Transport Layer – Overview

Kameswari Chebrolu

Milestones

- Progression in scale of networks
  - Point-to-point link (2 nodes)
  - Small local area networks (tens of nodes)
  - Extended local area networks (thousands of nodes)
  - Heterogeneous inter-networks (millions of nodes)

Milestones

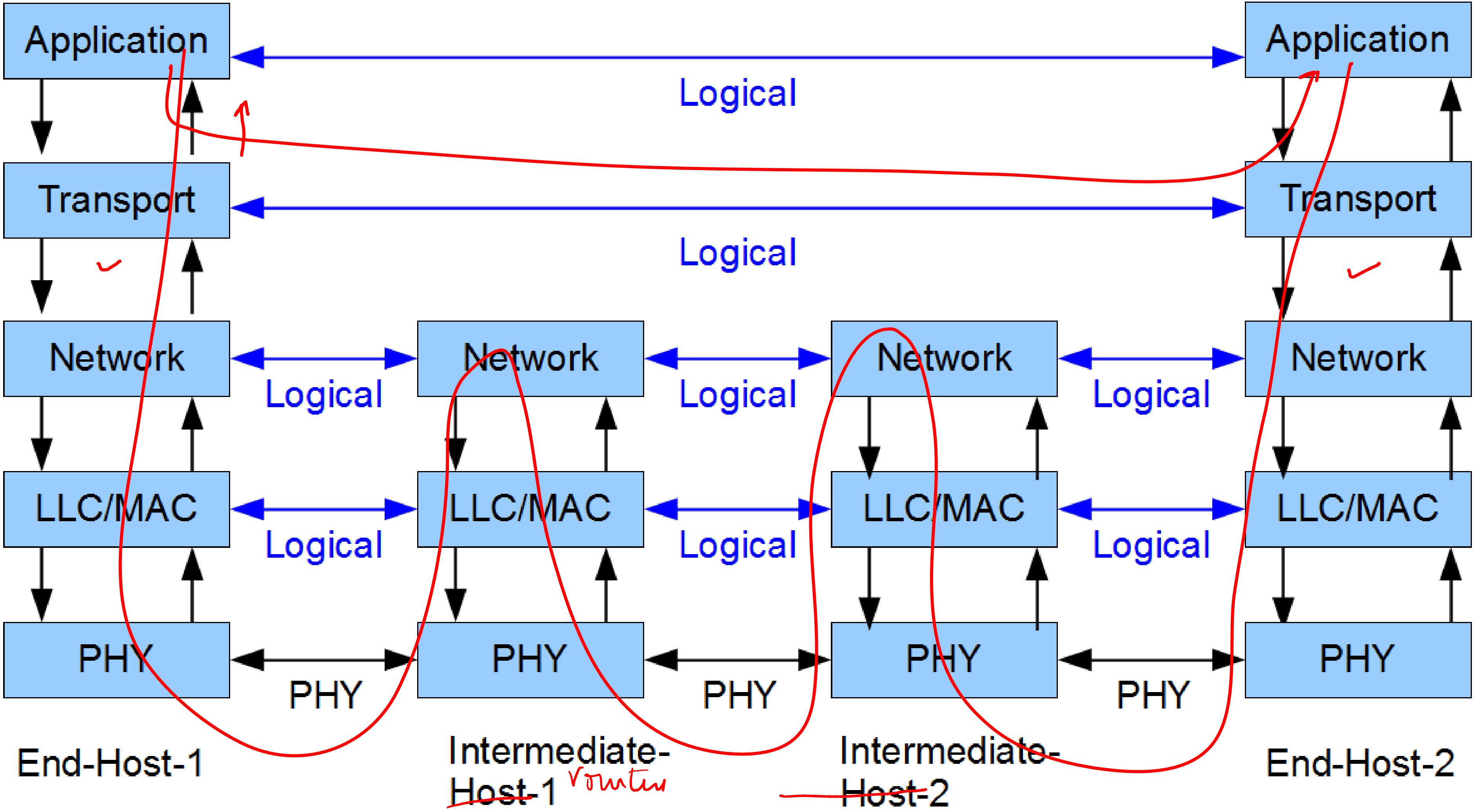
- Can now handle host-to-host delivery
  - Network layer (determines which next hop) uses services of link layer (delivers to next hop) which in turn uses services of physical layer (converts bits to signals) to deliver packets
- Next: Process to process communication → role of the transport layer

Transport Layer Service

tcp

web browsers
email
SSH

- 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’


