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 Gorry Fairhurst, Univ. of Aberdeen

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

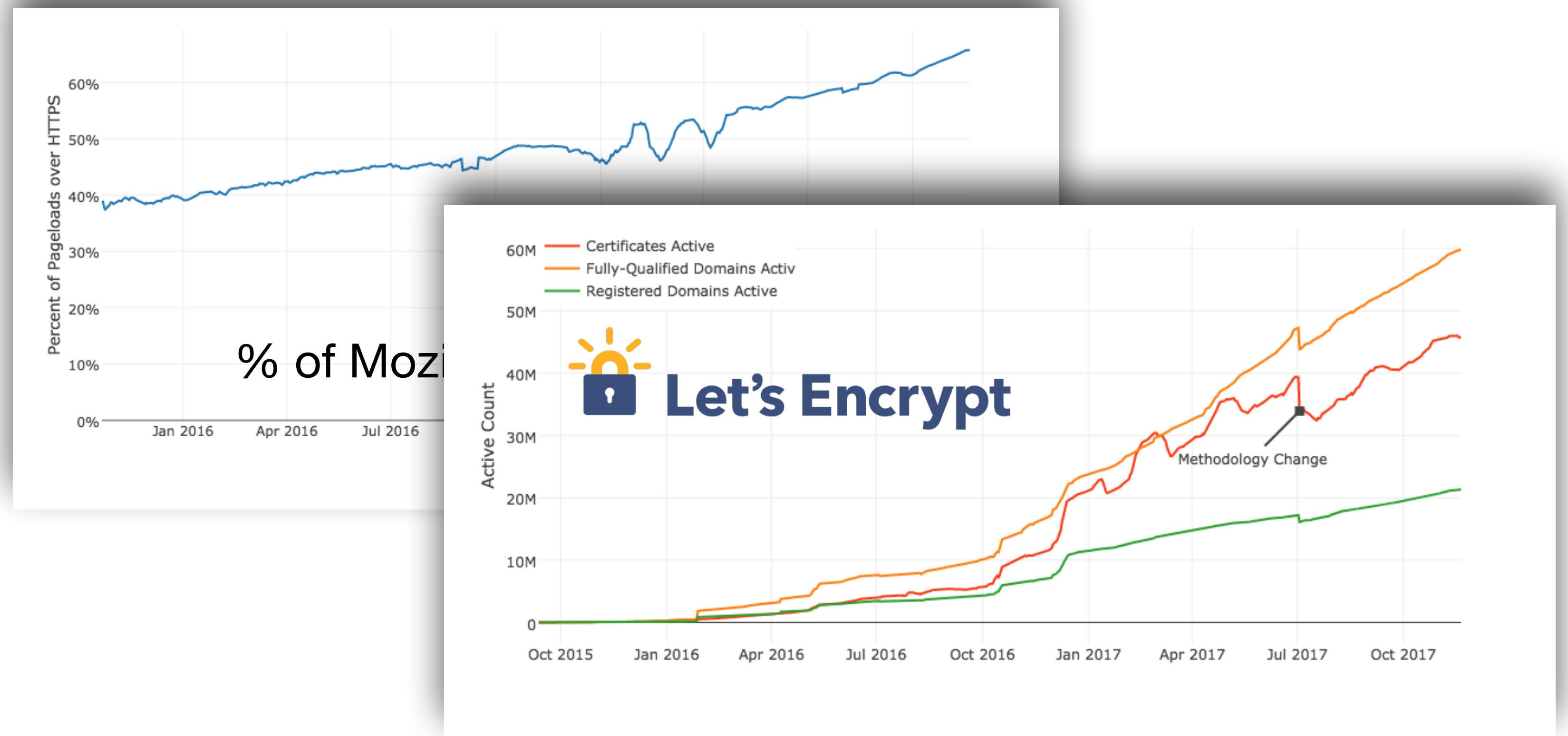
mami
measurement and architecture for a middleboxed internet

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..

