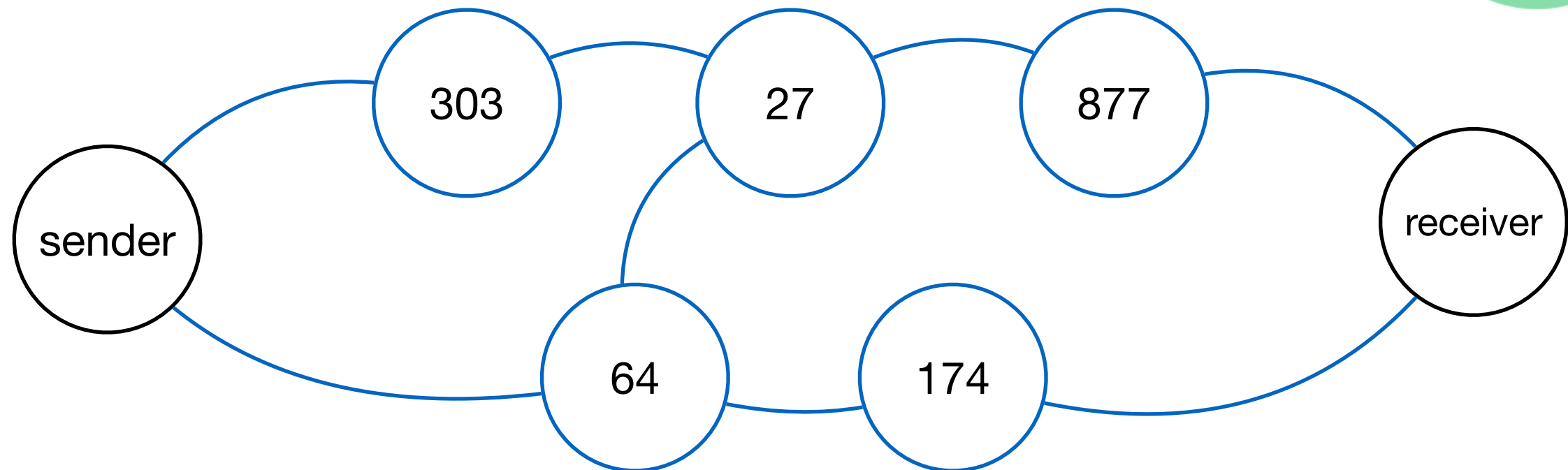
Path Tracing



- Each PLUS-aware hop XORs random value per node to PCF type 4 value.
- Value at receiver indicates which path was taken without identifying path.
- **Red** path: 1207
- **Orange** path: 238
- **Green** path: 968



