WP3 Middlebox Cooperation

Gorry Fairhurst WP3 Lead
2nd Technical review
3rd October 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..

Objectives

- Definition of **use cases and requirements** for an architecture for Middlebox Cooperation Protocol (MCP)
- **Design, implementation, and initial testing** of the MCP to provide an information exchange between end hosts and middleboxes
- Design of a **flexible transport stack (FTL)** to complement the MCP, restoring connectivity over the Internet
- **Threat and trust analysis** of the developed protocols, protocol extensions and transport layer mechanisms as a basis for Internet-scale deployment

Overview - Who does what?

Partner	MM	Task 3.1 Use Cases	Task 3.2 MCP Design	Task 3.3 FTL Design	Task 3.4 Implementation and Testing	Task 3.5 Threat and Trust Analysis
ETH	18	✓	✓	✓	✓	
TID	10	✓	✓			✓
UoA	12		✓	✓		
ZHAW	18	✓			✓	✓
ALU (Nokia)	6	✓		✓	✓	✓

WP3 Protocol Design and Implementation

- MCP Design → Path Layer UDP Substrate (PLUS)
- FTL Design
  - Transport Protocol Feature/Interface Analysis
  - Post Sockets abstract API
- Implementation
  - Integrate PLUS with QUIC on endpoints
  - FD.io middlebox pilot for measurement
- Red-team threat analysis of middlebox cooperation schemes

PLUS Design Goals

- **Sender-to-path signaling:** An endpoint should be able to explicitly expose any signals used by on-path devices.
- **Path-to- receiver signaling:** An endpoint should be able to request signals from devices on the path.
- **End-to-end integrity protection:** An on-path device should not be able to forge, change, or remove a signal sent by an endpoint.
- **Integrity protection over a scratch space:** An endpoint controls signaling between endpoints and the path, or from one on-path device to another.
- Does not assume a**uthentication of signals from on-path devices:** Possible to request and receive signals from a previously unknown on-path device.
- **Robust to attack**: The mechanism should present no significant surface for amplification attacks.

MCP Specifications: IETF Documents

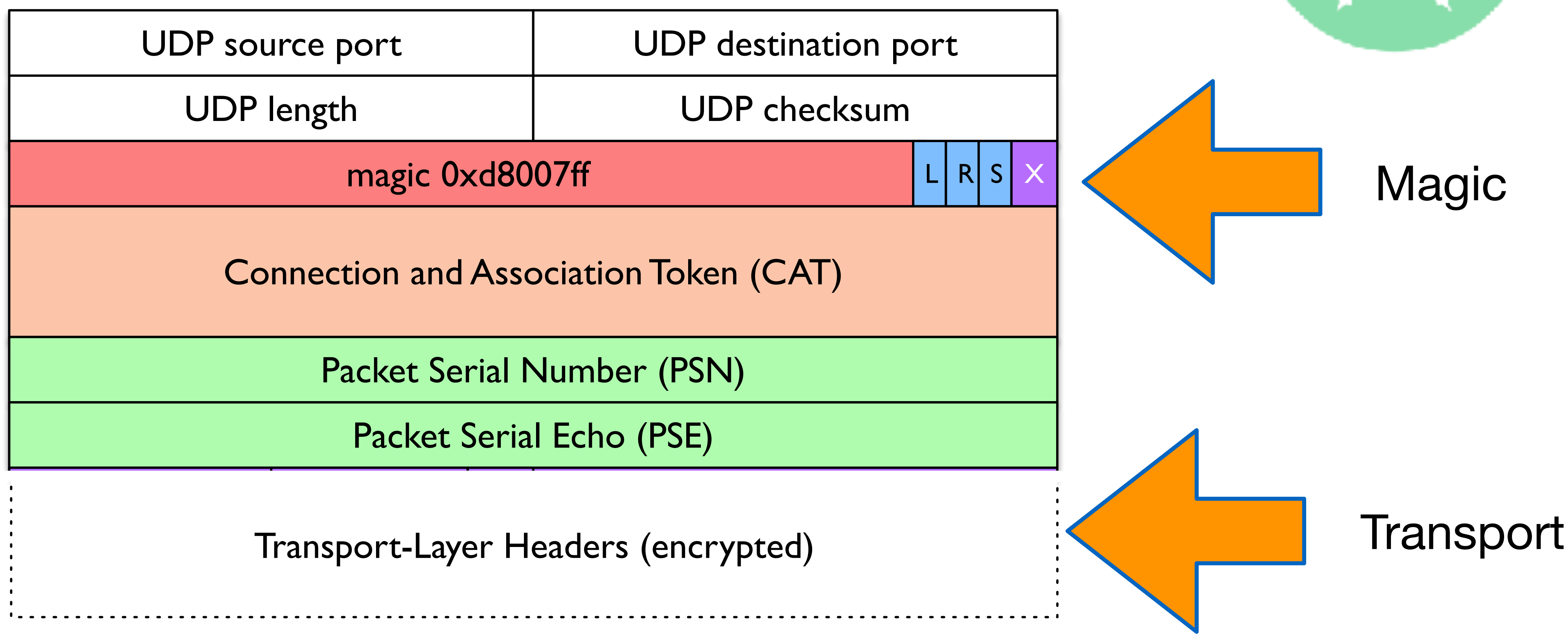
Inputs to PLUS design

- draft-trammell-spud-req
- draft-trammell-plus-abstract-mech
- draft-trammell-plus-statefulness
- draft-trammell-plus-spec