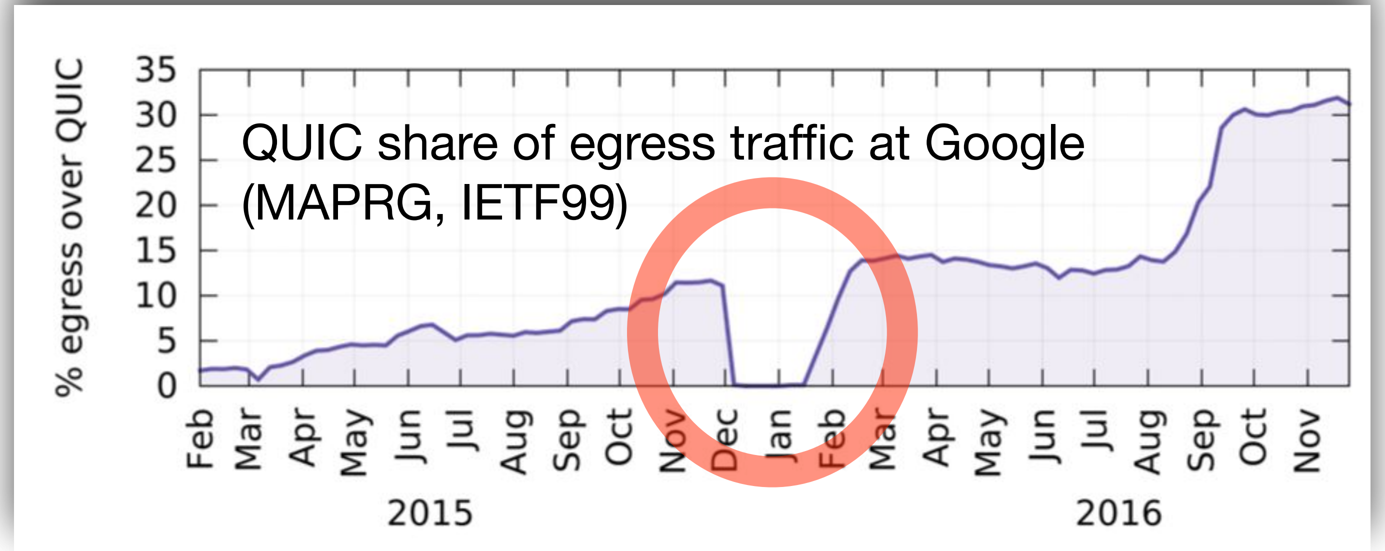
Supported by the Swiss State Secretariat for Education, Research and Innovation under contract number 15.0268. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the Swiss Government.

Increasing Deployment of Encryption



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

Protocol Stack Encryption



- QUIC: new, UDP-encapsulated transport, optimized for HTTP/2
- Developed/deployed by Google, 7% of Internet traffic end-2016.
- Under standardization in the IETF, expected deployments 2019.
- QUIC ***encrypts everything*** not needed to establish communication and forward packets.
- → Nothing that uses TCP headers will work on the new Internet, either.

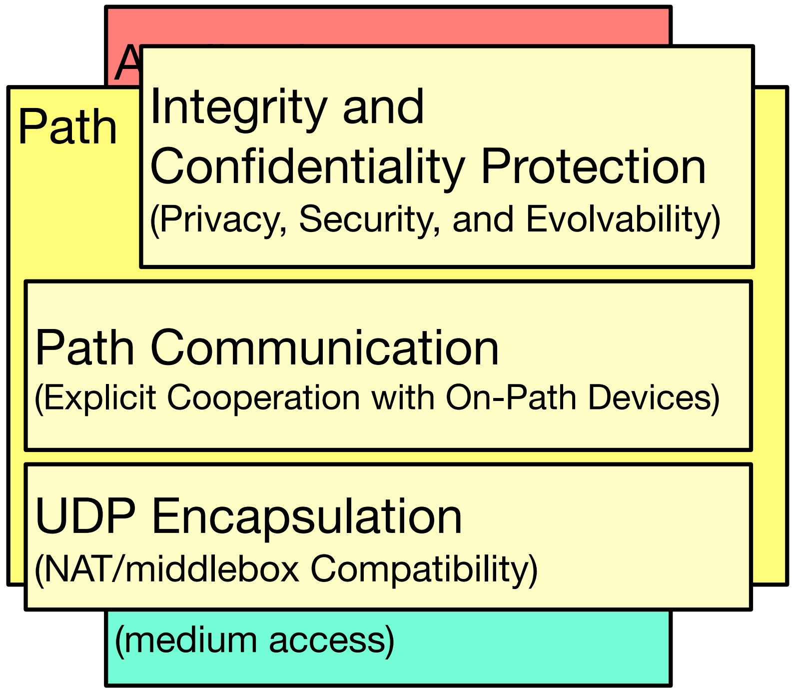
Explicit Cooperation

- 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.



