Transport Layer Overview

Manas Ranjan Lenka
School of Computer Engineering,
KIIT University

Transport Layer

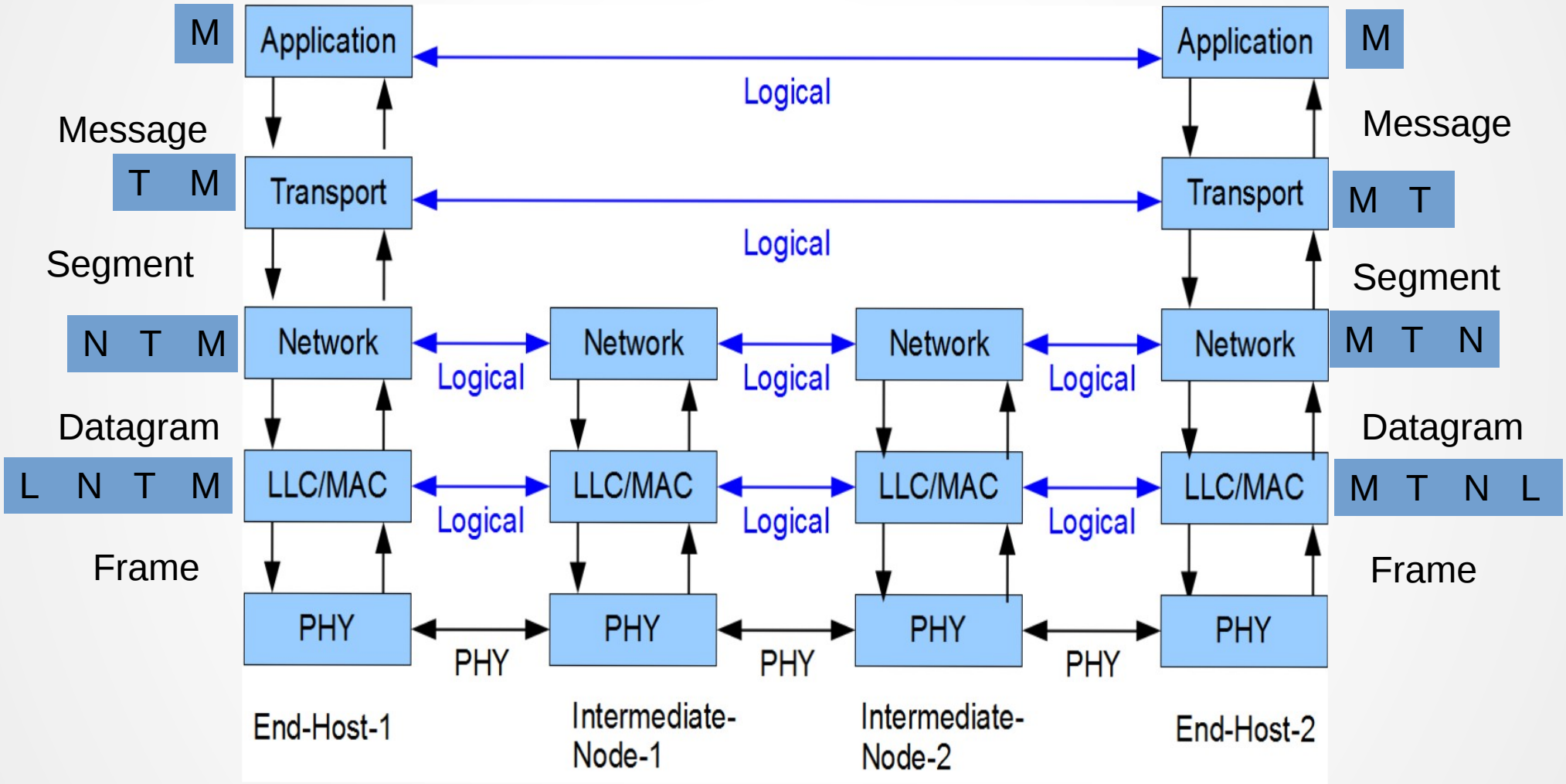
- Hosts run many application processes

- Transport layer provides logical communication between processes
  - Help multiplex/demultiplex packets to deliver to right process
  - Enhance network layer services

- Transport protocols also called end-to-end protocols since they are implemented on end hosts

- The unit of data at transport layer is termed ‘segment’

End to End vs Hop to Hop



Application Layer Expectations

- Guaranteed message delivery
- Ordered delivery
- Delay guarantees
- Bandwidth guarantees
- No duplication
- Support arbitrarily large messages
- Support flow control

Network Layer Limitations

- Best effort service model
- Packet Losses
- Re-ordering
- Duplicate copies
- Limit on maximum message size
- Long delays

Challenges

- Enhance network layer services to meet application expectations
  - Cannot provide services that inherently cannot be supported by network layer (e.g. delay guarantees, bandwidth guarantees)
- Different transport protocols offer different tradeoffs
  - User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Real-time Transport Protocol (RTP)

Transport-layer Services

- Process-to-Process communication
- Addressing
- Multiplexing/Demultiplexing
- Flow Control
- Error Control
- Congestion Control

