# *Implementing Ultra Ethernet Transport with XDP2*

Tom Herbert

XDPnet

# Overview

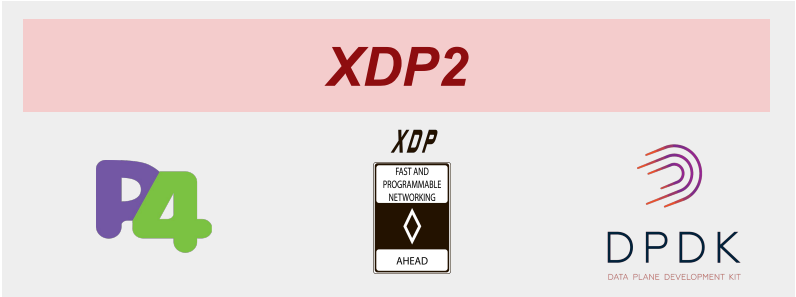
- XDP2 is a new open source programming model for the high performance SW and HW datapath
- The Ultra Ethernet Transport, or UET, is a new transport protocol designed expressly for AI/ML

XDP2 + Ultra Ethernet Consortium =

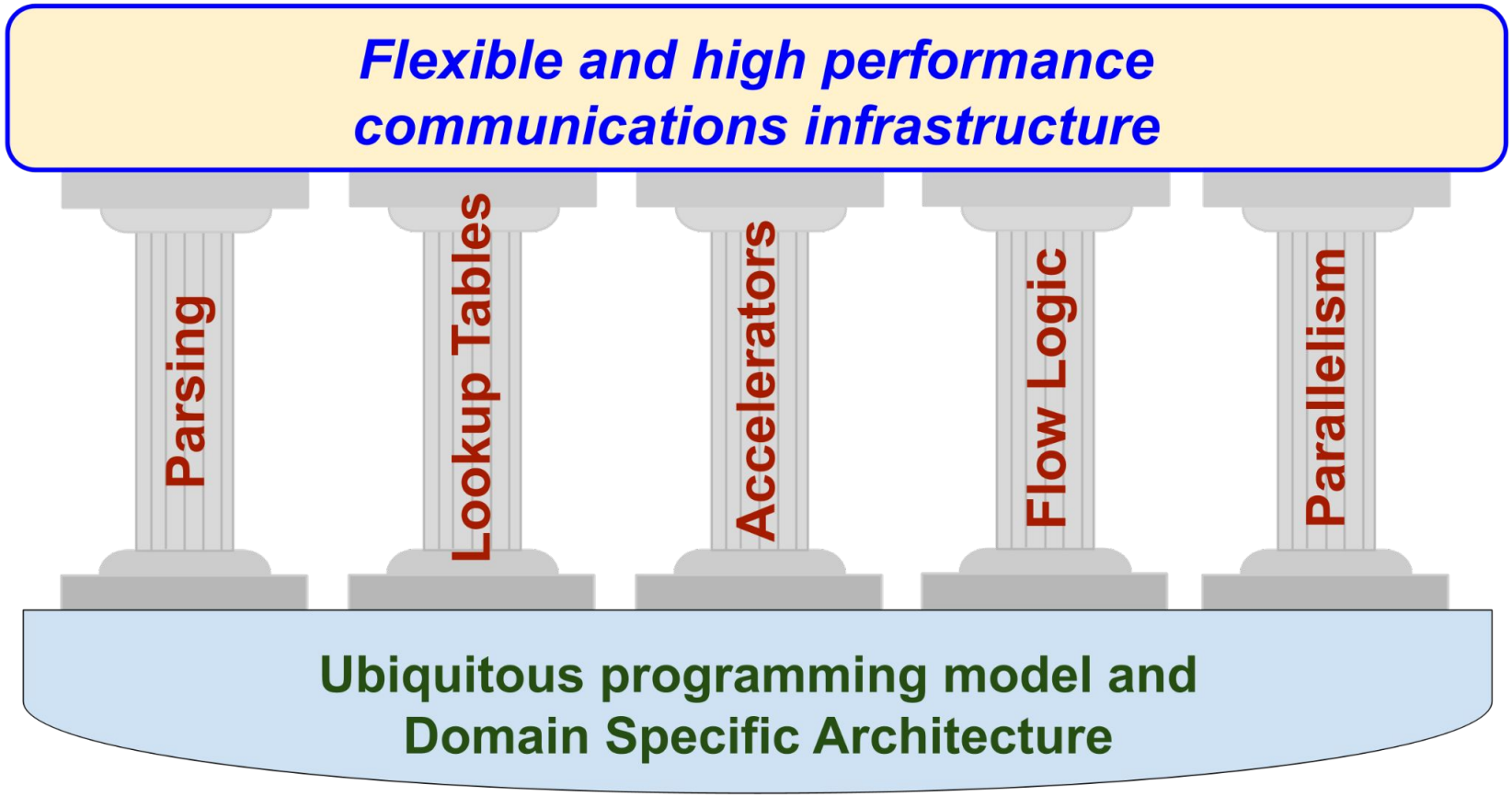
*Super fast and flexible open source UET implementation*