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

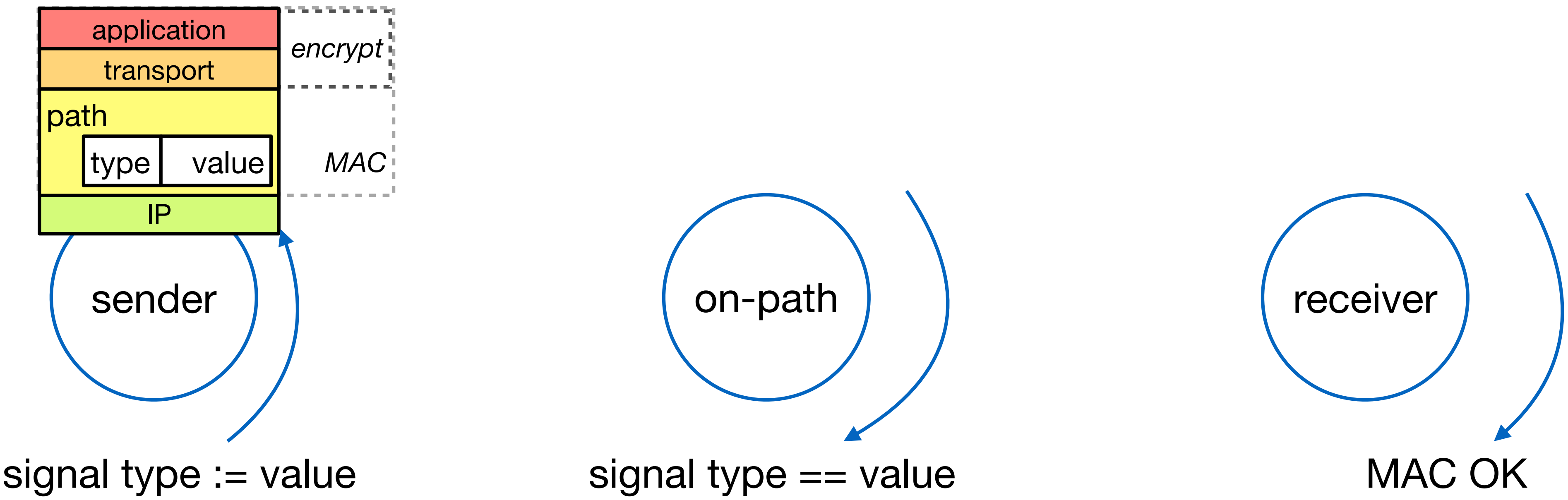
- Transport-Independent On-Path State
- Latency Measurement
- Loss and Congestion Measurement
- Path Trace Accumulation

} Today's talk

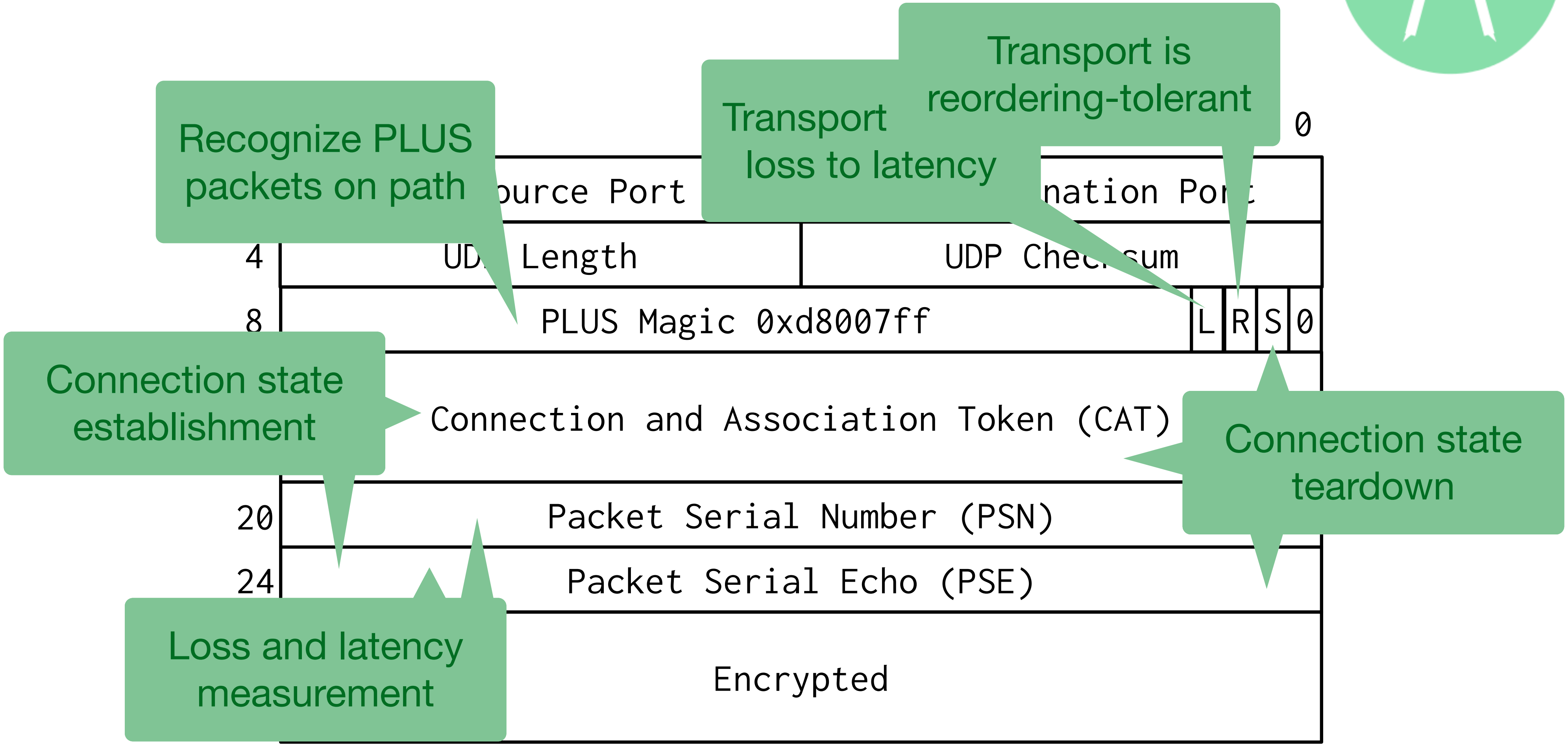
- Loss/Latency Tradeoff
- Path MTU Discovery