Other higher-layer middlebox cooperation mechanisms

- draft-ietf-acme-star

PLUS Basic Header Format



Header on top of UDP identified by magic number

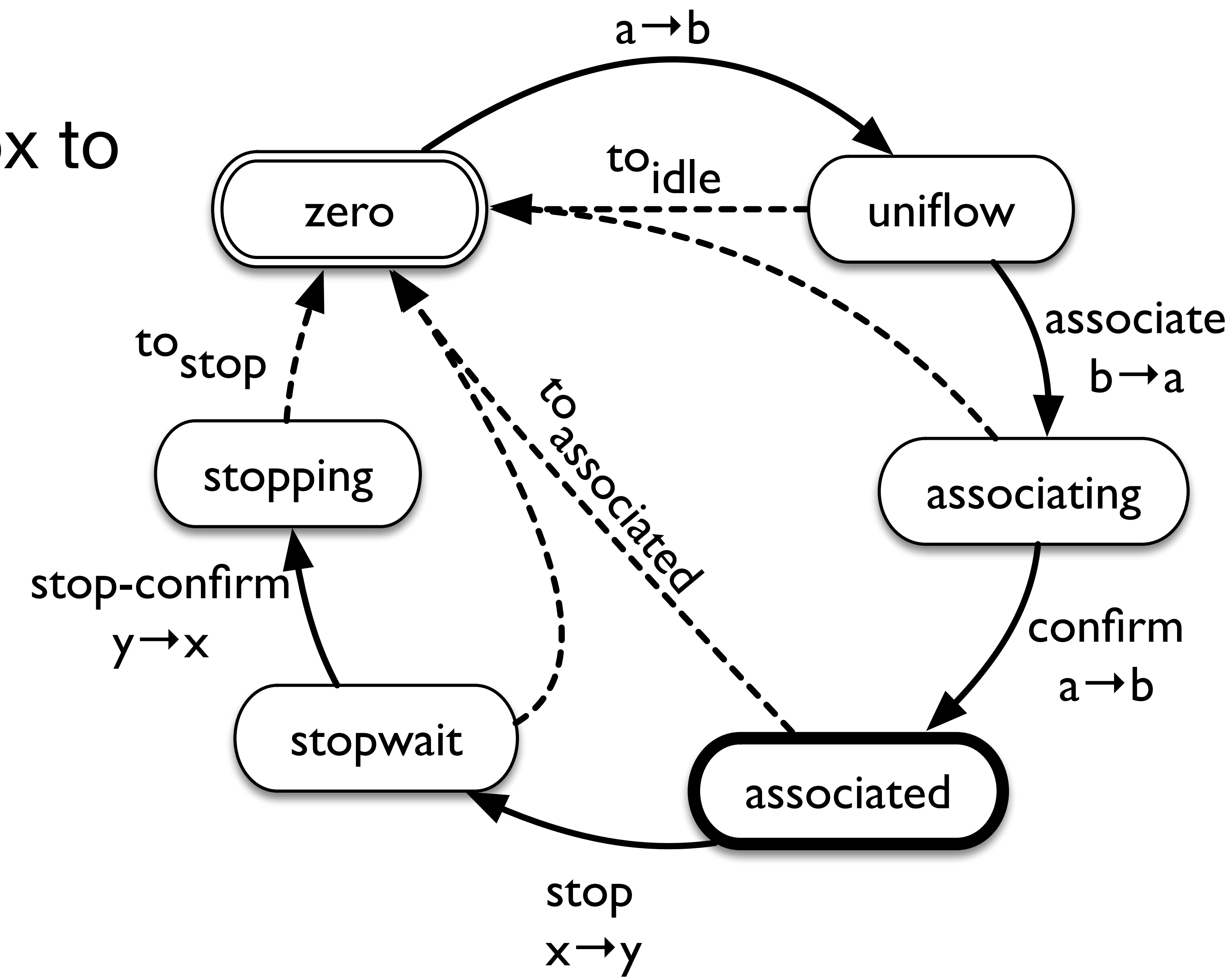
Basic and extended formats

L= Lola, R= May be reordered, S= Stop

Provides a common wire image for encrypted transports

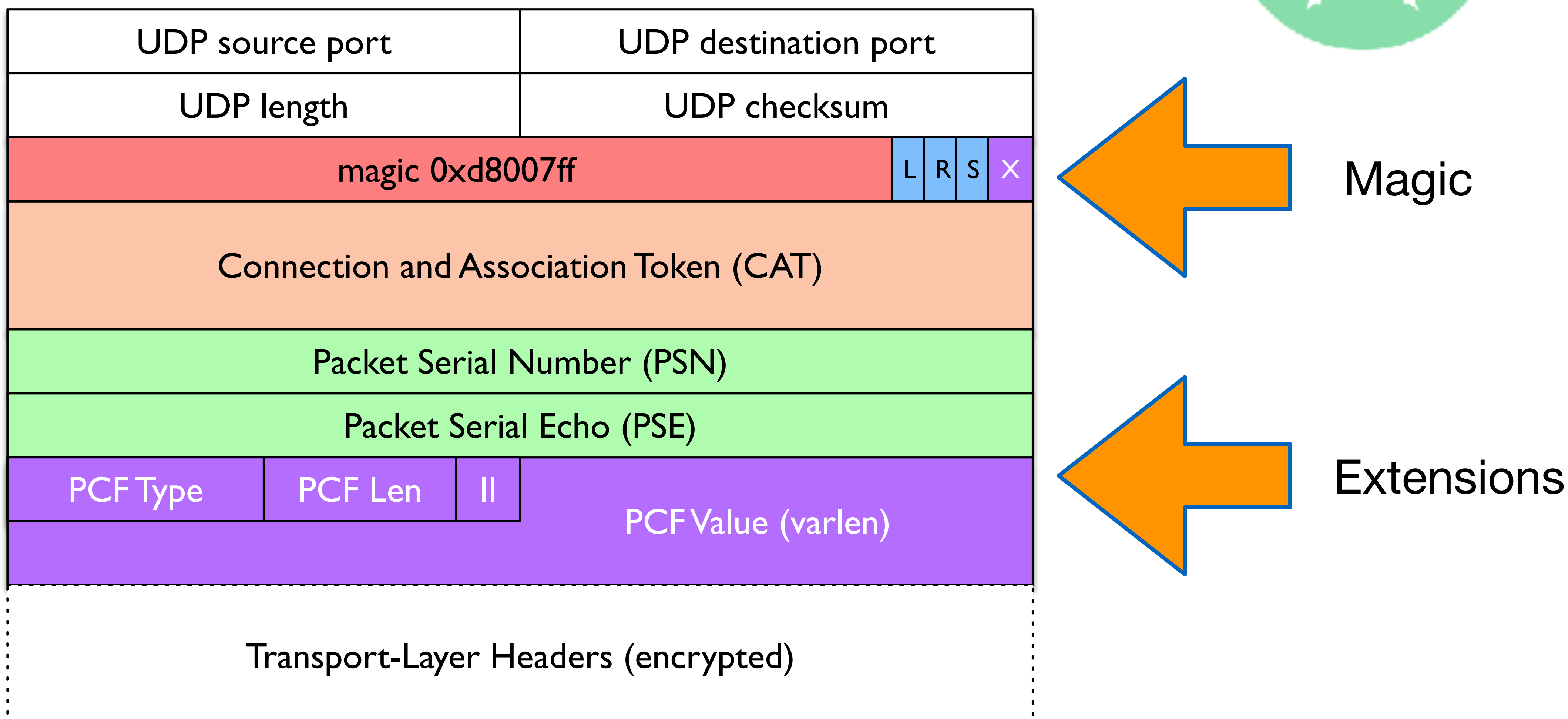
A transport-independent on-path state machine