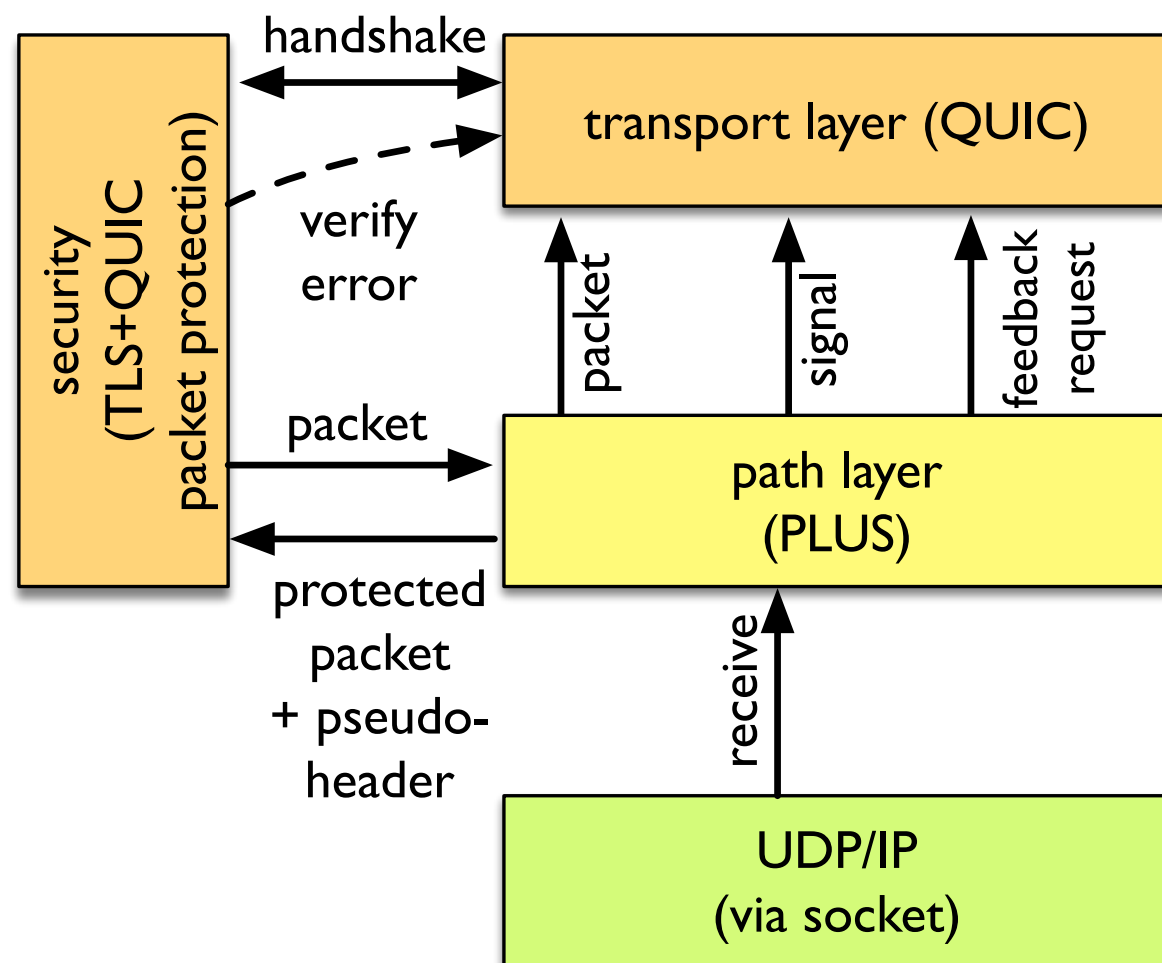
Path Tracing



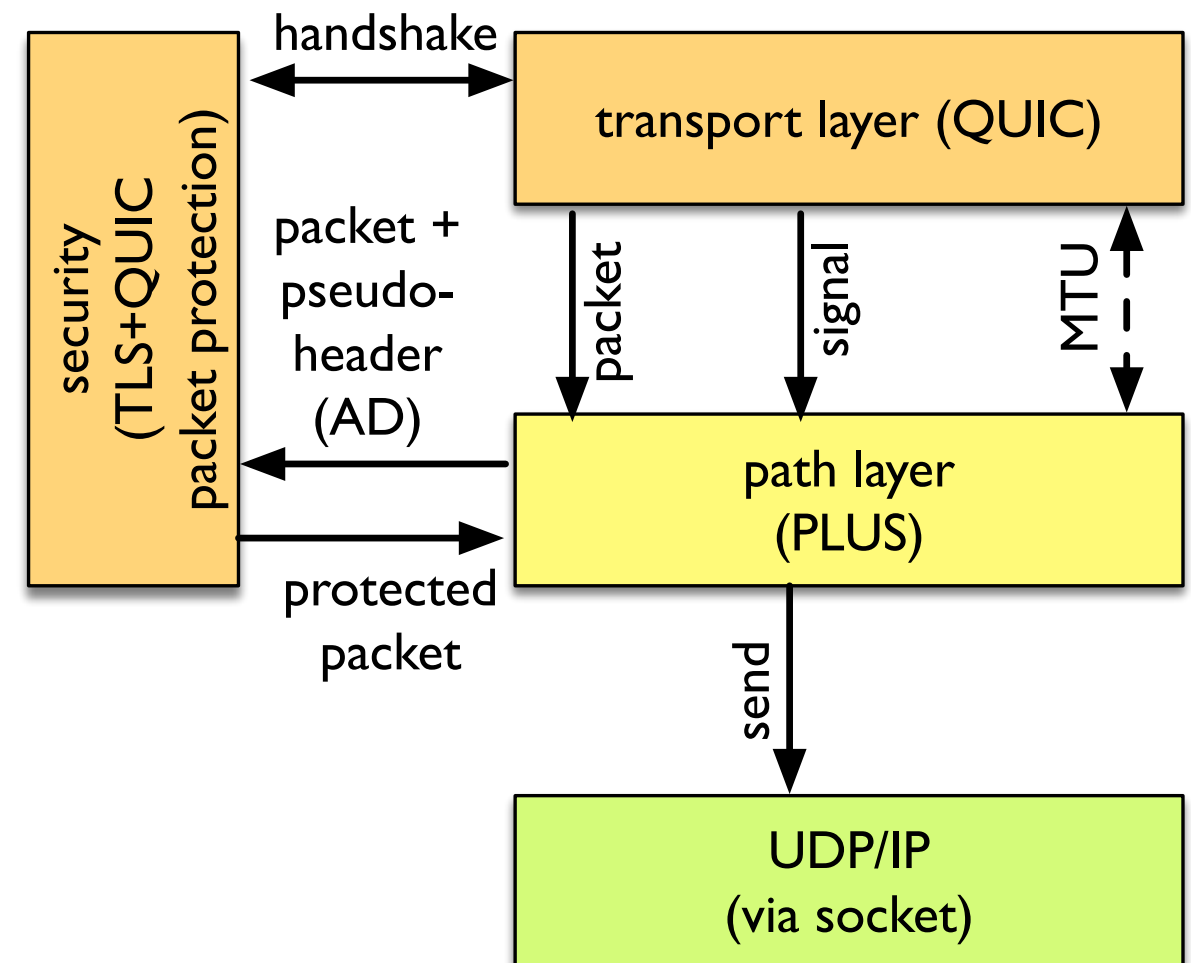
- Each PLUS-aware hop XORs random value per node to PCF type 4 value.
- Value at receiver indicates which path was taken without identifying path.
- **Red** path: 1207
- **Orange** path: 238
- **Green** path: 968