- Generic mechanism allows for future extensibility

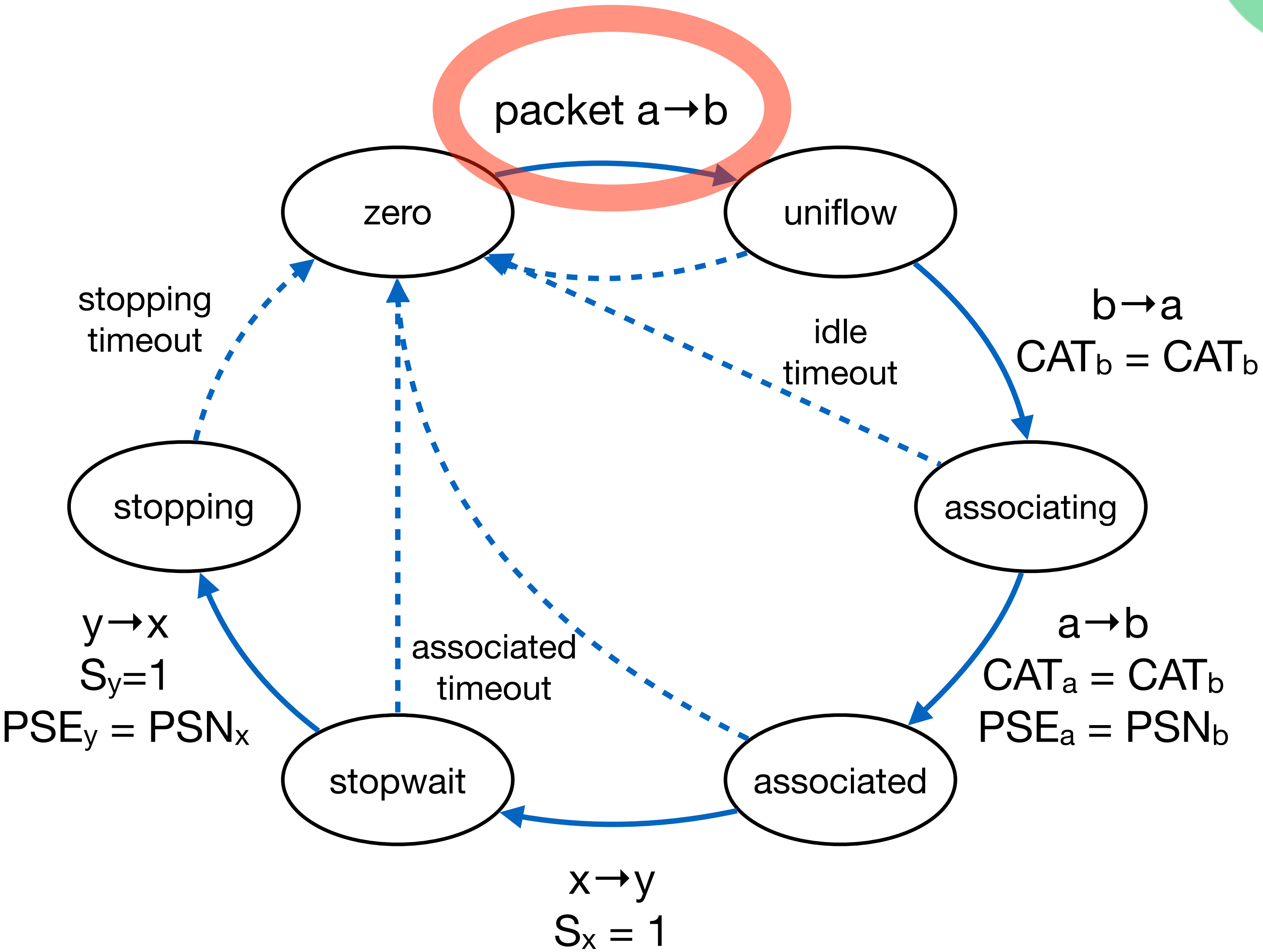
Sender to Path Signaling



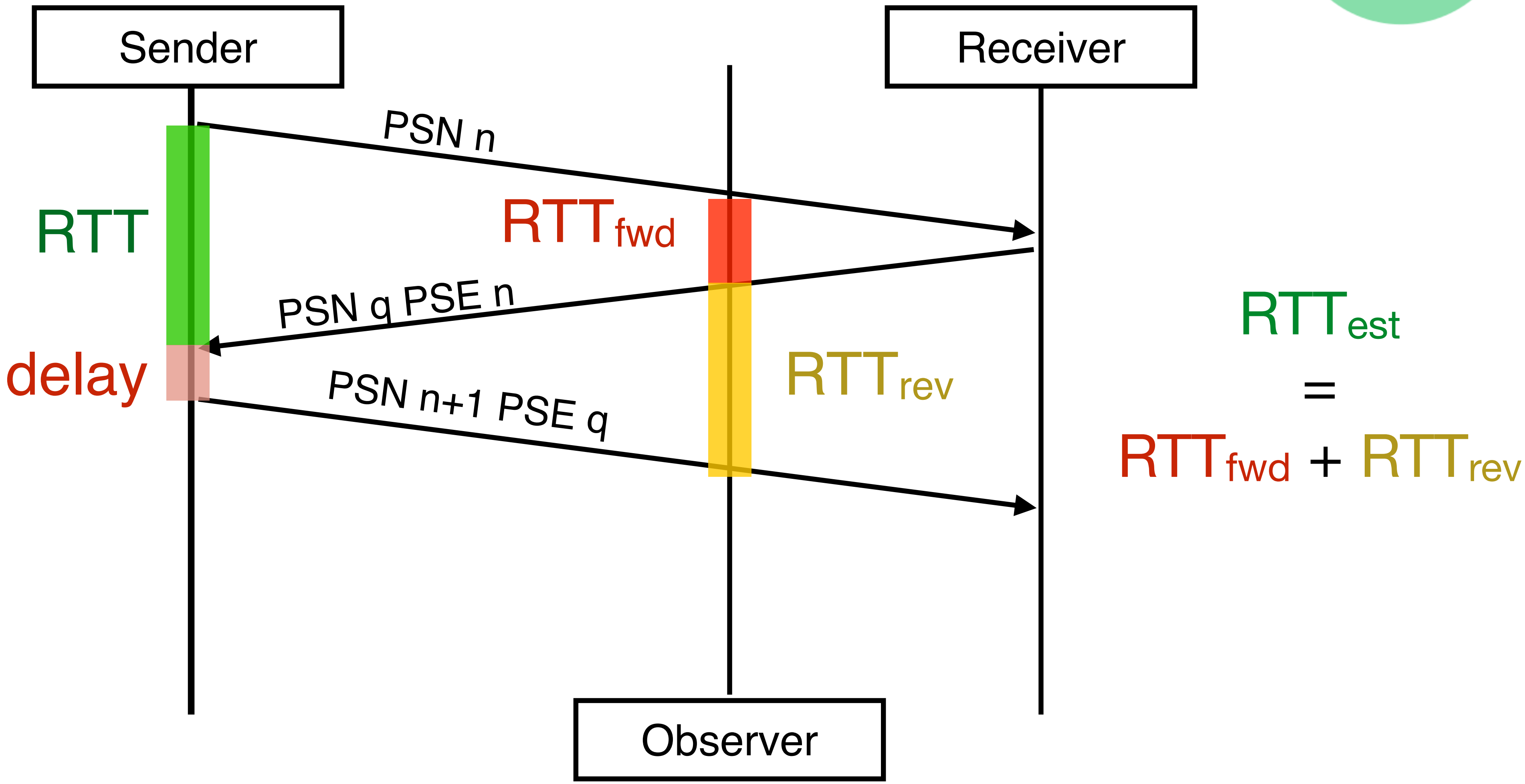