- Enables a middlebox to track the flow state
- e.g. NAT/Firewall



draft-trammell-plus-statefulness

PLUS Extended Header Format



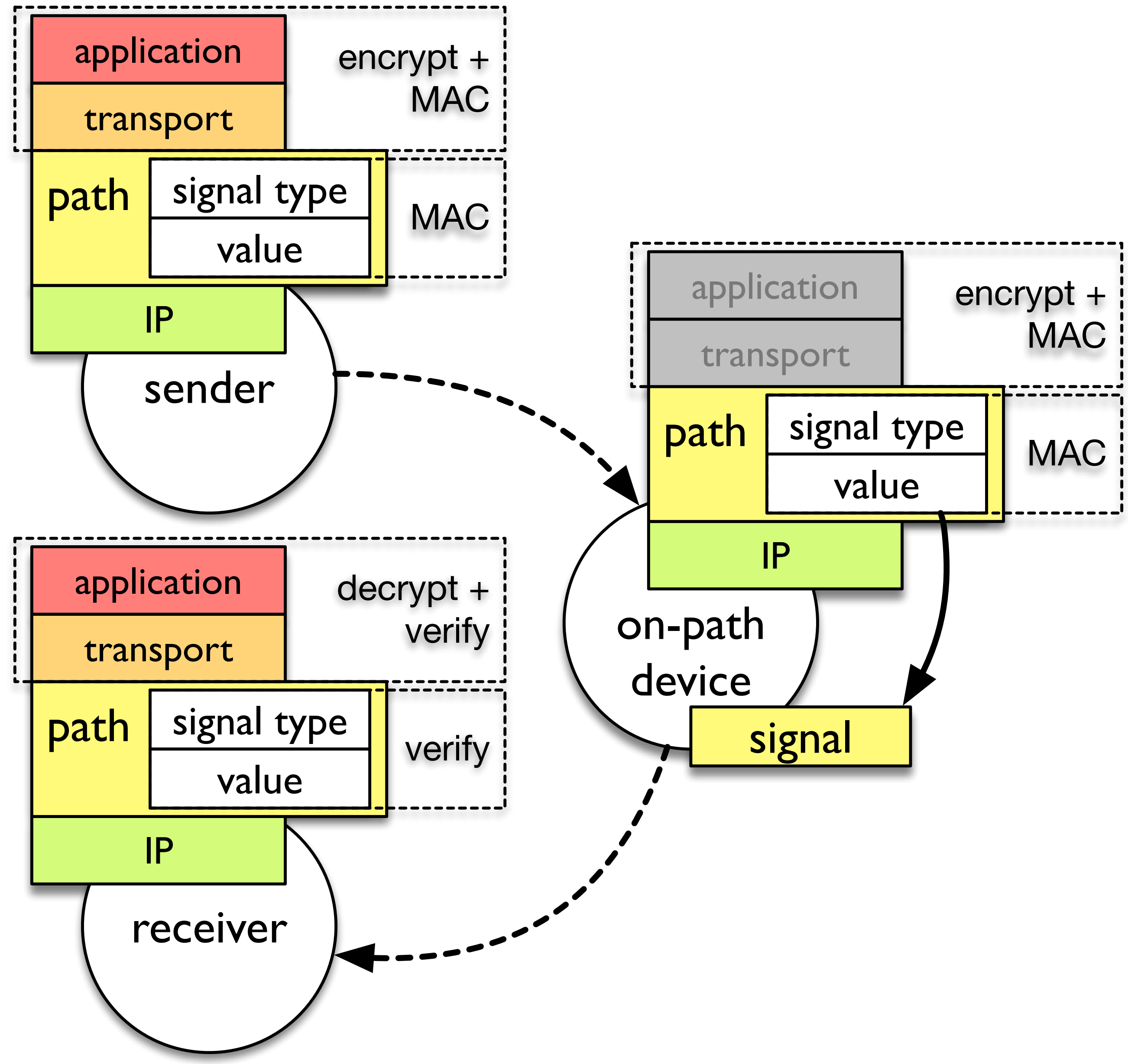
X=1

Each PLUS packet can carry only one PCF at a time

Sender decides which PCF is supported in a packet

Transport-independent in-band signaling: Sender to Path Signal

- Unencrypted signal
- Integrity protection
- Path can not verify
- Receiver may verify



