Separation of Data Path and Data Flow Sublayers in the Transport Layer

draft-asai-tsvwg-transport-review-03

Hirochika Asai <panda@wide.ad.jp>

PANRG, IETF 114 @ Philadelphia, PA, USA
July 28, 2022
Motivation: Reconsider the Internet Architecture for Future Protocol Development

- **End-to-End Principle**
  - “Dumb” network with smart end-hosts
- **Smarter networks break RFC 1122 & 1123 layering model.**
  - QoS (e.g., DiffServ, ECN, Segment Routing)
  - Middlebox (e.g., Firewall, Content caching, Transcoding, TCP acceleration)
  - New distributed computing paradigm (e.g., Pub/sub model for machine-to-machine communication, Edge computing, In-network computing)
Gap Analysis between IP Routing and Transport Layer Protocols

• **IP routing**
  - Hop-by-hop store-and-forward with a single or multiple queues

• **Transport layer protocols**
  - **End-to-End** functionality
    - Reliable data communication (integrity check, retransmission, etc.)
    - Flow control for receive buffer management
    - Transport layer security (TLS)
  - **Path-dependent** (soft dependent) functionality; with observation from endhosts
    - MTU discovery (ICMP)
      - MTU may not be consistent if the path is changed.
    - Congestion control / Explicit congestion notification (ECN)
      - Bandwidth capacity and congestion may change on path changes.
  - **Waypoint-dependent** middleboxes; with a gateway or policy-based routing.
    - Firewall (stateful)
    - NAT (NAPT)
    - Transparent proxy (content caching, transcoding, etc.)
    - TCP accelerator
More path-aware communications

- Multipath protocols
  - SCTP, MP-TCP, MP-QUIC, MP-DCCP, etc...
- AQM-aware congestion control (e.g., L4S)
- Pub/sub-based M2M communication
- Edge computing
- Network slicing
- In-network computing / In-network telemetry
- ...

emerging with net programmability
Problems with Transport Layer Protocols

• Tightly-coupled design of transport layer protocols
  • All components of transport layer functionality are integrated to one protocol.
    • e.g., multipath protocols are developed for each transport layer protocols.
      • SCTP, MP-TCP, MP-QUIC, MP-DCCP

→ Hard to develop both “IP-friendly” and “transport-layer-friendly” network extensions.
Reviewing Transport Layer Functionality: Data Path vs. Data Flow

• Data Path  → Stateless or per-path states
  • Trajectory & waypoint handling
  • Bidirectional connection
  • Resource monitoring (e.g., congestion monitoring like ECN or in-network telemetry)
  • Congestion control
  • Data flow multiplexing
  • Packet duplication for packet loss recovery (like FEC or detnet replication)

• Data Flow  → Per-flow states
  • Retransmission for reliable data communication
  • Flow control (buffer management)
  • Flow prioritization
  • End-to-end security
  • Inverse multiplexing for multipath protocols
Proposal: Data Path and Data Flow Sublayers

- Data flow layer: Retransmission, flow control, flow prioritization, end-to-end security, inverse multiplexing (over multiple data paths)
- Data path layer: In-band trajectory monitoring, waypoint management, bidirectional connection, quality monitoring, congestion control, data flow multiplexing, duplication

(may update RFC 1122, 1123)
Use Case: Flow Arbitration in IoT

1. Start data transfer
2. Congestion control message (like ECN)
3. Reduce speed responding to congestion control message
Use Case: Flow Arbitration in IoT (cont’d)

N.B. This scenario could be achieved with TCP, ECN and DSCP as the data path functionality is limited.
Next Step

• As research:
  • Discuss the gap between hop-by-hop and end-to-end from the viewpoint of data path and data flow
  • Review existing transport layer protocol functionality
  • Architect for new communication models of distributed computing
    • Pub/Sub-based machine-to-machine communication
    • Multiaccess edge computing
    • In-network computing

• As protocol development:
  • Design an example of data path and data flow protocol