Transport interfaces to PLUS: pilot implementation work under QUIC



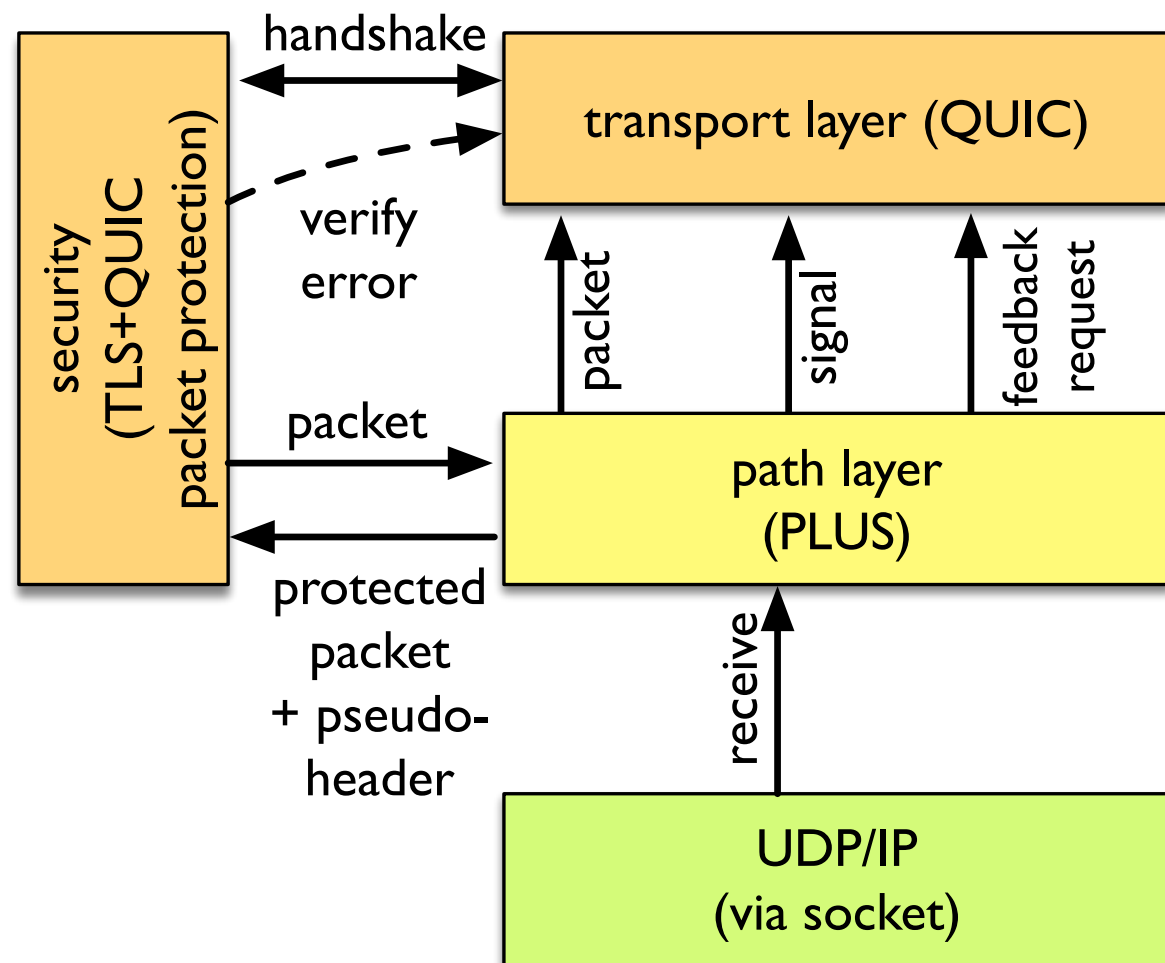
(a) receiver-side interfaces



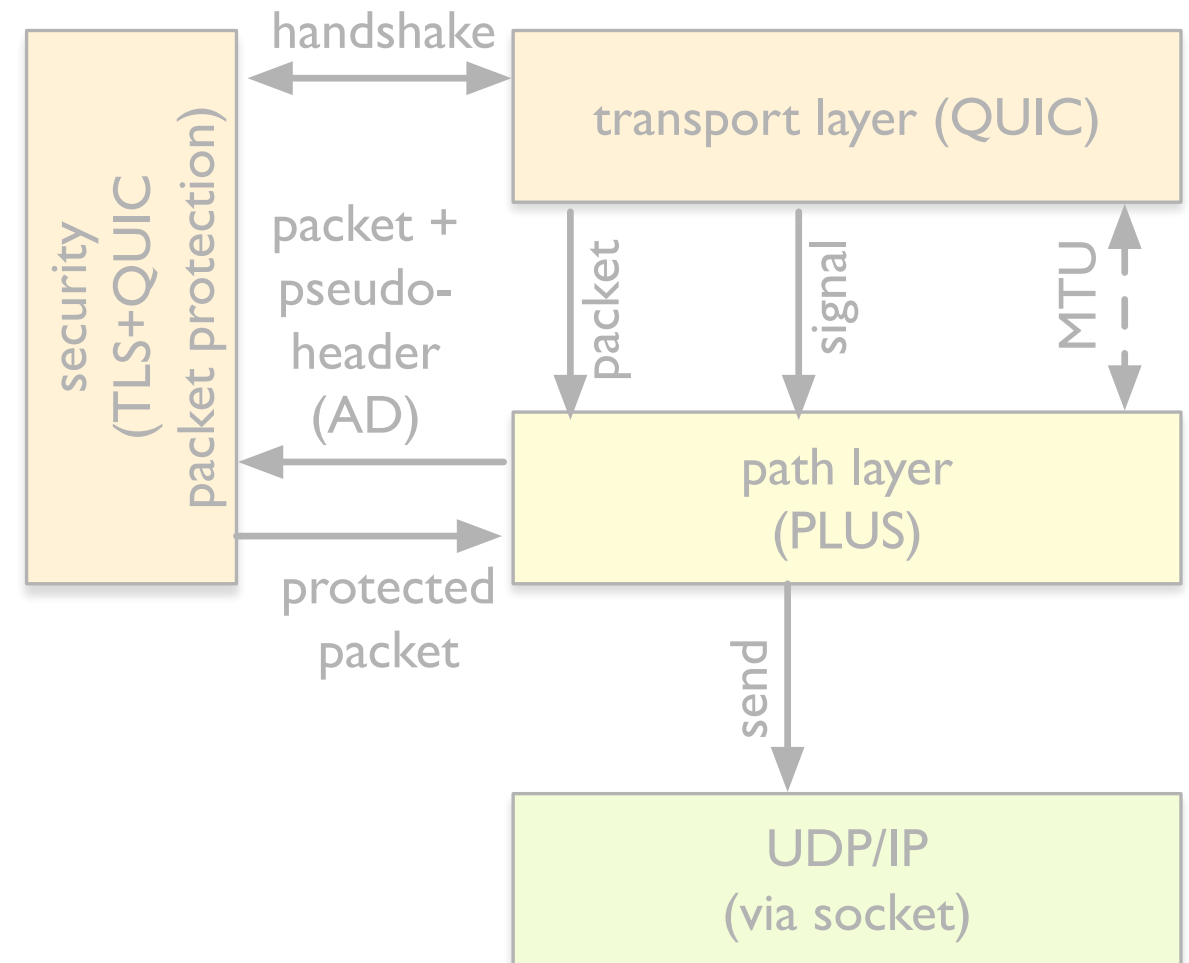
(b) sender-side interfaces



Transport interfaces to PLUS: pilot implementation work under QUIC



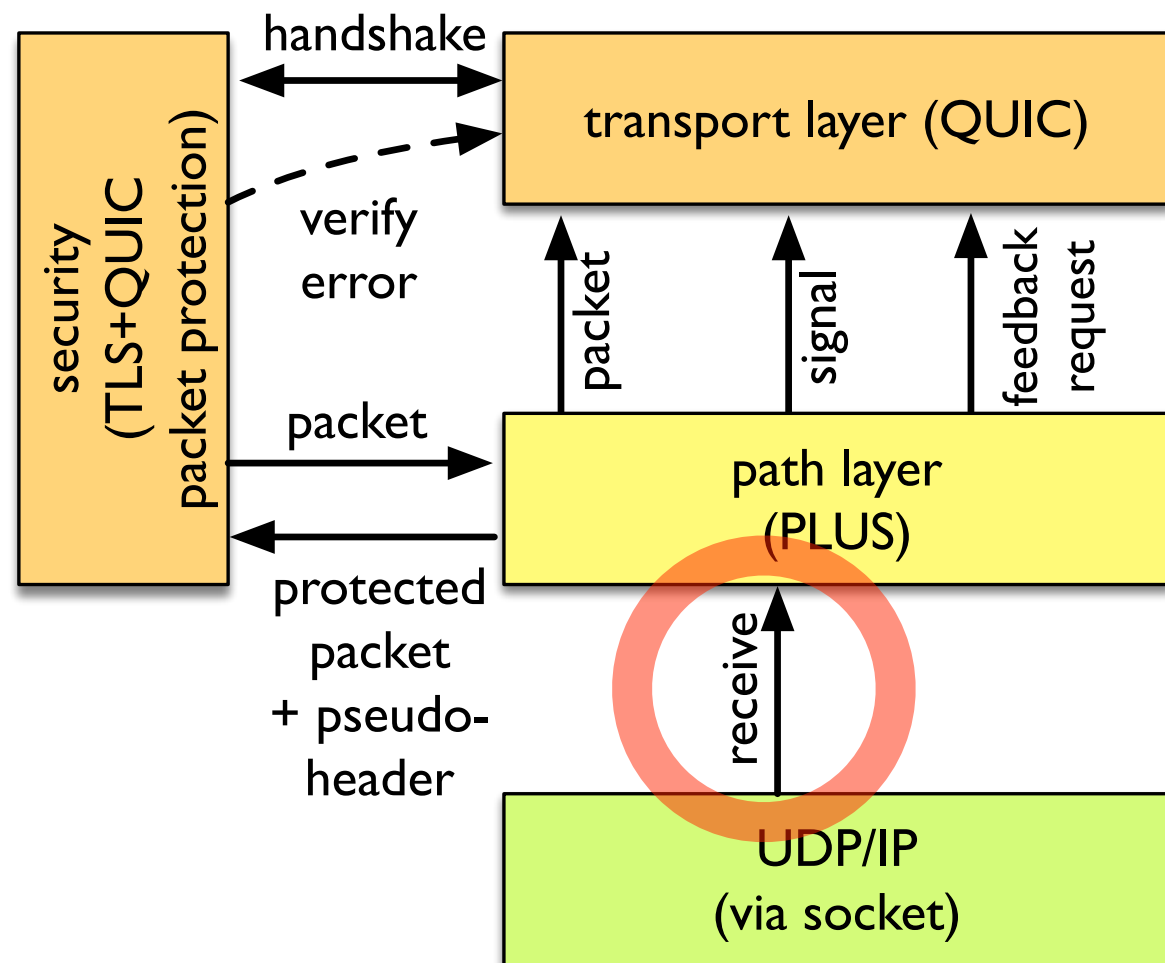
(a) receiver-side interfaces