# *XDP2* is the programming model for the datapath

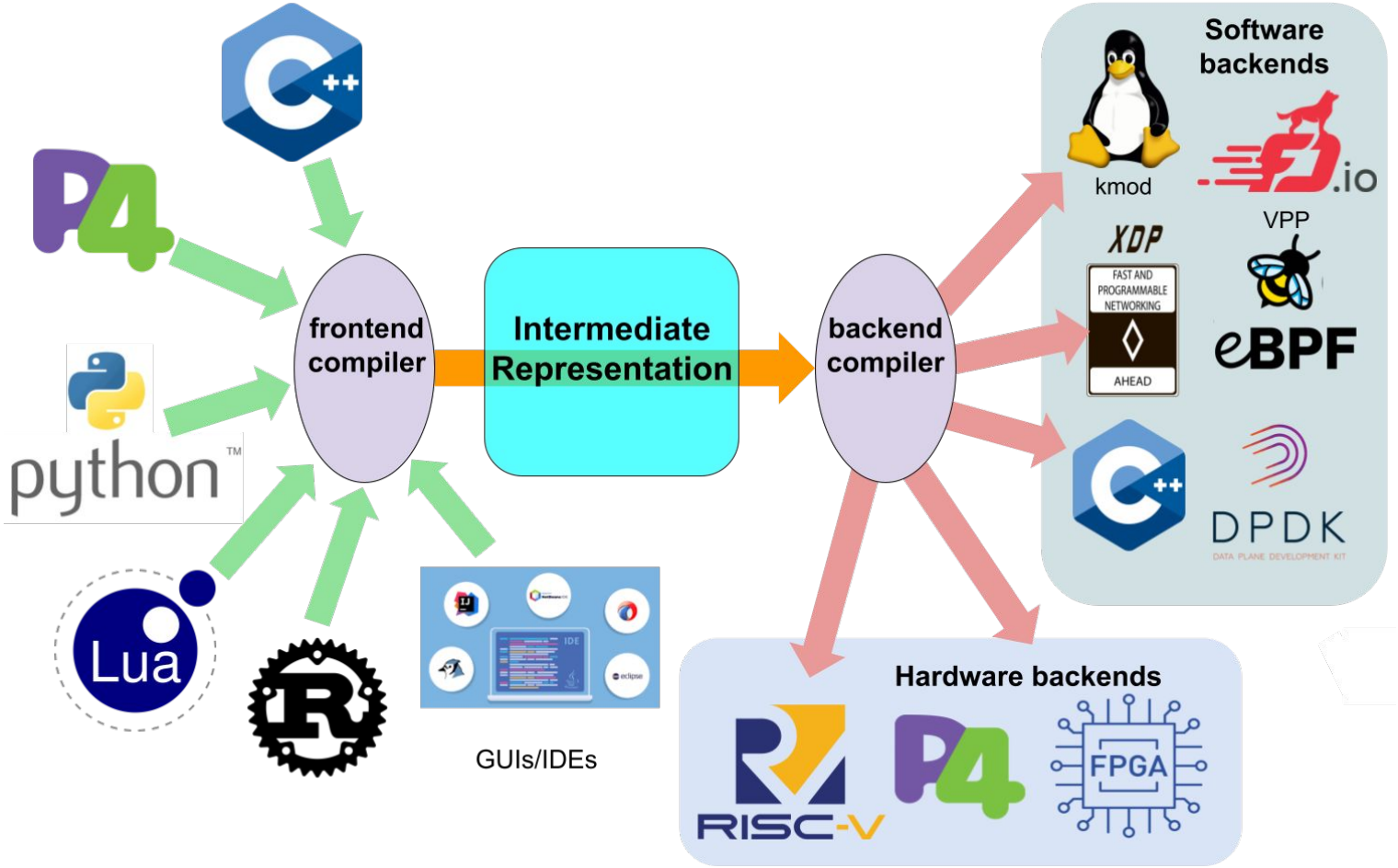
- User writes their datapath in a language that's convenient for them
- Their code compiles into a variety of target
- It's a type of convergence layer for things like P4, XDP/eBPF, DPDK/VPP