L: LoLa

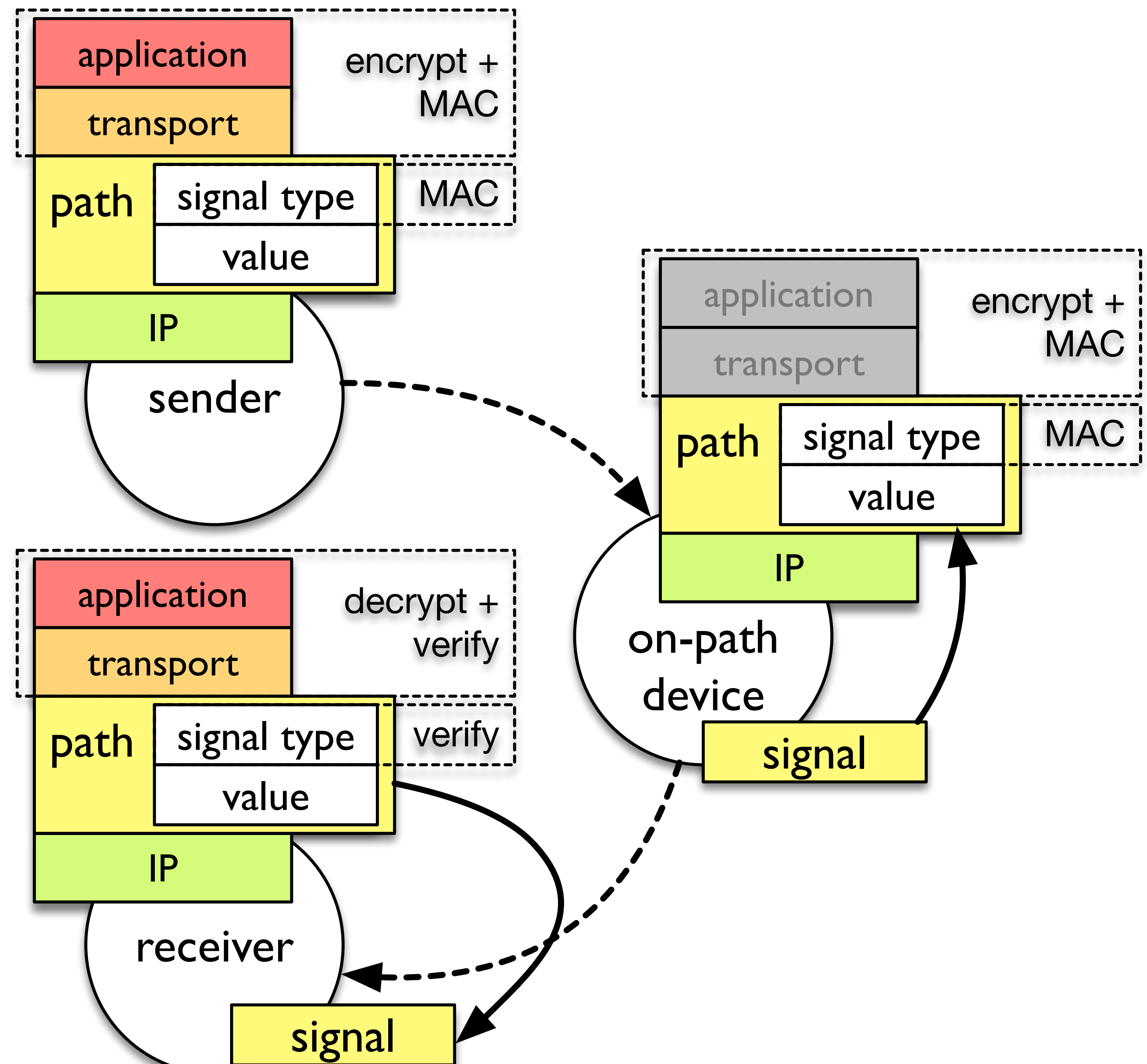
R: Reordering

S: Start of Session

PCF 1: Loss/Congestion Exposure

Transport-independent in-band signaling: Path to Receiver Signal

- Sender enables
- Unencrypted signal
- No integrity protection
- Use of info advisory



PCF 2: PMTU

PCF 3: Path tracing

MCP throughput guidance

Status of MCP Specifications

Initial specification contributed to IETF

Feedback received and design refined

MCP/PLUS design is being finalised

Rationale for the design published in CNSM 2017

Final version will be in task deliverable (M24→M30)

Principles for Measurement in Protocol Design
[ACM CCR April 2017; Best of CCR 2017]

A Path Layer for the Internet: Enabling Network Operations on Encrypted Protocols

[IEEE/IFIP CNSM 2017]

PLUS and QUIC in IETF

- PLUS (MCP) work is currently stagnating in the IETF
  - Concerns that a generic metadata exposure protocol could be used to force metadata injection on endpoints
- Google proposed a new protocol (QUIC)
  - Work adopted as an IETF activity in 2017
  - All energy in transport/web space going into QUIC, which will actually deploy at scale in the near term (2018)

PLUS and QUIC in MAMI

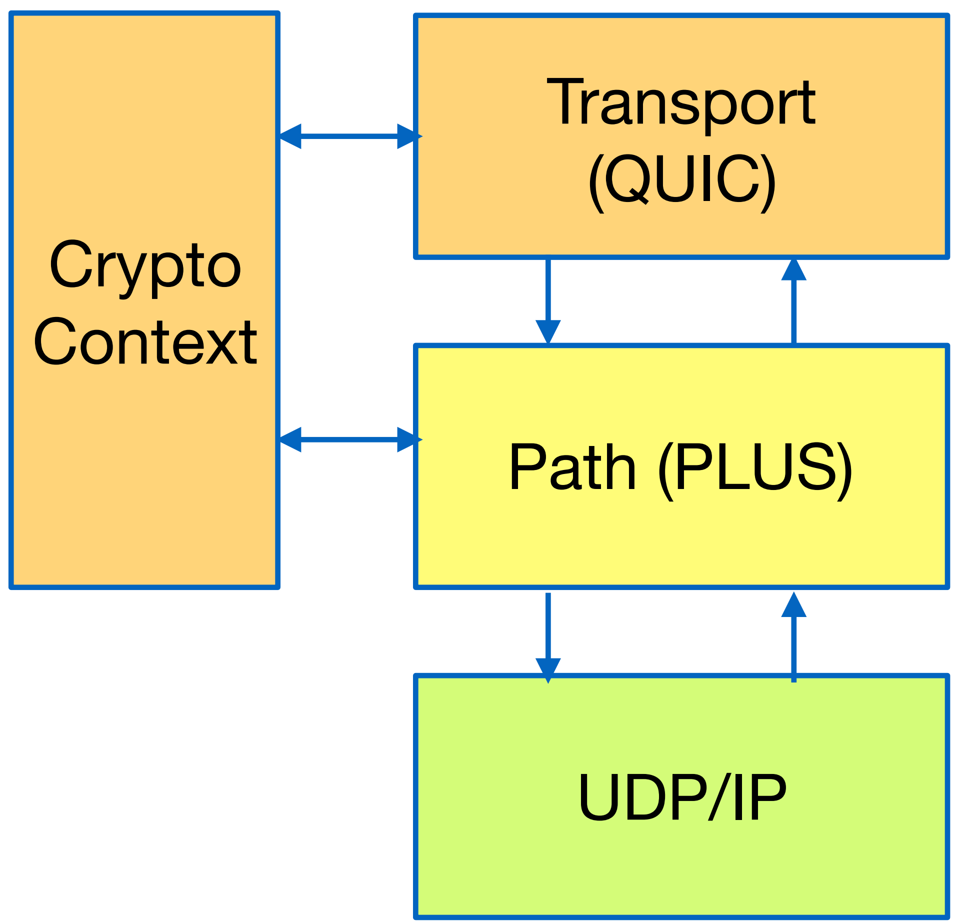
MAMI decided to focus on using PLUS mechanisms in QUIC