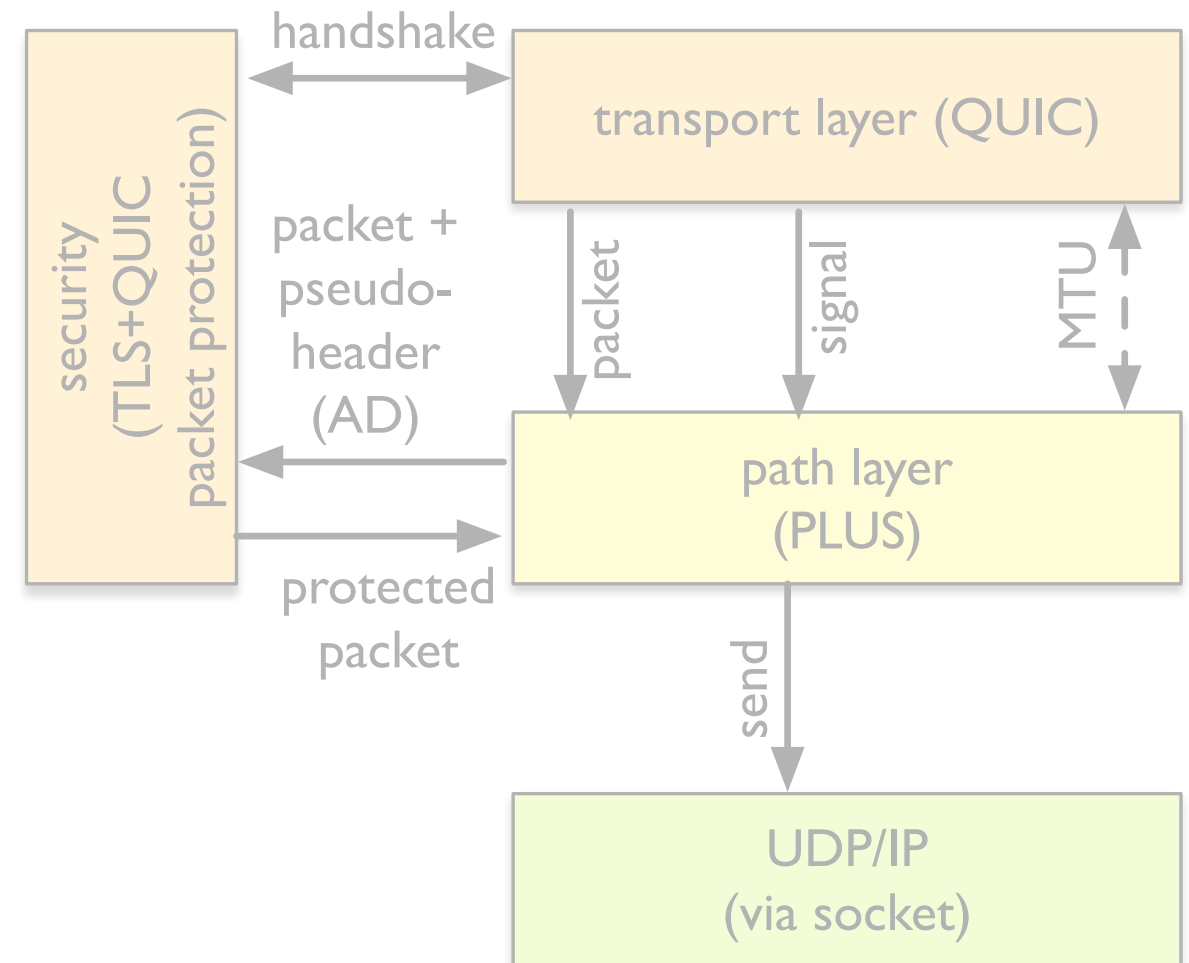
(b) sender-side interfaces



Transport interfaces to PLUS: pilot implementation work under QUIC



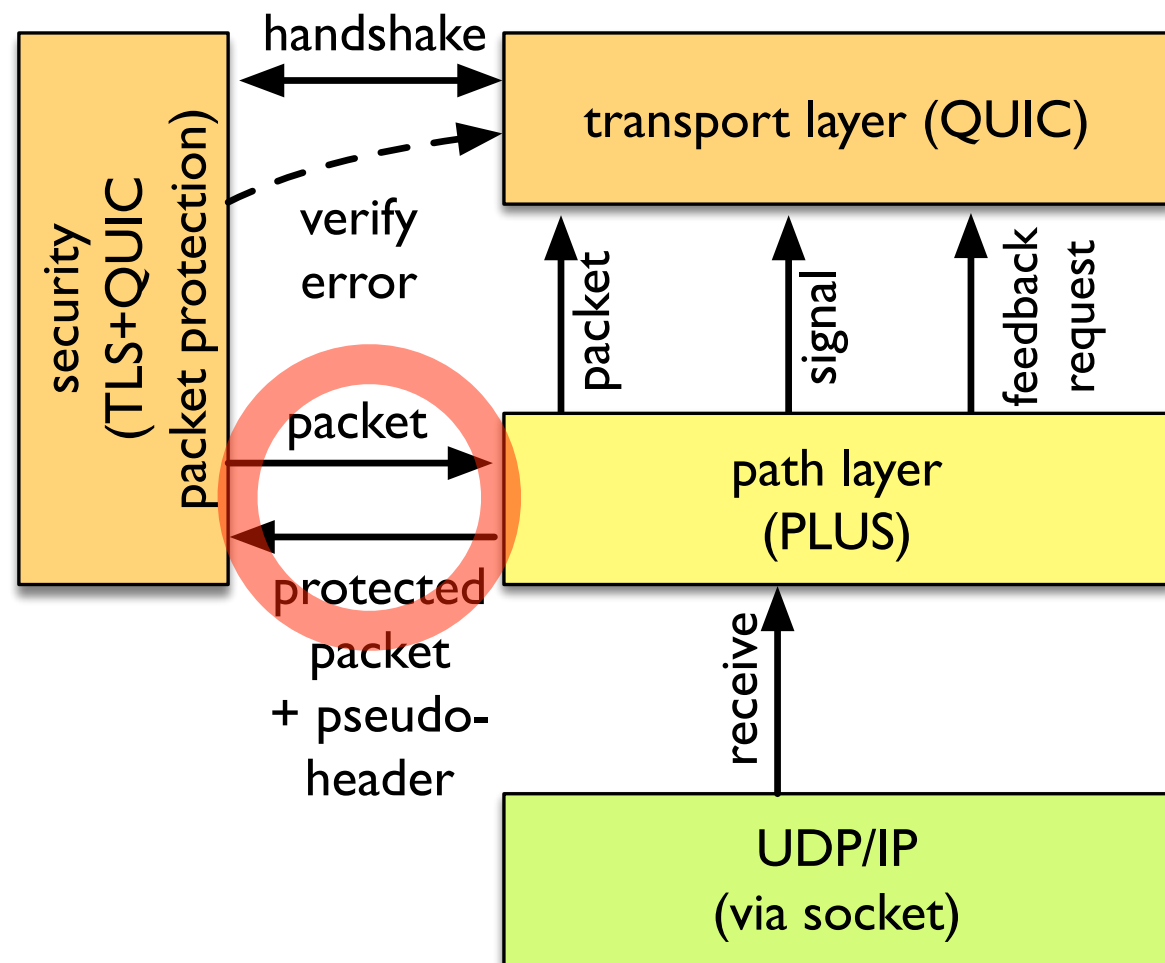
(a) receiver-side interfaces



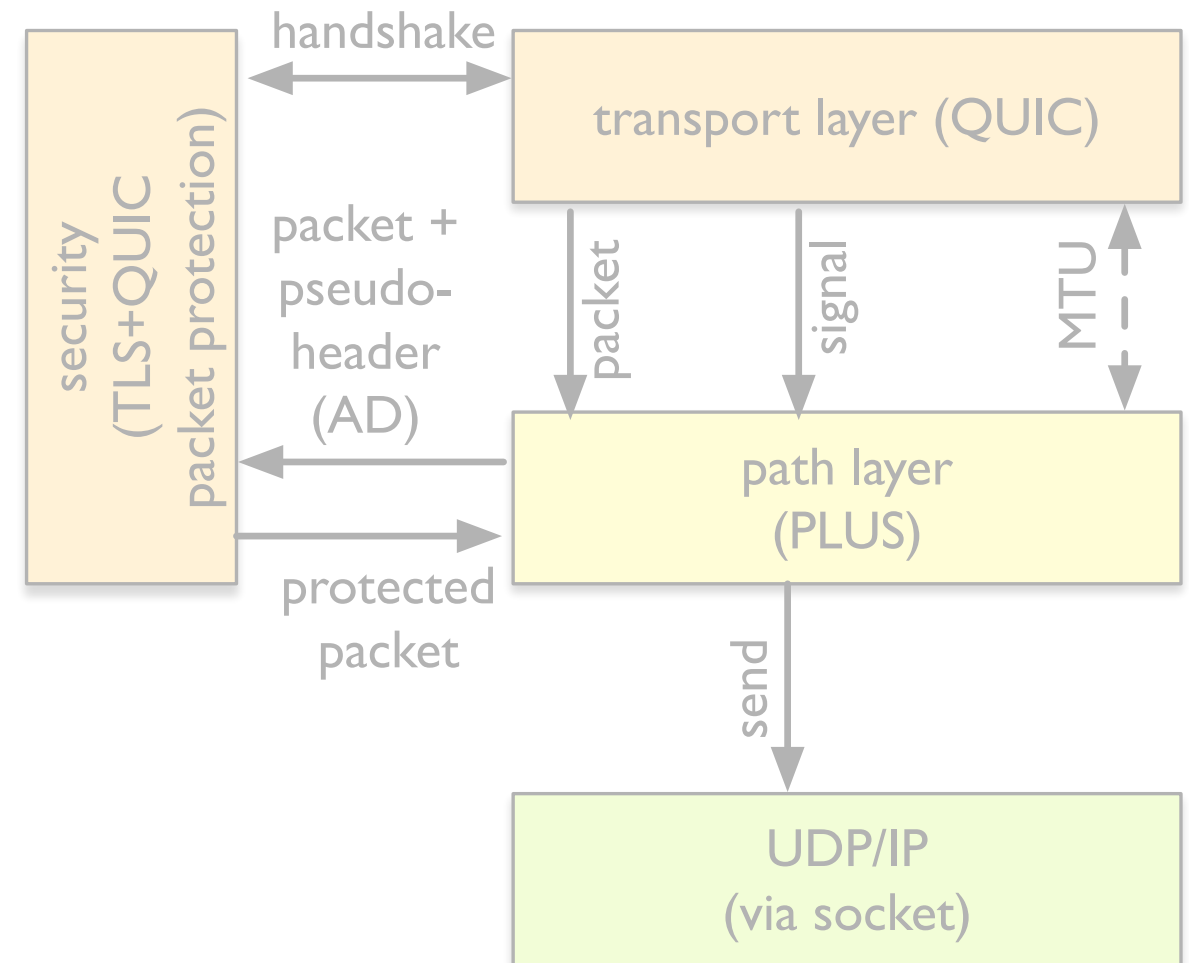
(b) sender-side interfaces



Transport interfaces to PLUS: pilot implementation work under QUIC