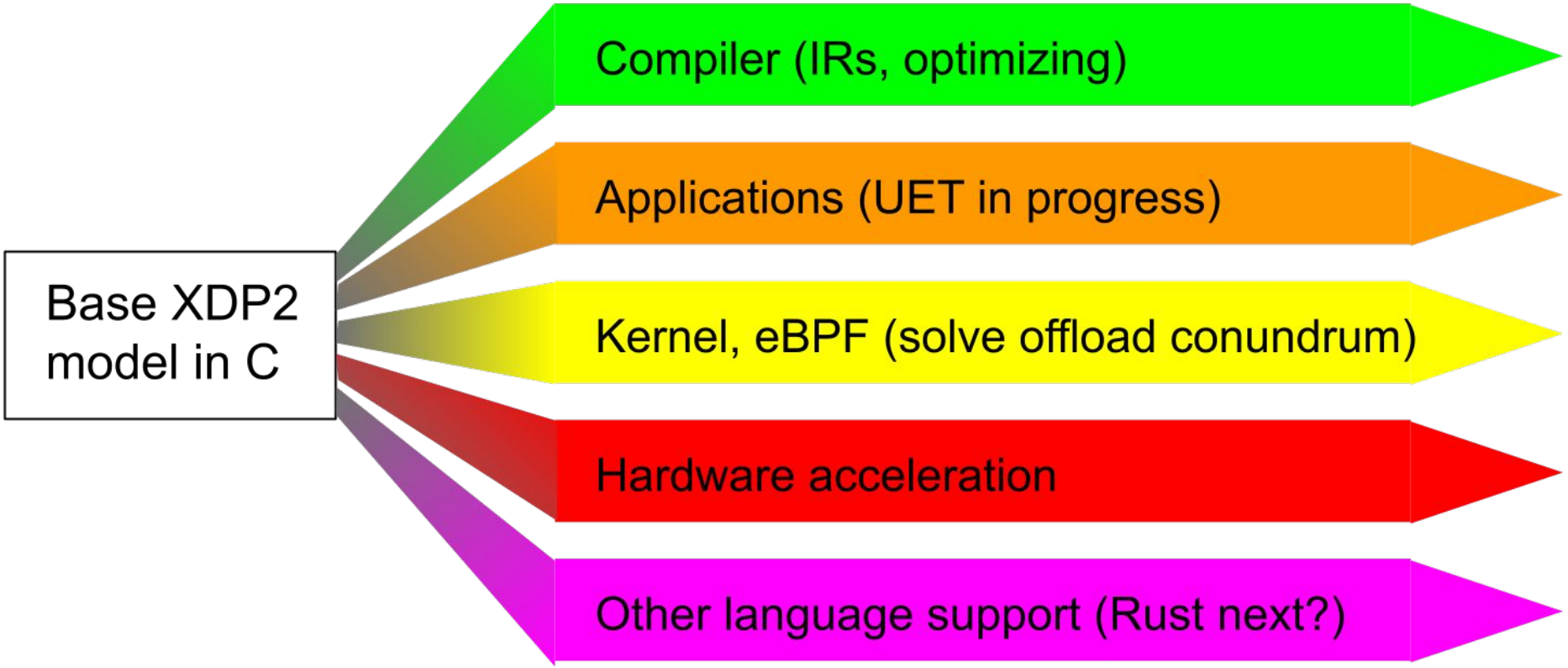
# The five pillars of the high performance datapath



Write once, run anywhere, run well!



