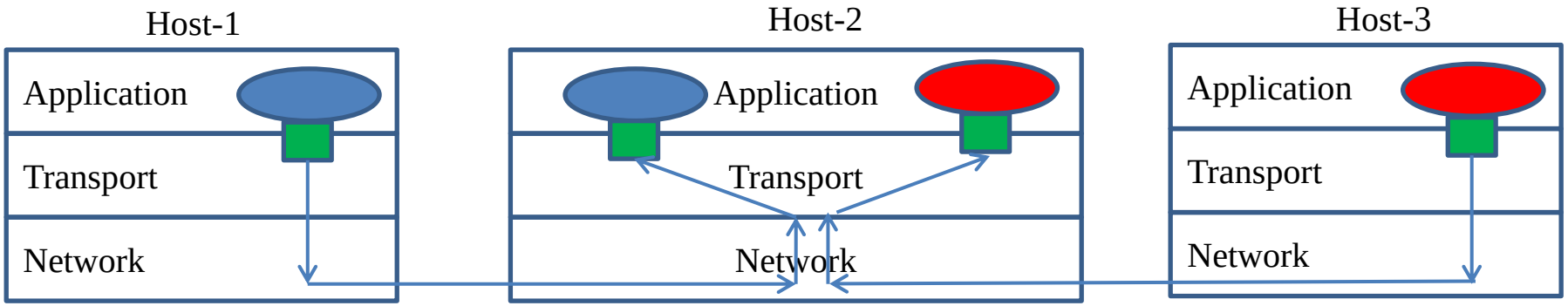
Transport Layer – Overview

Kameswari Chebrolu

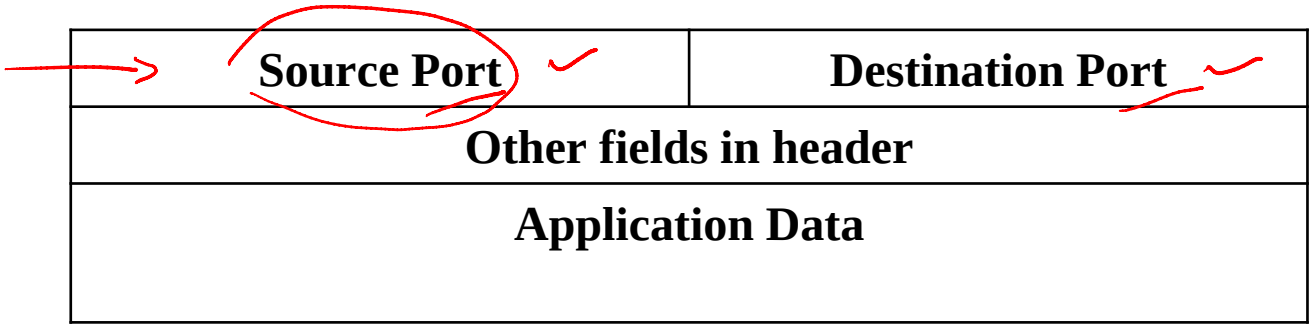
Break

Multiplexing/Demultiplexing



Demultiplexing: Deliver segments to the right socket

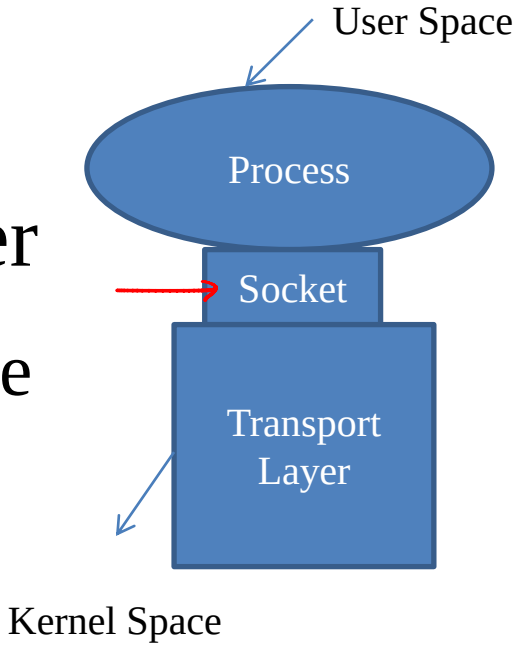
Multiplexing: Assemble segments such that they get delivered to right socket