Application
Application
Logical
Transport
Transport
Logical
Network
Network
Network
Network
Logical
Logical
Logical
LLC/MAC
LLC/MAC
LLC/MAC
LLC/MAC
Logical
Logical
Logical
PHY
PHY
PHY
PHY
PHY
PHY
PHY
End-Host-1
Intermediate-~~Host~~-1 router
Intermediate-~~Host~~-2
End-Host-2

Application Layer Expectations

Email
File transfer

Guaranteed delivery

- Guaranteed message delivery
- Ordered delivery
- Delay 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 → MTU
- Long delays

Challenge

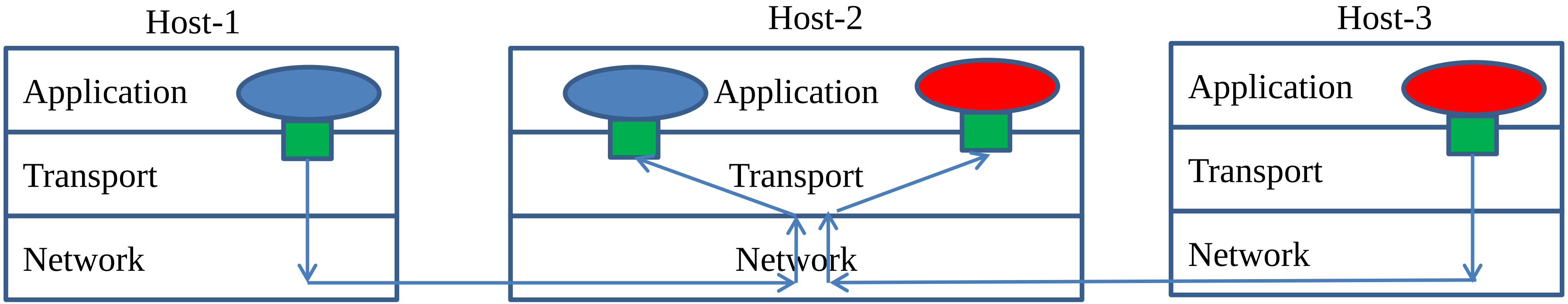
- Enhance network layer services to meet application expectations
  - Cannot provide services that inherently cannot be supported by network layer (e.g. delay guarantees)
- Different transport protocols offer different tradeoffs

  logical

  - User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Remote Procedure Call (RPC), Real-time Transport Protocol (RTP)

Break

Multiplexing/Demultiplexing



Demultiplexing: Deliver segments to the right socket

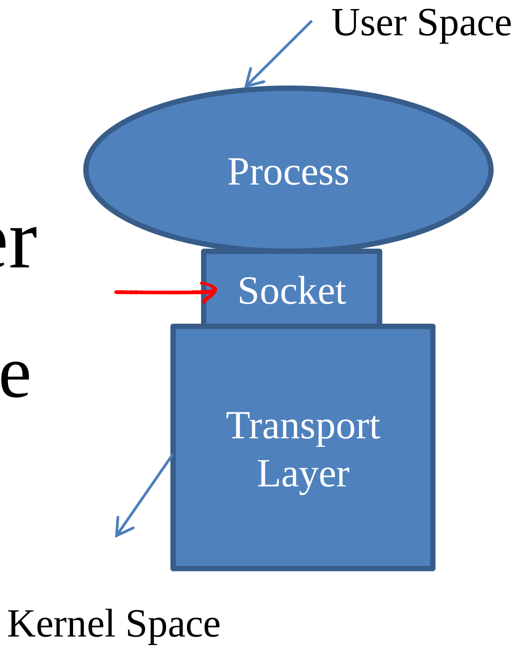
Multiplexing: Assemble segments such that they get delivered to right socket

<table>
<tr><td><b>Source Port</b></td><td><b>Destination Port</b></td></tr>
<tr><td colspan="2"><b>Other fields in header</b></td></tr>
<tr><td colspan="2"><b>Application Data</b></td></tr>
</table>

Transport Layer Segment

Sockets

- Socket: An interface between an application process and transport layer
  - The application process can send/receive messages to/from another application process (local or remote) via a socket
- In Unix jargon, a socket is a file descriptor – an integer associated with an open file