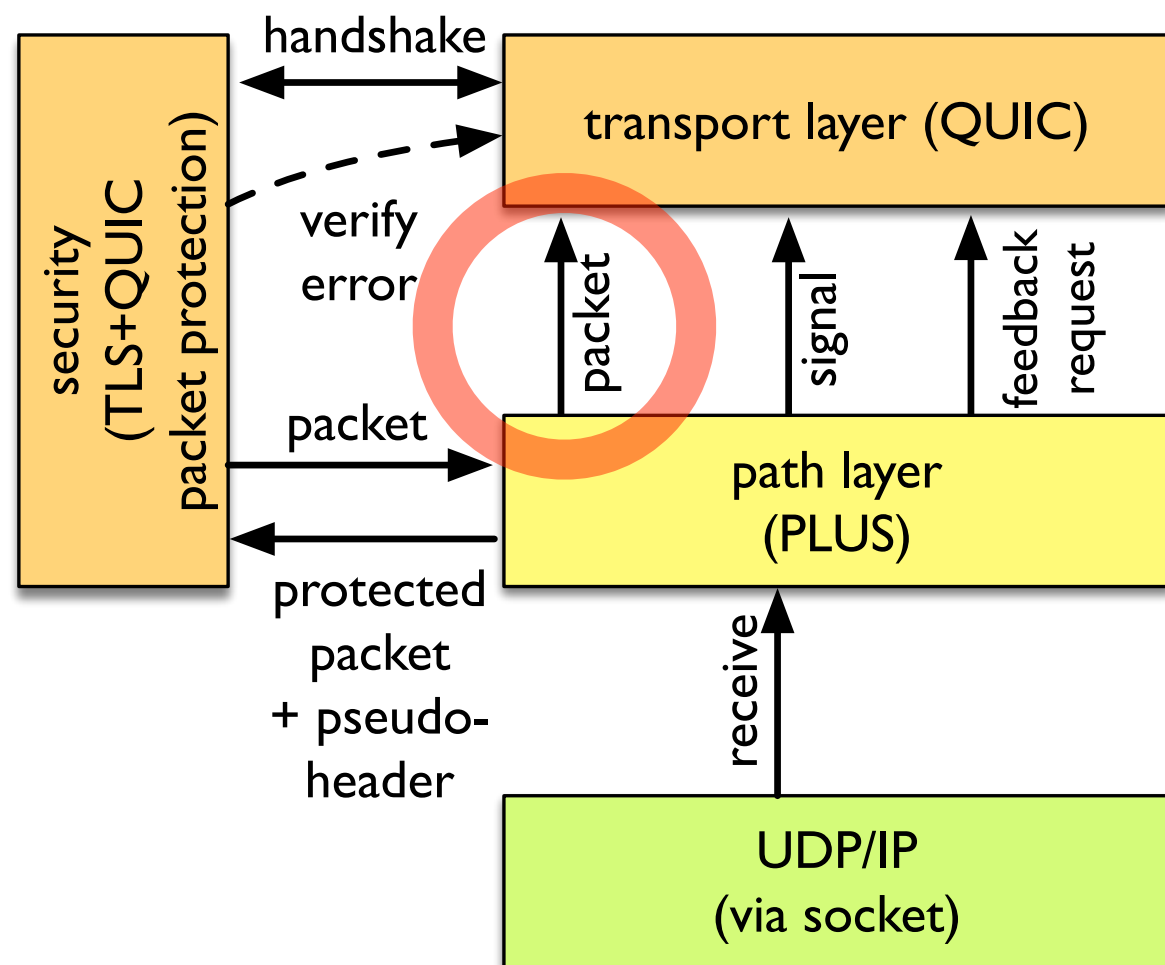
(a) receiver-side interfaces



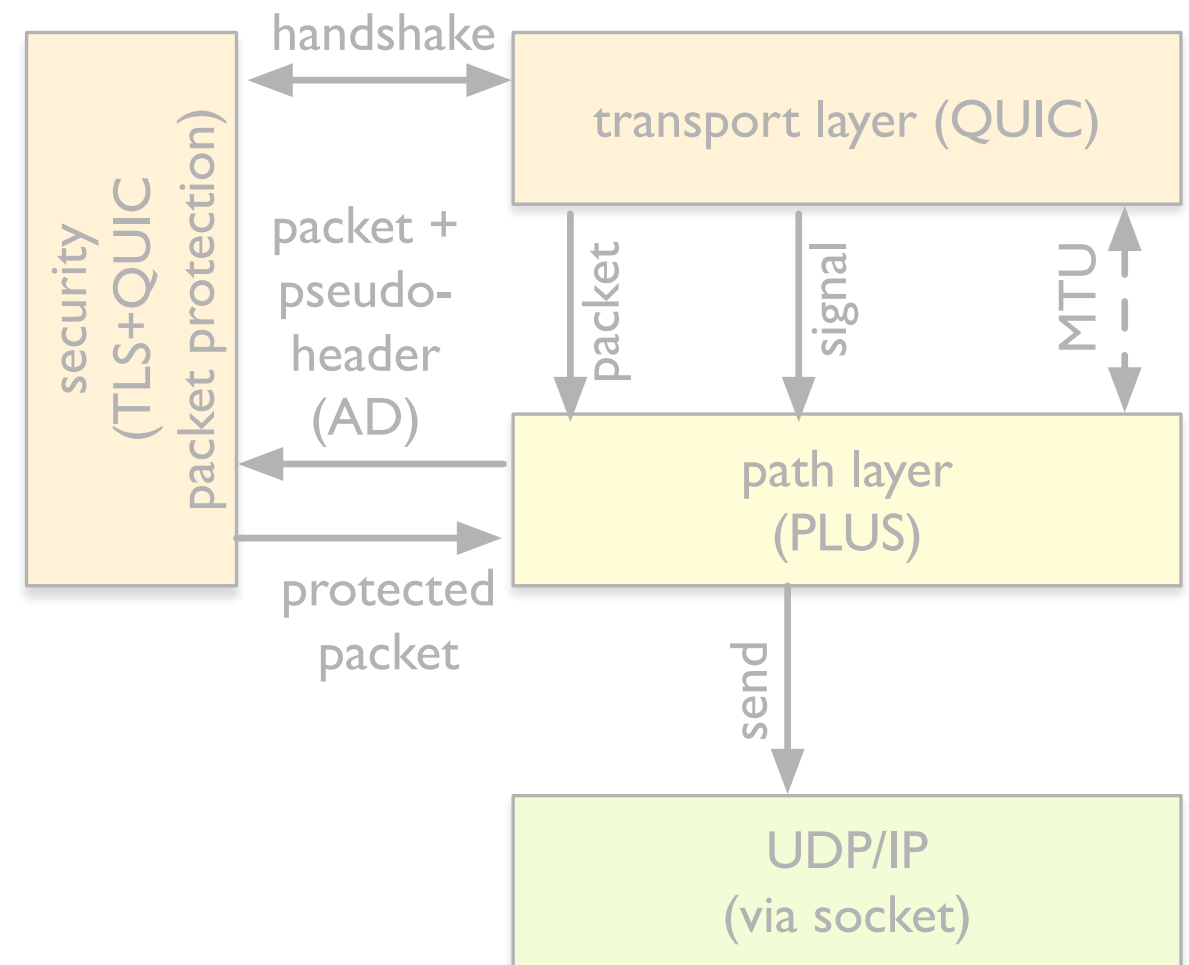
(b) sender-side interfaces



Transport interfaces to PLUS: pilot implementation work under QUIC



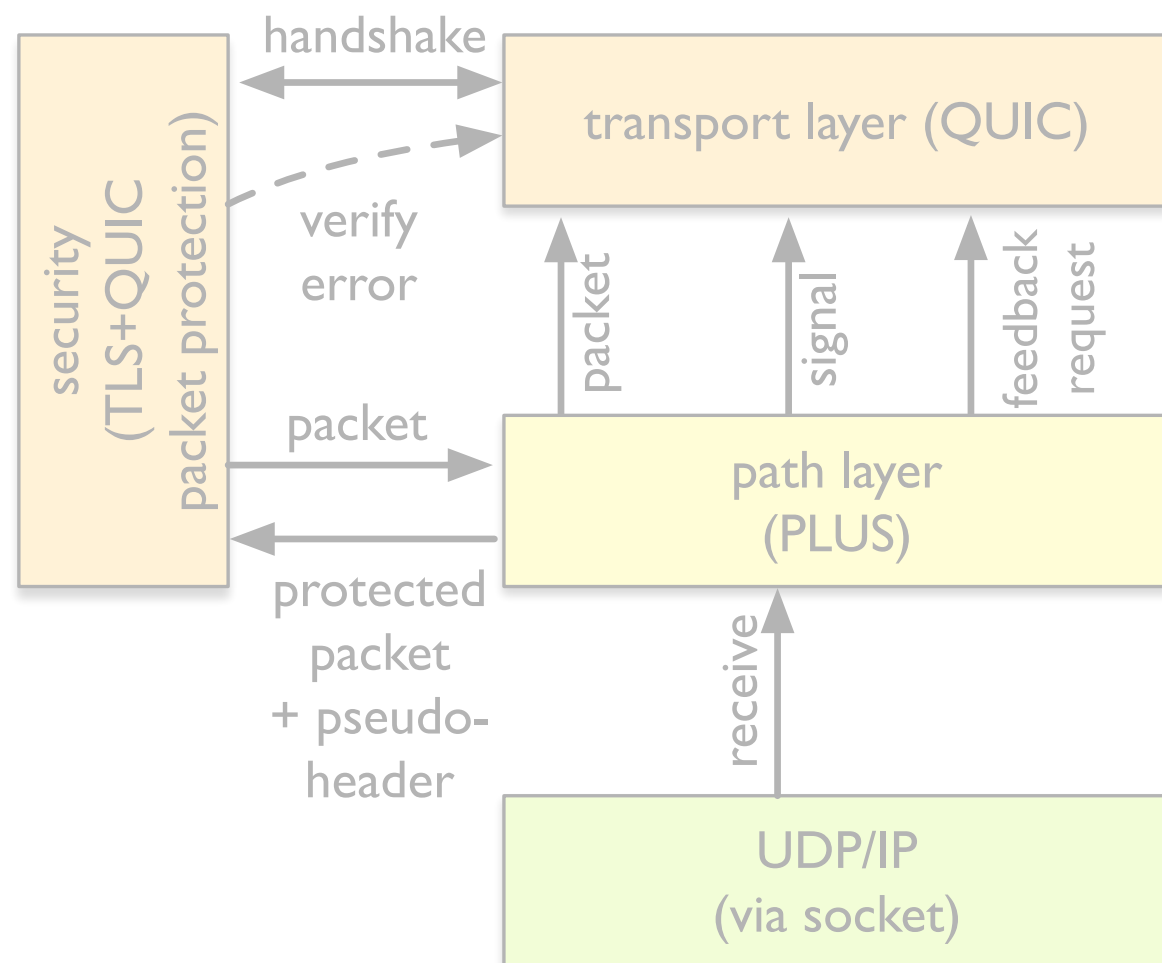
(a) receiver-side interfaces



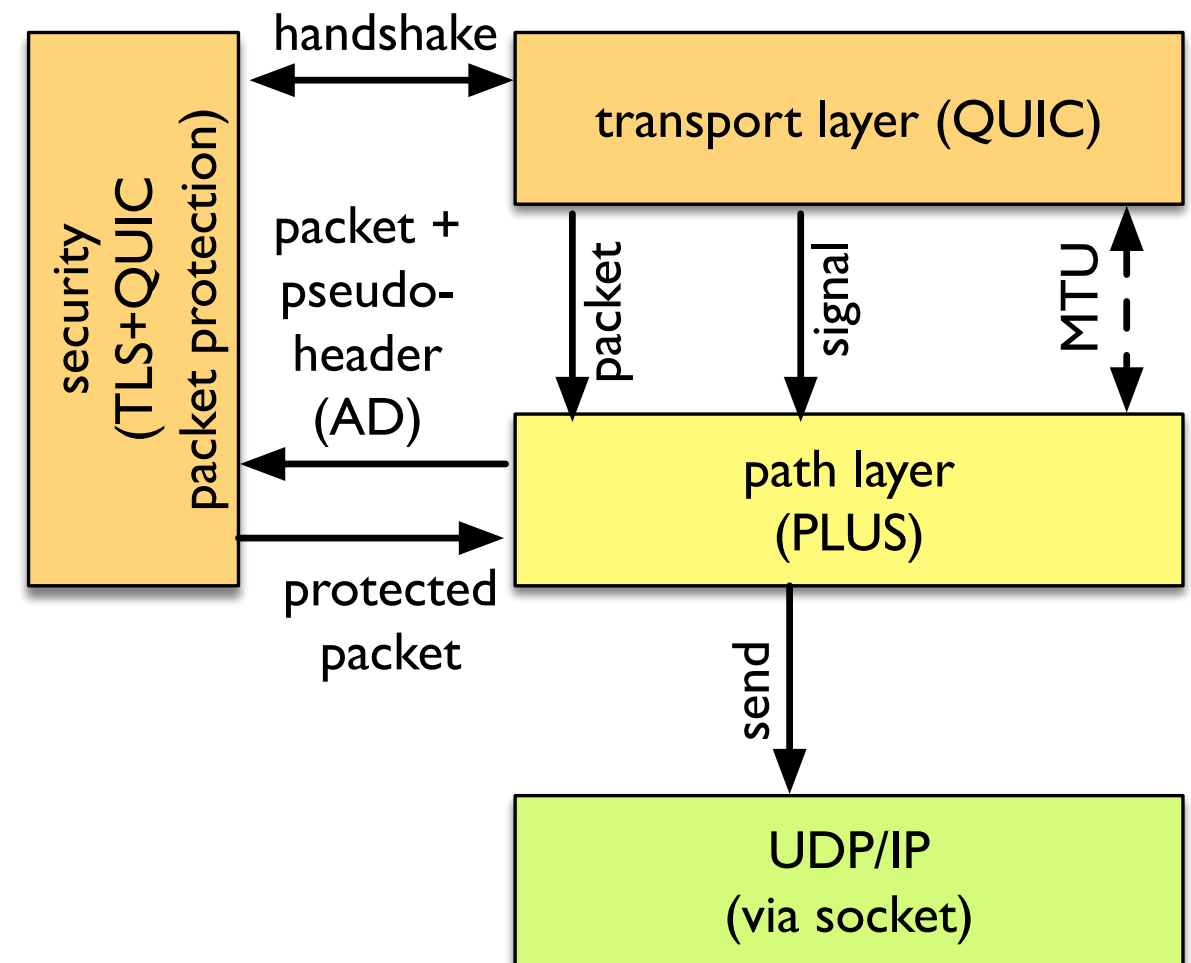
(b) sender-side interfaces