Basic PLUS Header



Transport-Independent On-Path State

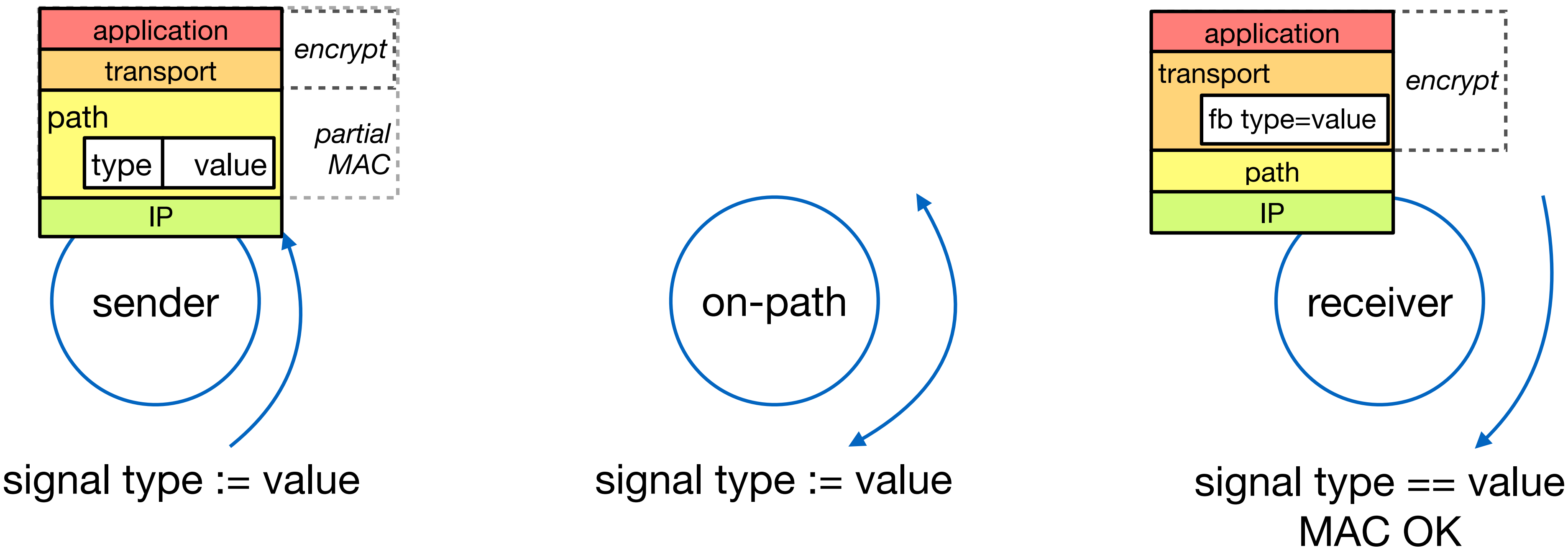


Latency Measurement

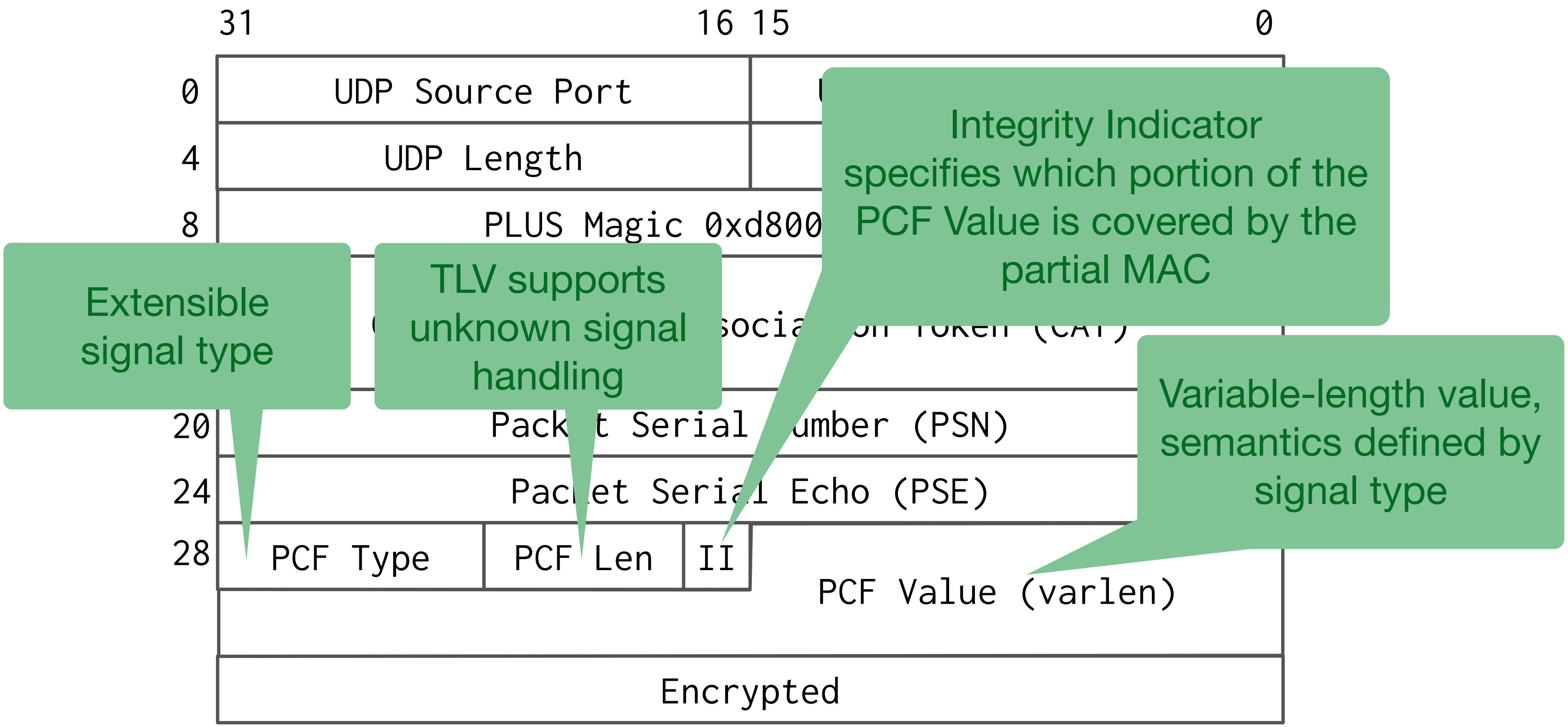