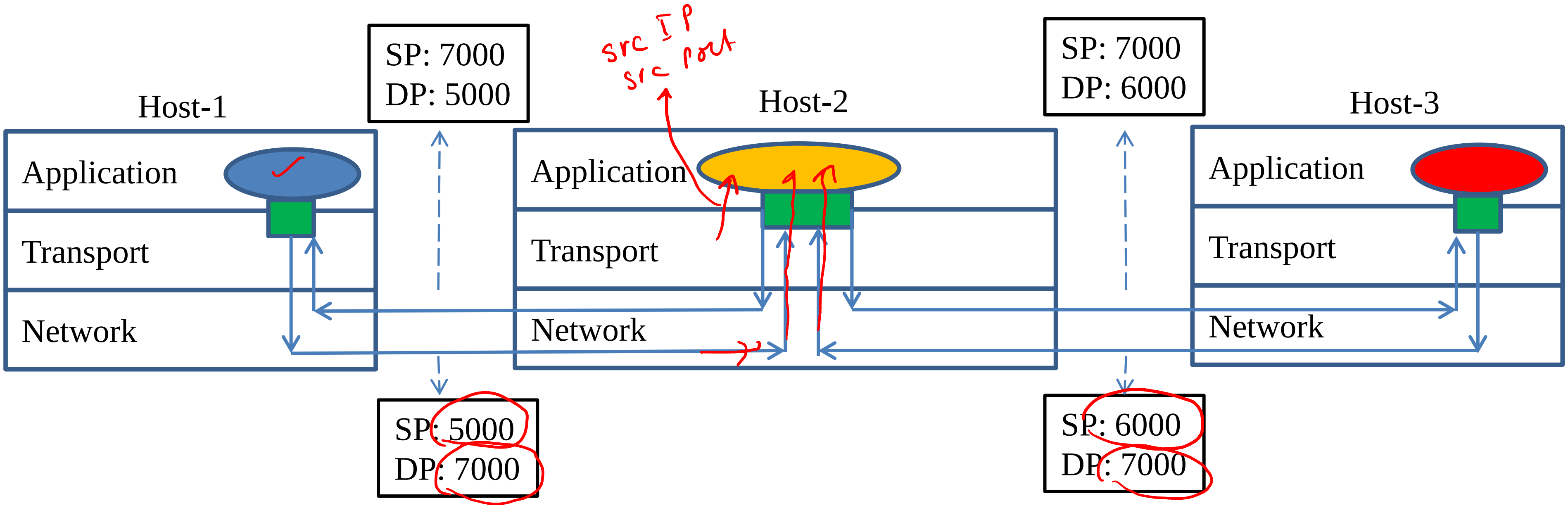
Multiplexing/Demultiplexing

- Application developer can
  - specify type of transport protocol UDP TCP
  - configure a few parameters related to transport protocol
- To help mux/demux a segment
  - Sockets have unique identifiers (one of them is ports)
  - Segments carry fields that help identify right socket
    - Fields of relevance: Source and destination port

Connectionless Mux/Demux

- Used with UDP sockets
- Socket identified by two-tuple:
  - Destination IP address, Destination port number
- Transport layer checks port information in segment and directs to right socket
- IP datagrams with different source IP addresses and/or source port numbers directed to same socket