Transport interfaces to PLUS: pilot implementation work under QUIC



(a) receiver-side interfaces

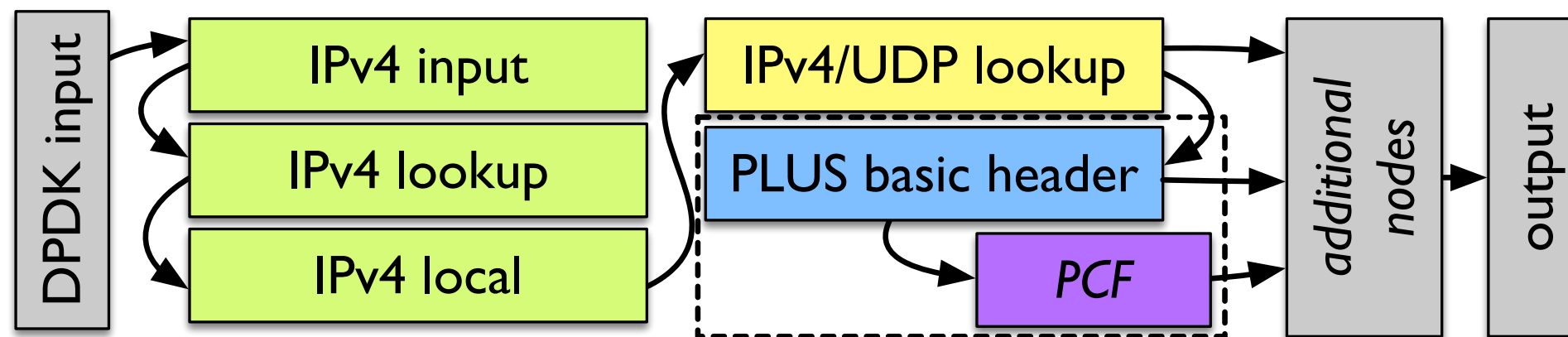


(b) sender-side interfaces



Building PLUS-aware middleboxes with fd.io VPP

- fd.io VPP: framework for building userspace network devices on any DPDK platform, using *packet vectors* for scalability.
- PLUS middlebox support implemented as VPP nodes
 - Core node handles state machine and basic header flags
 - One extension node per PCF type
 - Modifications to UDP logic to recognize PLUS magic





PLUS and QUIC

- Both PLUS and QUIC propose encryption and UDP encapsulation to enable transport evolution.
- PLUS proposes additional explicit signaling to replace information that encryption removes.
 - Declarative and advisory, but better than inference.
- Many basic PLUS features appear in QUIC in diminished form:
 - QUIC's PN is a PSN, but without echo
 - QUIC's CID is a CAT, but not on every packet
- Additional QUIC features proposed based on PLUS experience:
 - No PSE, but latency spin bit replaces it for passive RTT