- PSN/PSE are explicit measurement signals replacing TCP SEQ/ACK + TSOPT

Path to Receiver Signaling with Feedback



Extended PLUS Header



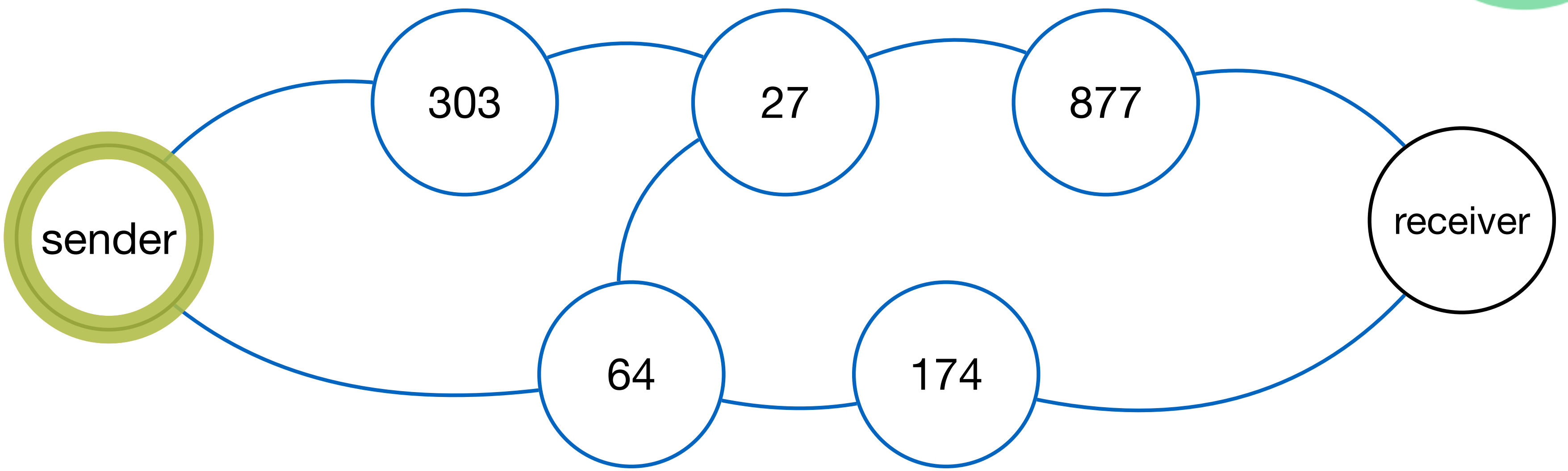
Loss and Congestion Measurement

- 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: 11(full)
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.