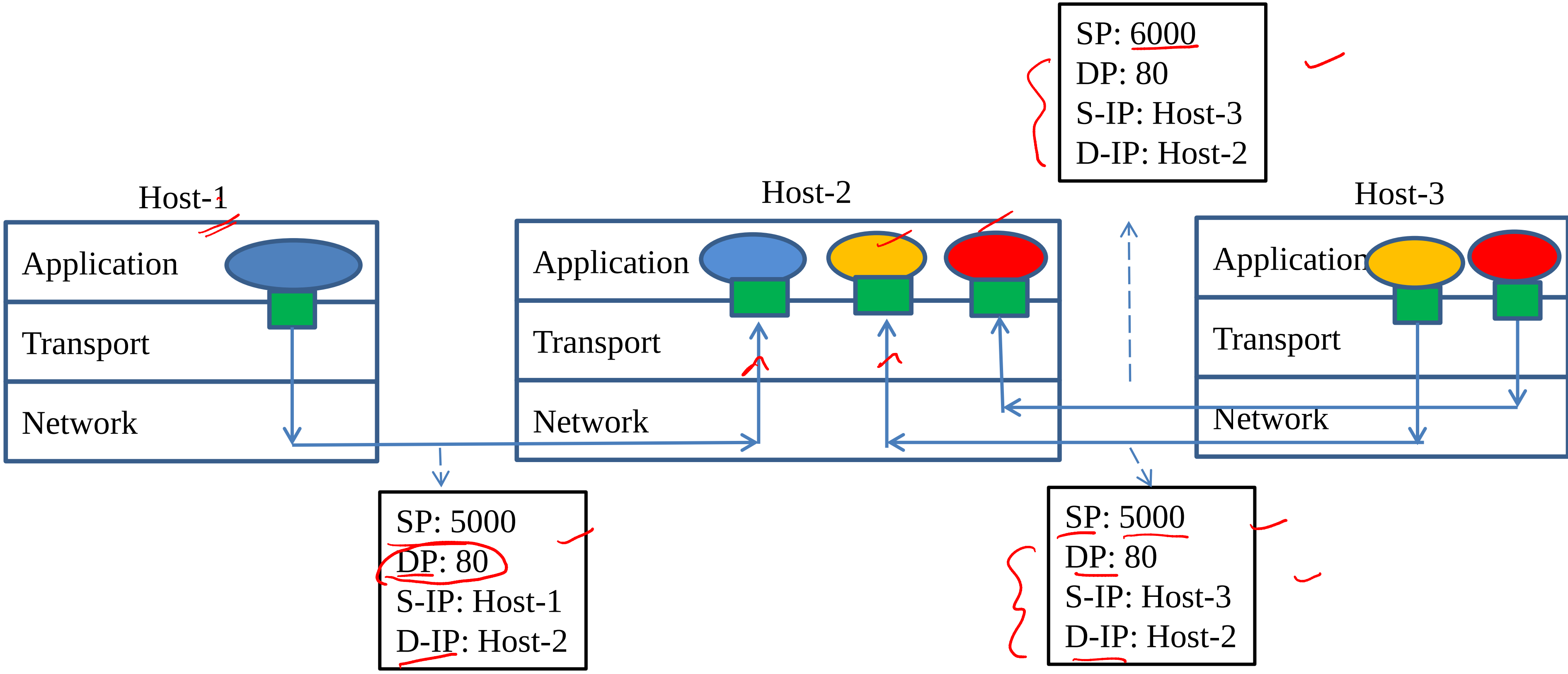
Example


Host-1
Application
Transport
Network
SP: 7000
DP: 5000
SP: 5000
DP: 7000
src IP
src port
Host-2
Application
Transport
Network
SP: 7000
DP: 6000
SP: 6000
DP: 7000
Host-3
Application
Transport
Network

Connection-oriented Mux/Demux

- Used with TCP sockets
- Socket identified by 4-tuple:
  - Source IP address ✓
  - Source port number ✓
  - Destination IP address ✓
  - Destination port number ✓
- All four values are used to direct segment to the right socket

Example



SP: 6000
DP: 80
S-IP: Host-3
D-IP: Host-2

SP: 5000
DP: 80
S-IP: Host-1
D-IP: Host-2

SP: 5000
DP: 80
S-IP: Host-3
D-IP: Host-2

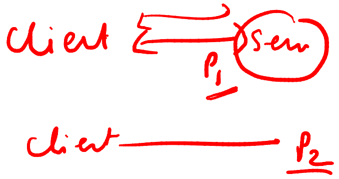
Obtaining Port Information

- Client contacts server

  src IP, src port

  - Client picks a random src port and sends message
  - Server knows identity of client process (based on source port in received message)

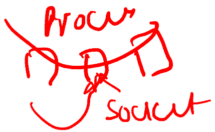
- How does client know server's port info?

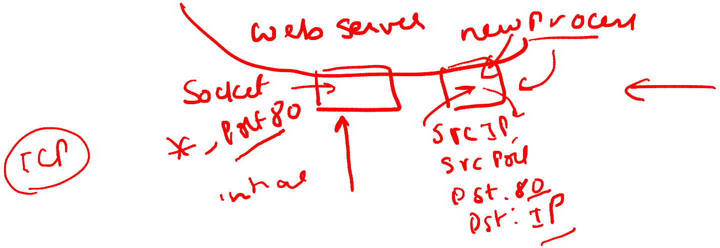
  

  - Server's listen to messages on well known ports
  - Refer to /etc/services in Unix systems
  - In some applications, well known port is the starting point to agree upon some other port

A Note on Servers

- Server host listens on a designated port but has different socket for each connecting client
  - Each socket identified by its own 4-tuple
  - There need not be one-to-one correspondence with sockets and processes
    - E.g. Threaded server have many sockets but one process





Summary

- The role of transport layer is to provide logical communication between processes
  - All transport protocols provide multiplexing and demultiplexing capability
  - Others try to enhance network services to meet application specific requirements
- Different types of mux/demux and role of sockets