Editing applicability and manageability documents for QUIC

- draft-ietf-quic-manageability
- draft-ietf-quic-applicability

Ensure lessons learned from PLUS can be applied to QUIC protocol features (e.g. passive measurability)

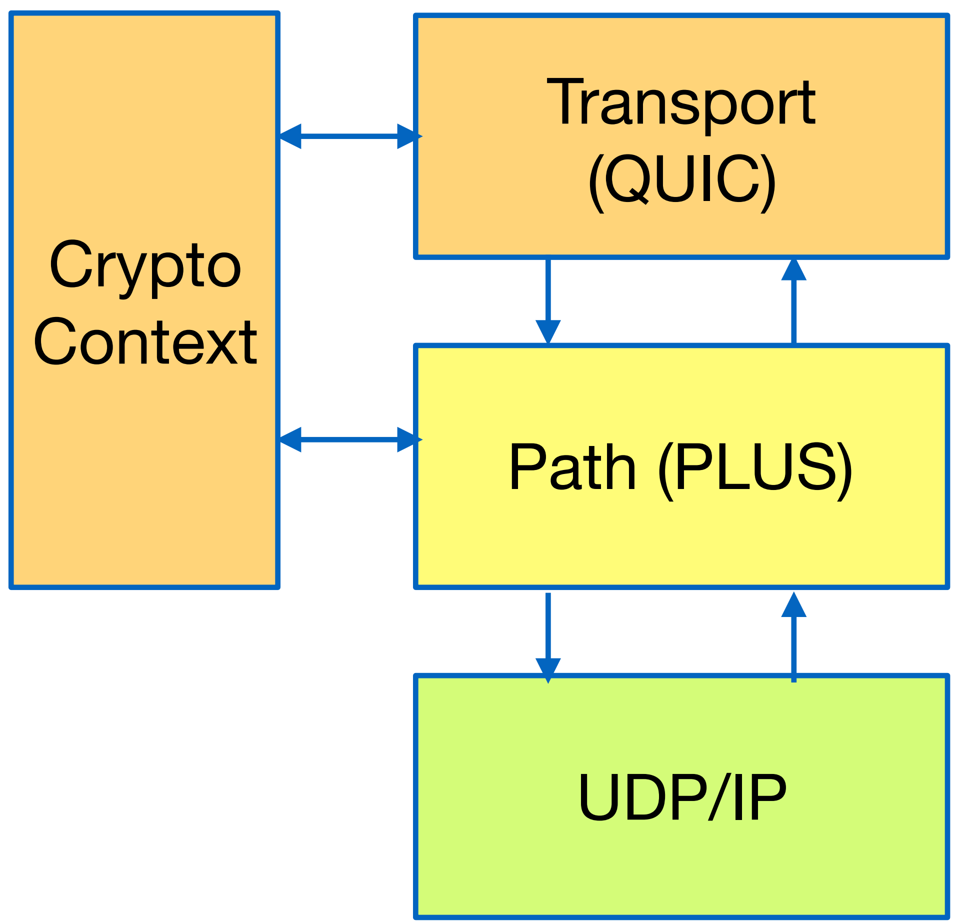
Implementation of MCP for QUIC



PLUS QUIC Stack

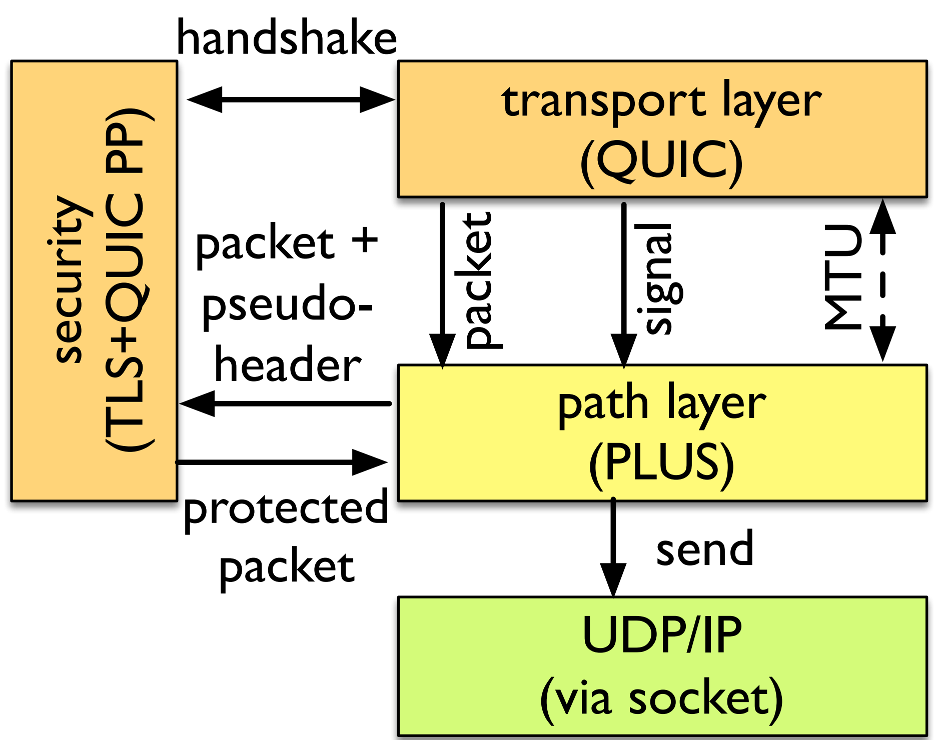
- PLUS and transport are coupled
- Crypto context needed for authentication of PLUS header material
- Transport provides: feedback channels, flags