# XDP2 development vectors



# Ultra Ethernet Transport (UET)

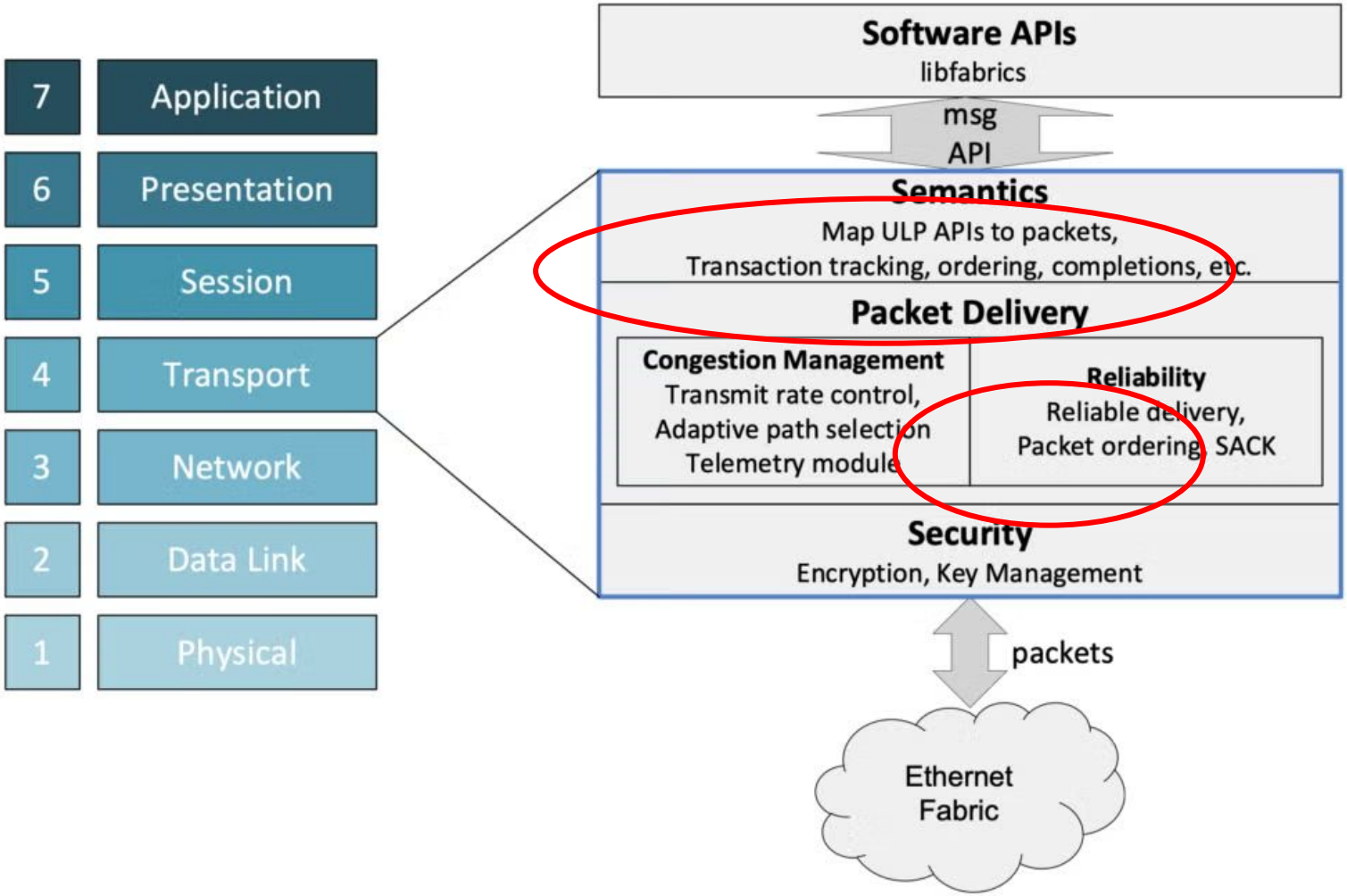
The **Ultra Ethernet Consortium** is developing a full-communications stack architecture to meet the growing network demands of AI & HPC at scale