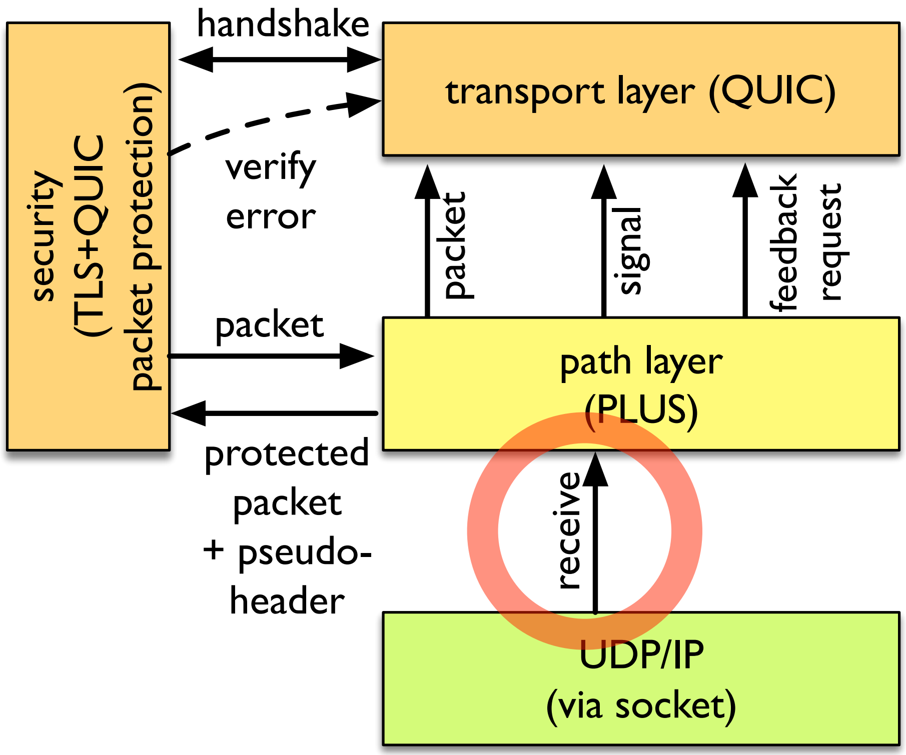
Path Tracing



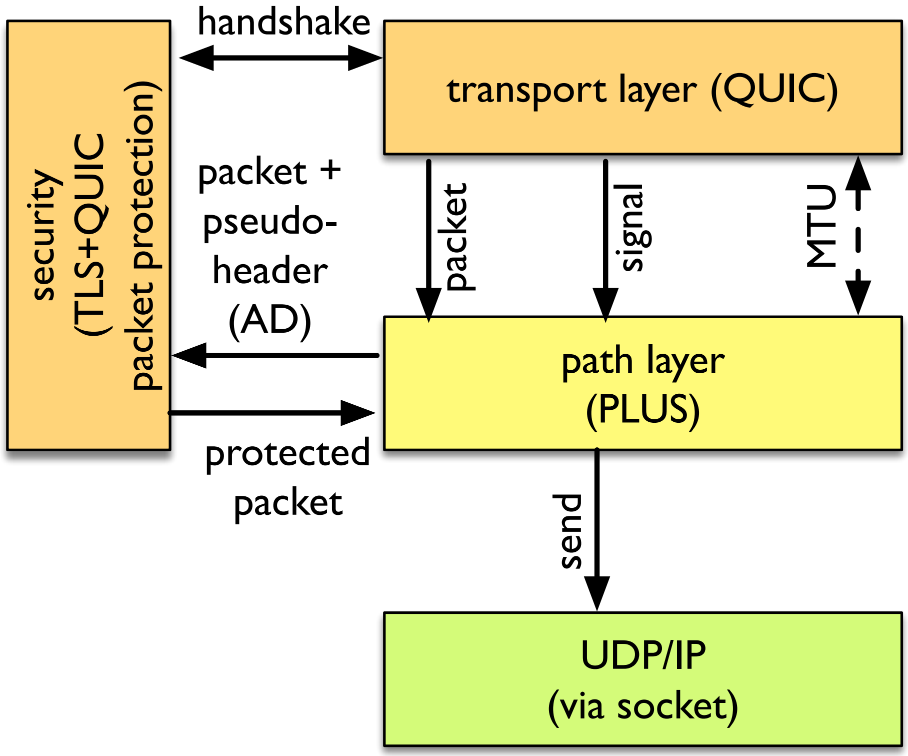
- Each PLUS-aware hop XORs random value per node to PCF type 4 value.
- Value at receiver indicates which path was taken without identifying path.

- Red path: 1207
- Orange path: 238
- Green path: 968

Transport interfaces to PLUS: pilot implementation work under QUIC



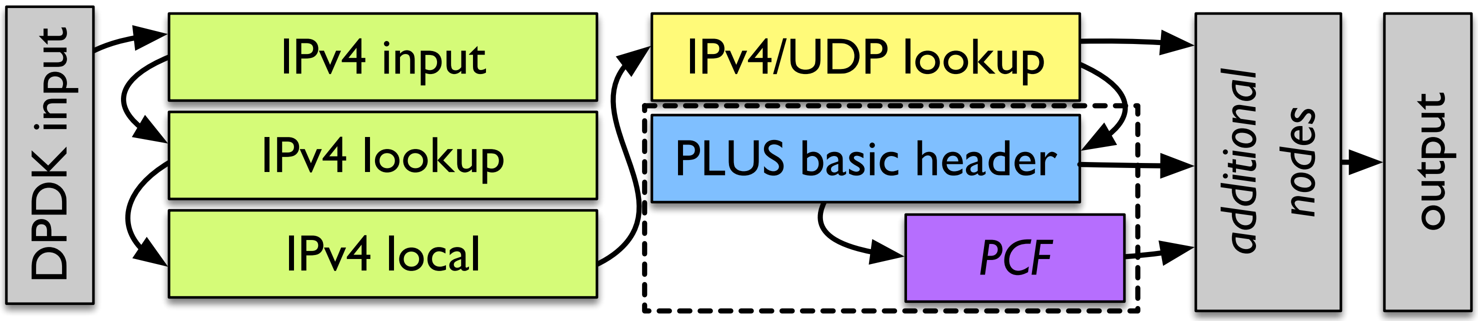
(a) receiver-side interfaces



(b) sender-side interfaces

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 proposed to replace it for passive RTT

Conclusions

- Adding a ***path layer*** to the Internet architecture to enable ***explicit cooperation*** between endpoints and middleboxes can replace manageability and measurability lost through encryption.

- PLUS provides a testbed for experimenting with explicit cooperation approaches.