~~QUIC~~ GUIC

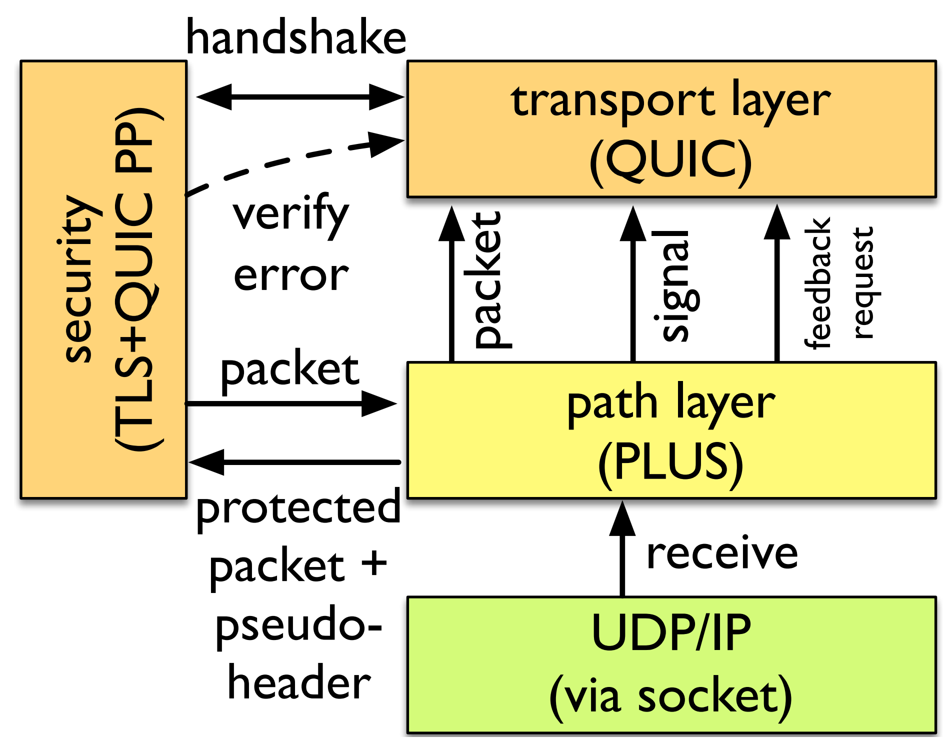


- QUIC spec expected Nov 2017
- Reference implementation by End of 2017
- At the moment google's QUIC (GUIC) is the best we have for experimentation.

PLUS API Design



PLUS Packet Transmission



PLUS Packet Reception

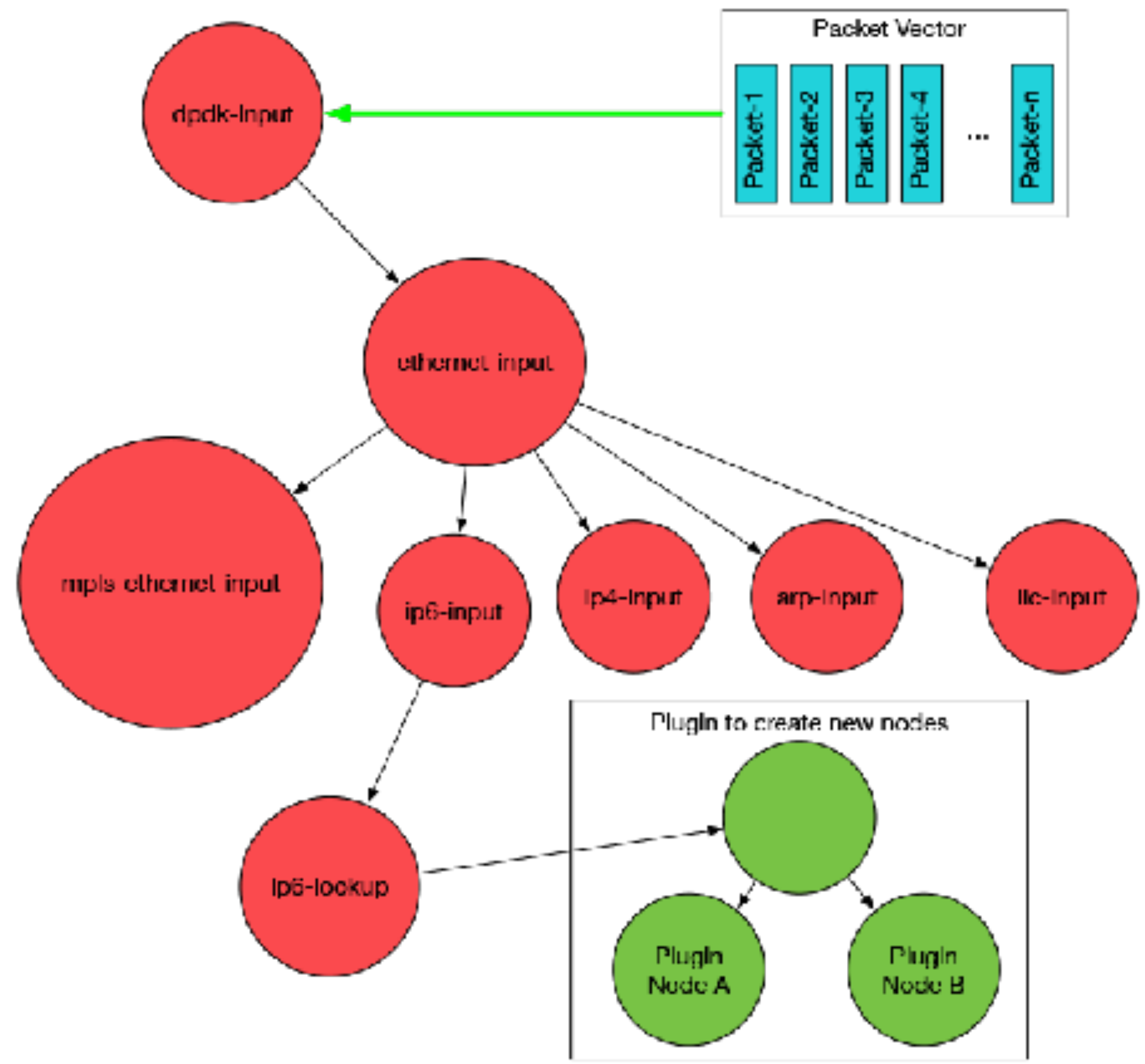
PLUS Tools

- **plus-pcap**
  - Support added to `gopacket` to decode PLUS packets
  - Command line tool `plus-pcap` using gopcap/gopacket
  - Reads PCAP files containing PLUS packets or dumps traffic live
  - Outputs JSON