**Ultra Ethernet Transport Layer**, or **UET**, is the protocol layer that provides reliable packet delivery and messaging semantics for Upper Layer Protocols

# UET features

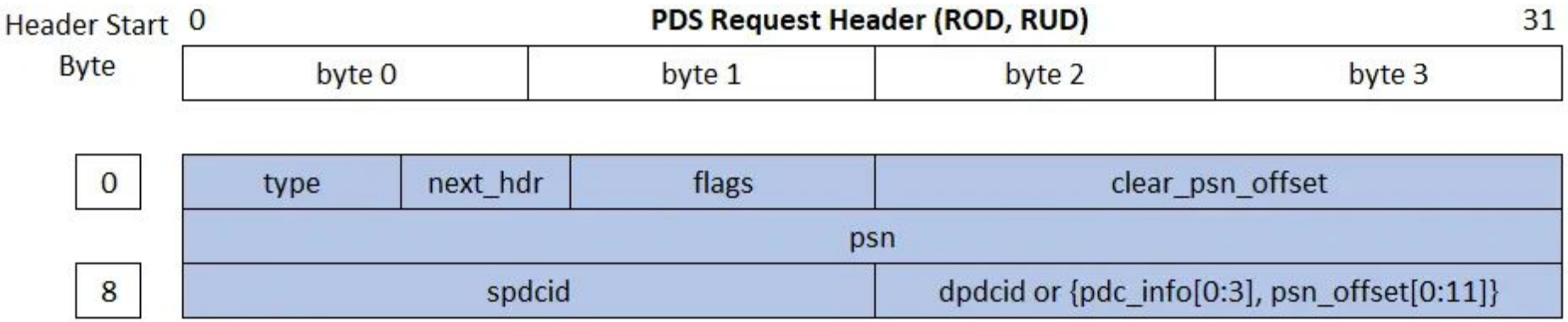
- Low latency, high throughput, simple implementation in HW and SW
- Lightweight connections (PDC)
- Message and initiator/target based comm
- Congestion control with rapid response
- Unordered and ordered packet delivery
- Multi-path and packet spraying
- RDMA and collective operations

# UET stack


7
Application
6
Presentation
5
Session
4
Transport
3
Network
2
Data Link
1
Physical
Software APIs
libfabrics
msg
API
Semantics
Map ULP APIs to packets,
Transaction tracking, ordering, completions, etc.
Packet Delivery
Congestion Management
Transmit rate control,
Adaptive path selection
Telemetry module
Reliability
Reliable delivery,
Packet ordering, SACK
Security
Encryption, Key Management
packets
Ethernet
Fabric

# Packet Delivery Sublayer (PDS)

- Implements reliable delivery of protocol messages
- Request packets, response packets (ACK/NACK), and control packets
- Four *packet delivery* modes