Transport-layer protocols

- reliable, in-order delivery (TCP)
  - congestion control
  - flow & error control
  - connection setup
- unreliable, unordered delivery: UDP
- services not available:
  - delay guarantees
  - bandwidth guarantees

Multiplexing/demultiplexing

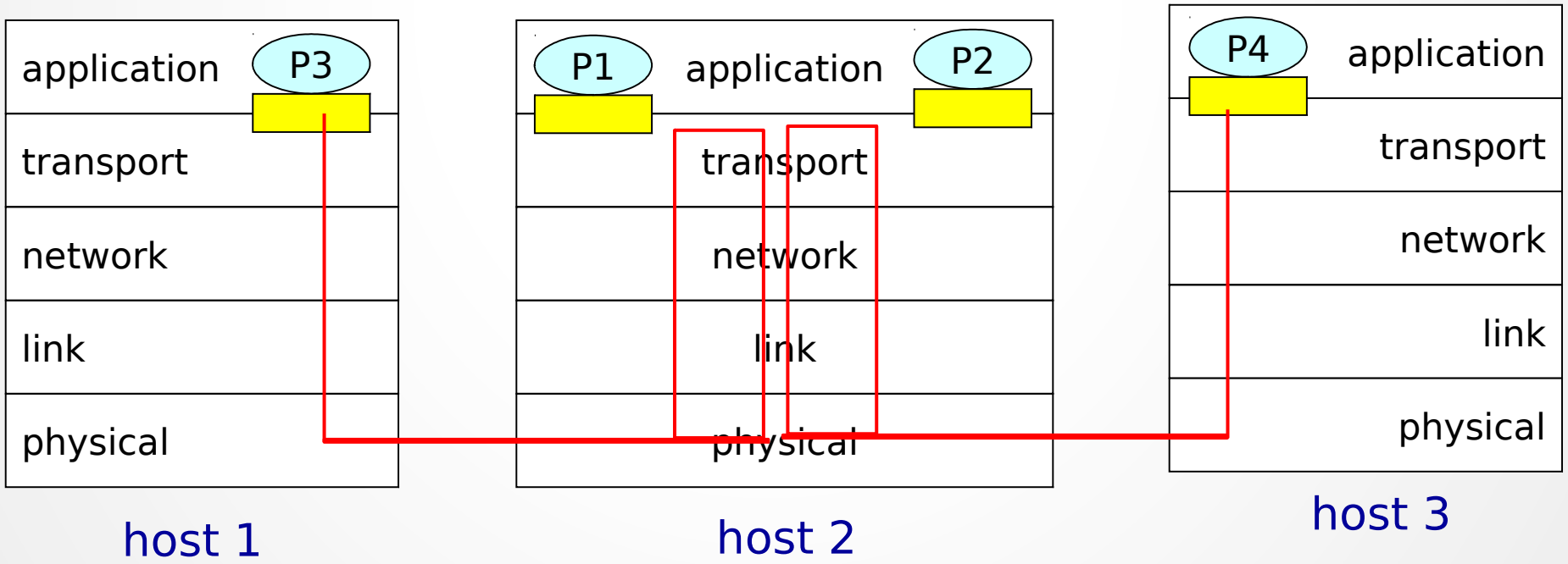
Demultiplexing at rcv host:

delivering received segments to correct socket

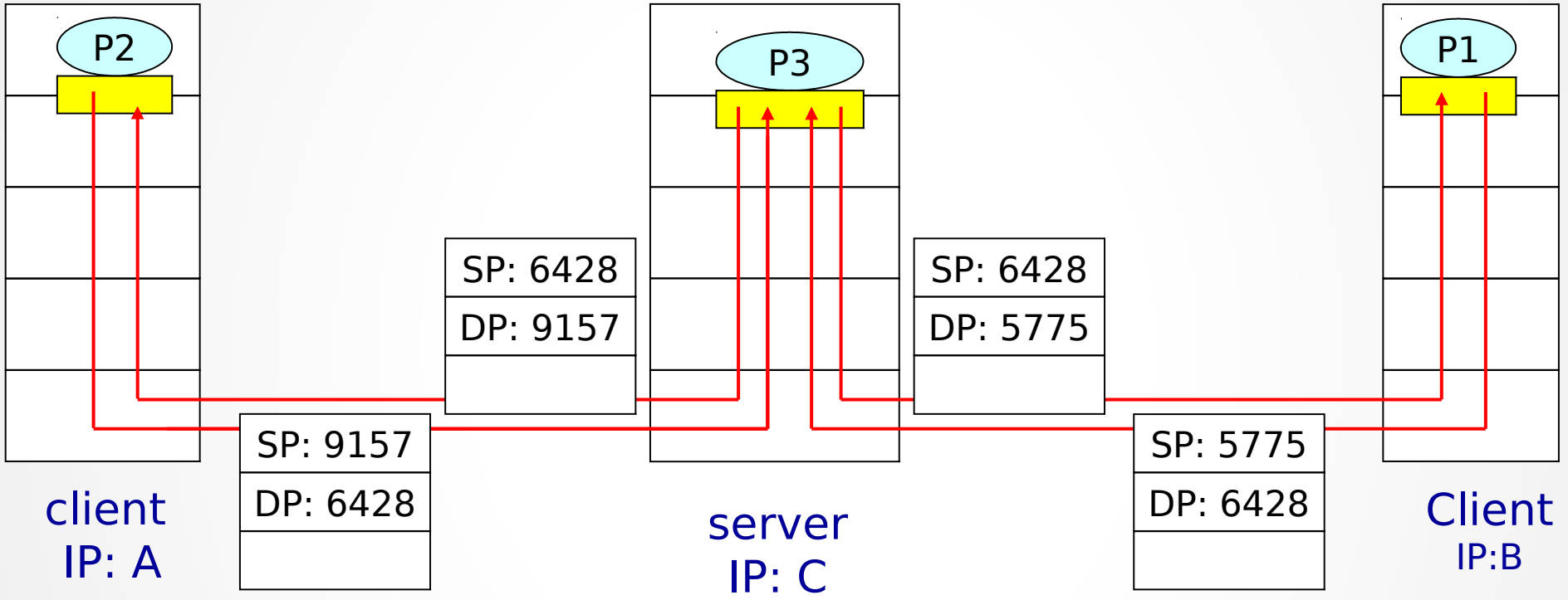
Multiplexing at send host:

gathering data from multiple sockets, enveloping data with header (later used for demultiplexing)

= socket = process

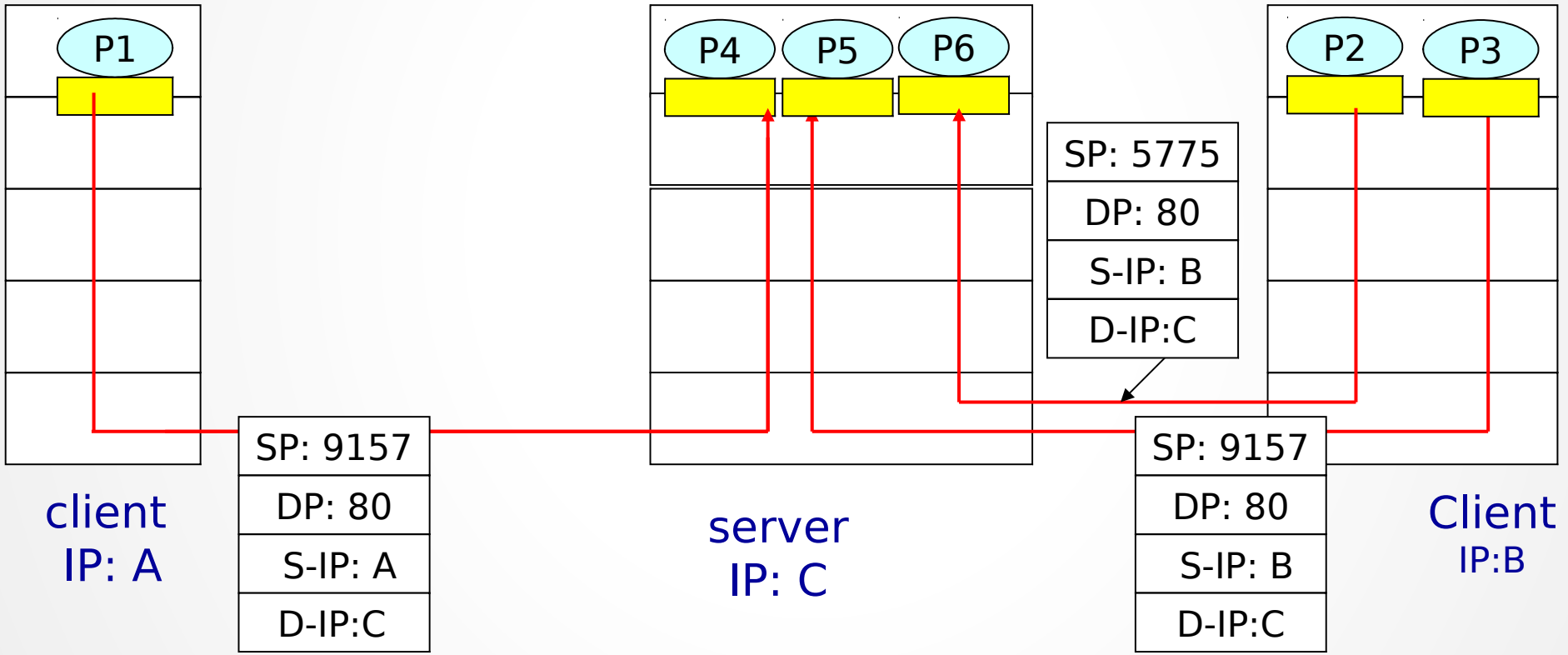


Connectionless demux



A UDP socket is fully identified by a 2-tuple consisting of a destination IP and port.

Connection-oriented demux



A TCP socket is fully identified by a 4-tuple consisting of a Source IP,port and destination IP, port.