- **pluspector**
  - Command line tool for debugging
  - Relay connections (does not use raw sockets), generate and echo packets
  - Simple fuzzing: can generate packets and randomly modify bytes
  - NOT a tool to debug transport protocols using PLUS

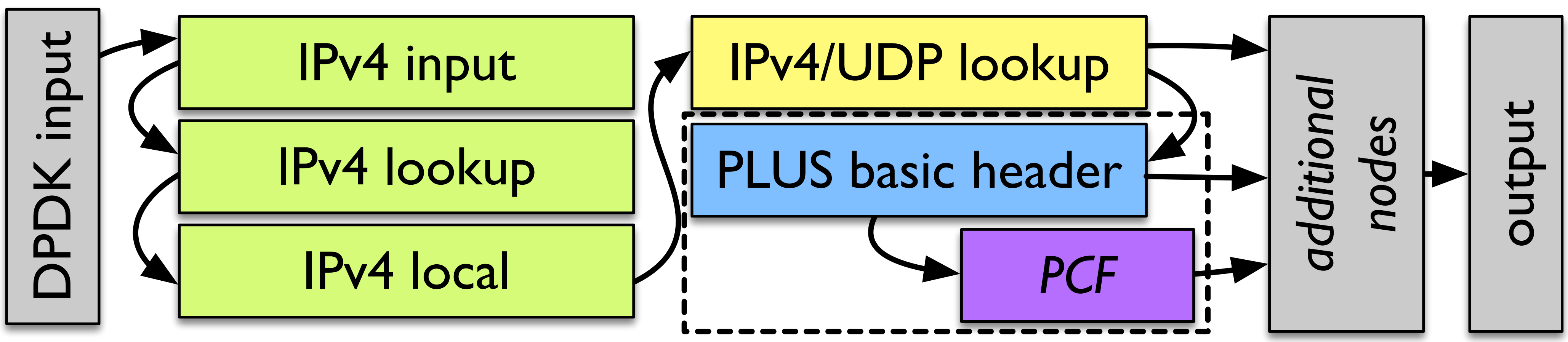
fd.io Architecture

- Shared memory message bus
- Very **high performance low level API**
- Messages passed along the bus are specified in a simple Interface Definition Language, which is used to create *C client libraries* and *Java client libraries*



mami

VPP MCP Implementation



PLUS is a substrate

- UDP lookup must check magic as well as ports

Basic header handles state machine

- One plugin node per PCF, since only one PCF per packet

Red-Team Analysis of MCP/PLUS

PLUS currently stagnates in IETF, due to privacy concerns

- PLUS explicitly exposes metadata
- Does PLUS help mass surveillance?

Goal: contrast metadata exposure from:

- PLUS (and encrypted transport), versus
- TCP (and encrypted application-layer data, e.g., TLS)

Security Analysis Documents

Related IETF contributions from MAMI:

draft-trammell-privsec-defeating-tcpip-meta

draft-fairhurst-tsvwg-transport-encrypt

RTT exposure privacy analysis to QUIC RTT DT:
github.com/britram/trilateration

Related IETF contributions from outside MAMI:

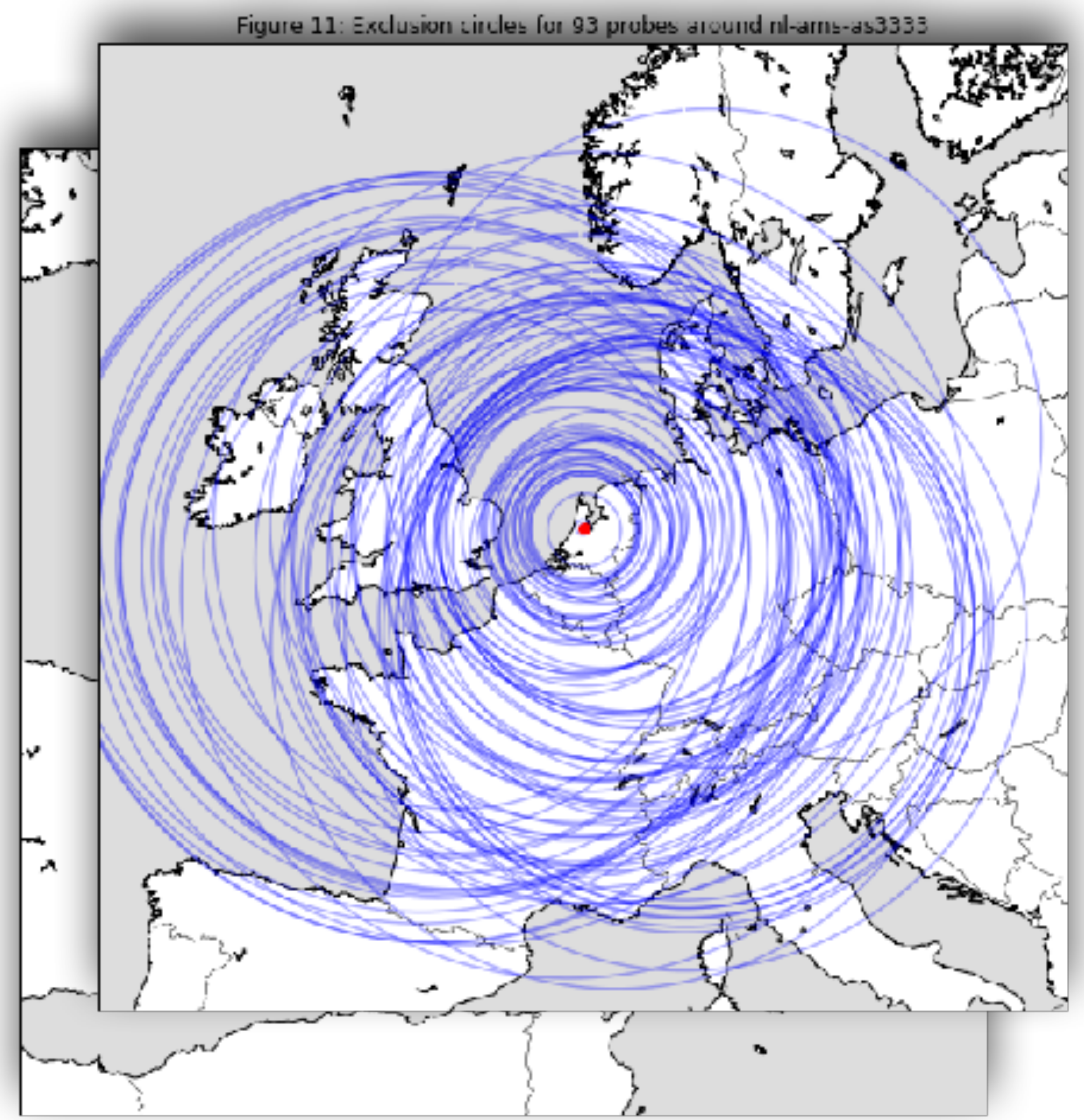
draft-mm-wg-effect-encrypt

draft-dolson-plus-middlebox-benefits

draft-iab-marnew-report

Is RTT exposure to the path a threat to geoprivacy?

No.



- min(rtt) from Atlas anchoring measurements, fiber lightspeed assumption

A Flexible Transport Layer (FTL)

- Selection of protocols based on a composite of transport protocol features
- Discovery of usability of protocols/features along a path
  - Fallback and connection racing mechanisms
- Definition of ***unified*** (abstract) API independent of protocol implementation selected

FTL Specifications: IETF Documents

API/transport state-of-the-art

IETF Transport Services [RFC8095]

draft-ietf-taps-transports-usage

draft-ietf-taps-transports-usage-udp

API/transport evolution

draft-trammell-taps-post-sockets

Toward a unified API:
a few insights about transport APIs

Applications deal in messages of arbitrary size

Message reception is ***inherently asynchronous***

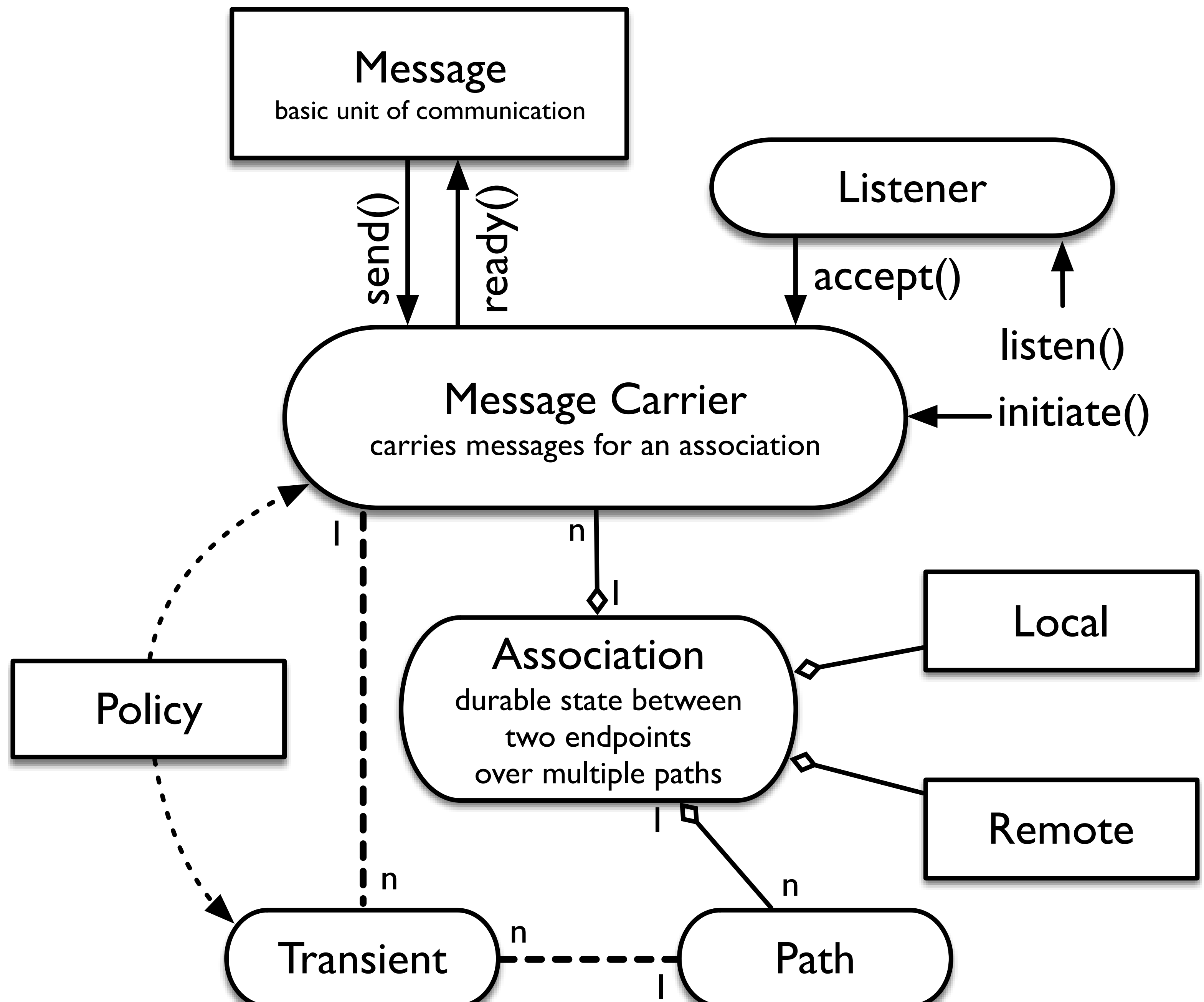
The network of the future is ***explicitly multipath***

Applications ***don't care about the transport layer***

Transport must ***guarantee security properties***

Post-Sockets

- A FTL needs an API to abstract away the cost of that flexibility
- Post Sockets provides this API based on insights derived from experience with sockets



WP3 Summary

- MCP specifications:
  - Input from standards contribution to PLUS
  - MCP specs being finalised
  - Next: broader focus on middlebox cooperation schemes
- Implementation
  - Reference software implementation: done
  - fd.io testbed: ready
    - fd.io node development ongoing
    - transition to experimentation in WP2
- Red Team Analysis: in progress