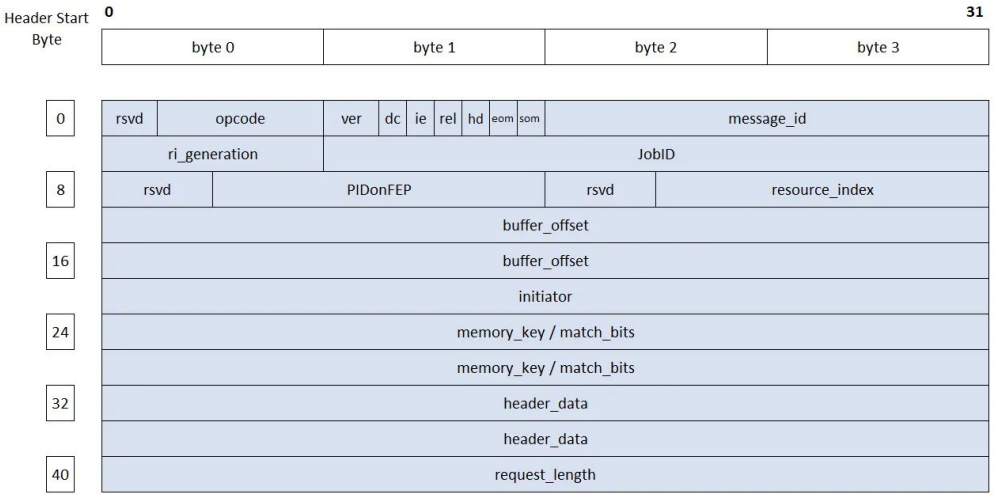


# Semantic Sublayer (SES)

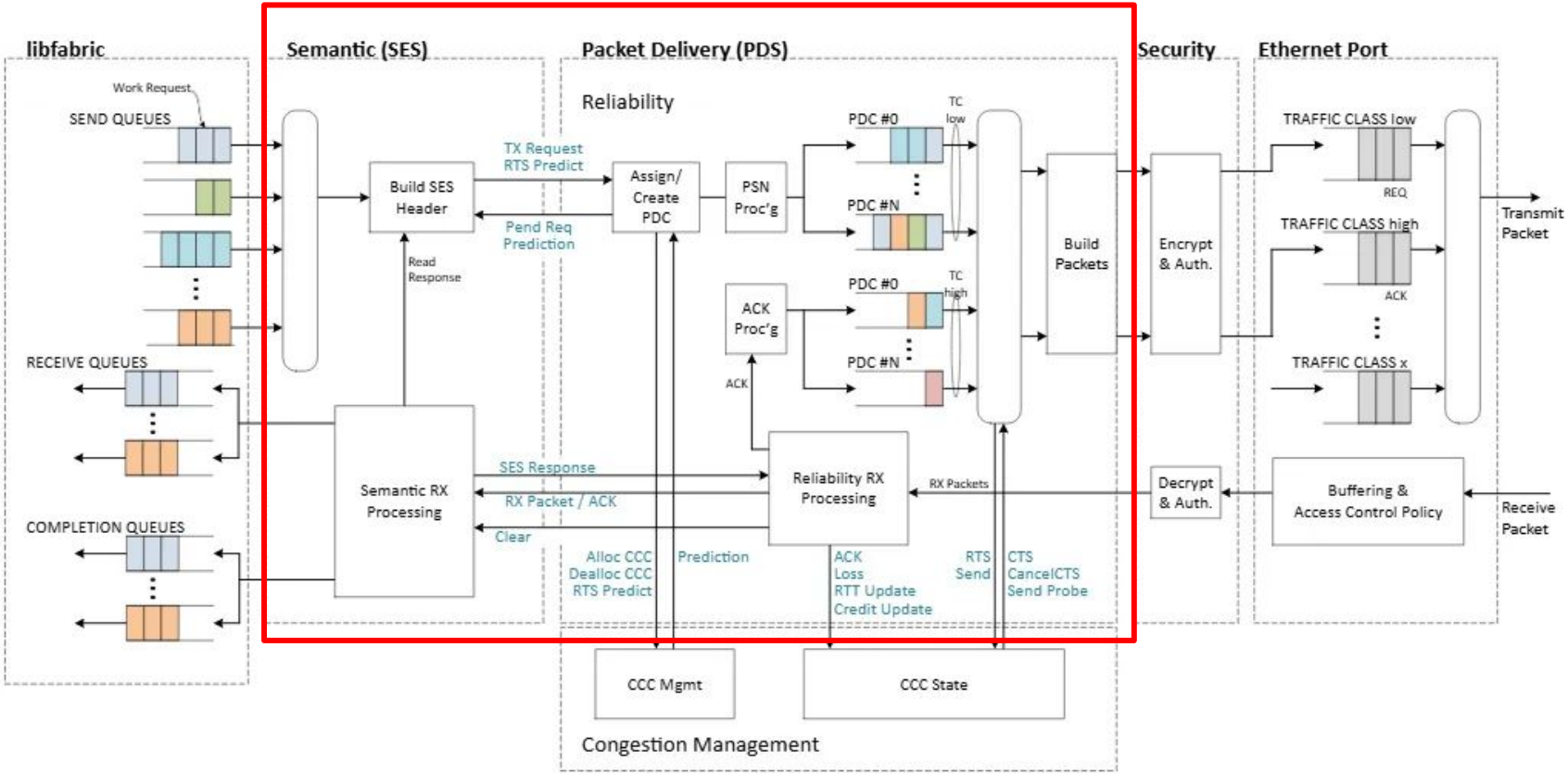
Defines protocol for a initiator-target data transactions including:

- **SEND**, **WRITE, READ**
- **ATOMICS**
- **RENDEZVOUS** and **DEFERRABLE SEND**
- **RESPONSE** (from target to initiator)

*Standard Request Header*

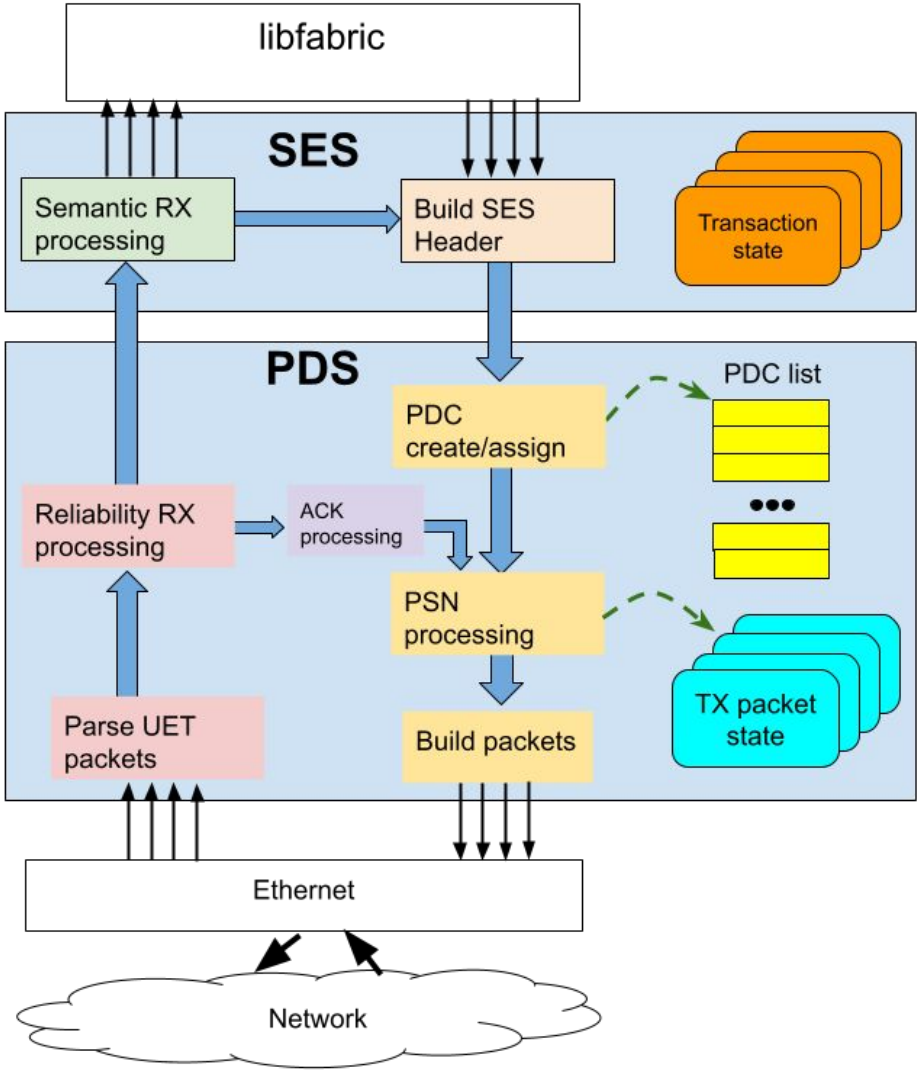


# Implementing UET in XDP2



# Program components

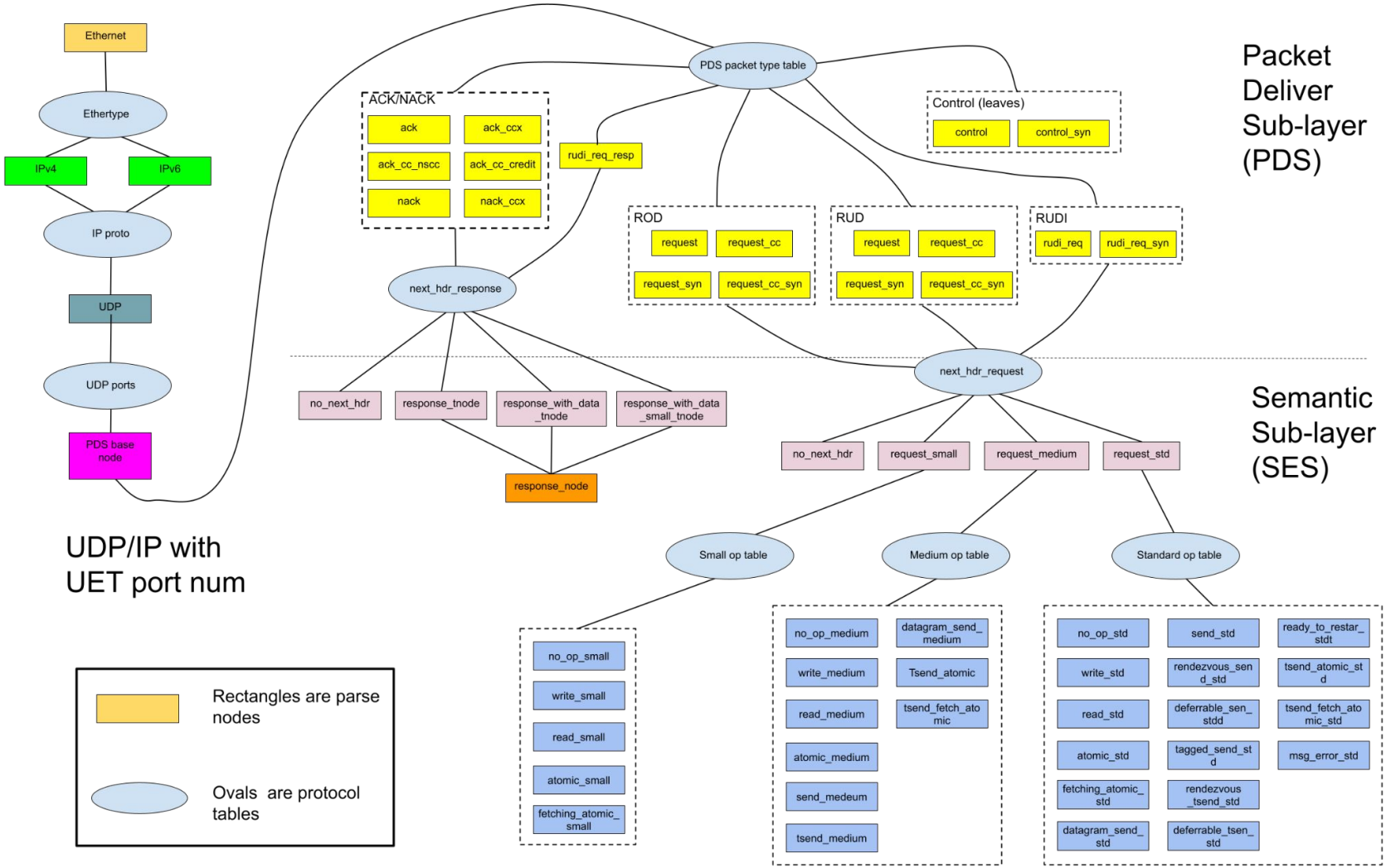
- Receive path
- Transmit path
- Timeouts
- Interface to ULP (libfabric)
- PDC (connection) managements



# XDP2 Services needed

- Protocol parser
- Tables
- Timers
- PVbufs (to hold scatter gather data)
- Bitmaps
- parse_dump to dump UET packets
- Configuration facility
- CLI

# UET parse graph



# The code on GitHub: **xdp2-dev/xdp2**

UET directories and files

- src/include/uet
- src/lib/uet
- src/test/uet
- src/tool/packets/uet
- documentation/protocol/uet.md

Thank you!