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



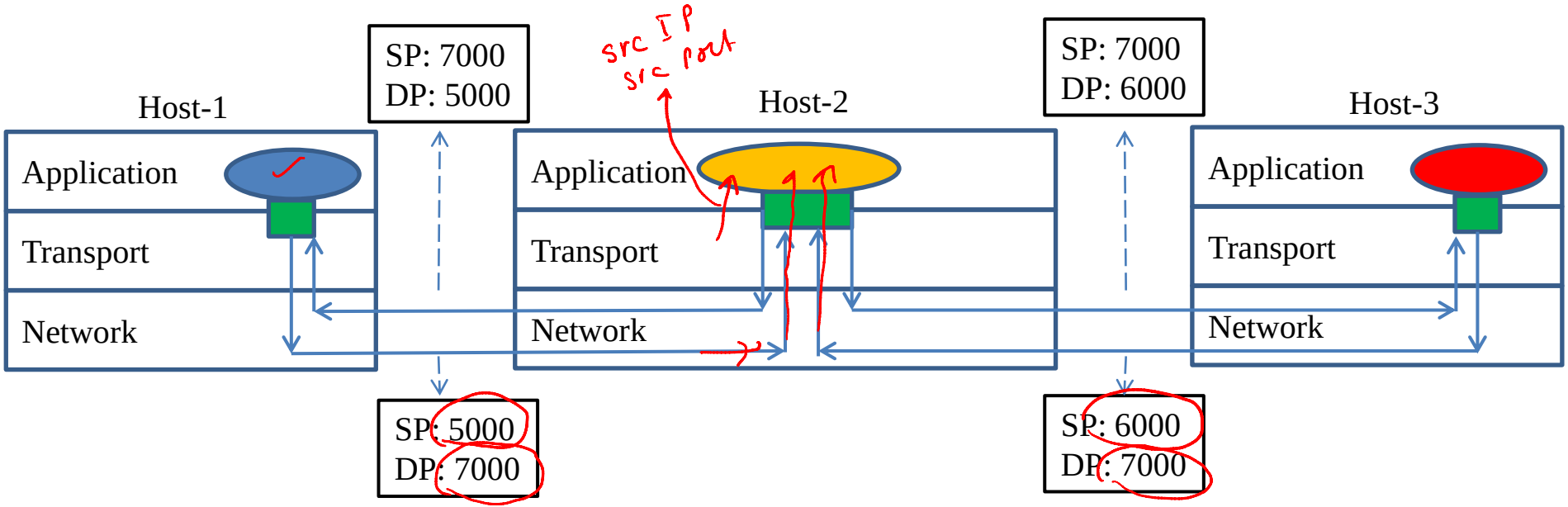
Multiplexing/Demultiplexing

- Application developer can
  - specify type of transport protocol
  - 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

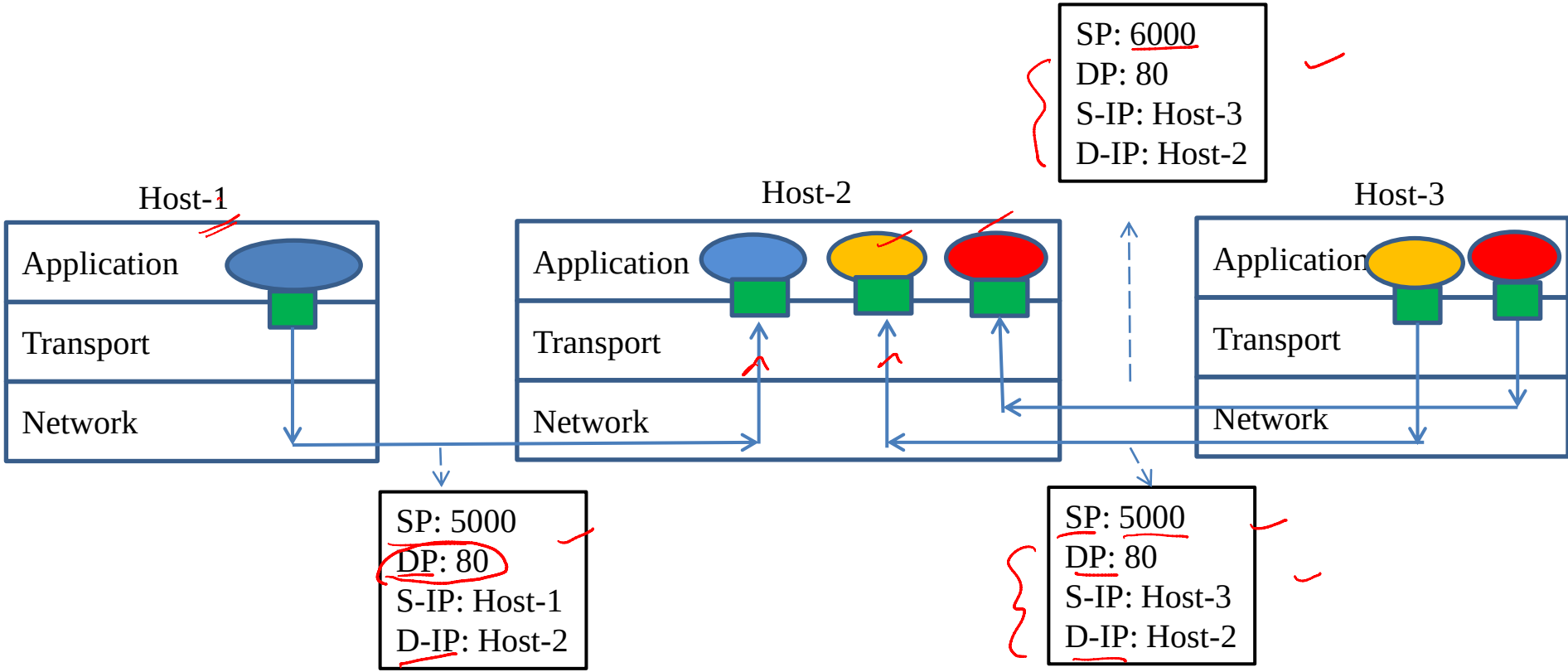
Example


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

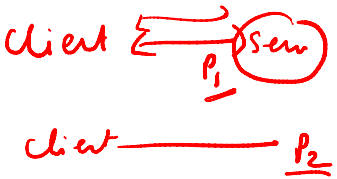
- Used with TCP sockets
- Socket identified by 4-tuple:
  - Source IP address
  - Source port number
  - Destination IP address
  - Destination port number
- All four values are used to direct segment to the right socket

Example


SP: 6000
DP: 80
S-IP: Host-3
D-IP: Host-2
Host-1
Application
Transport
Network
Host-2
Application
Transport
Network
Host-3
Application
Transport
Network
SP: 5000
DP: 80
S-IP: Host-1
D-IP: Host-2
SP: 5000
DP: 80
S-IP: Host-3
D-IP: Host-2

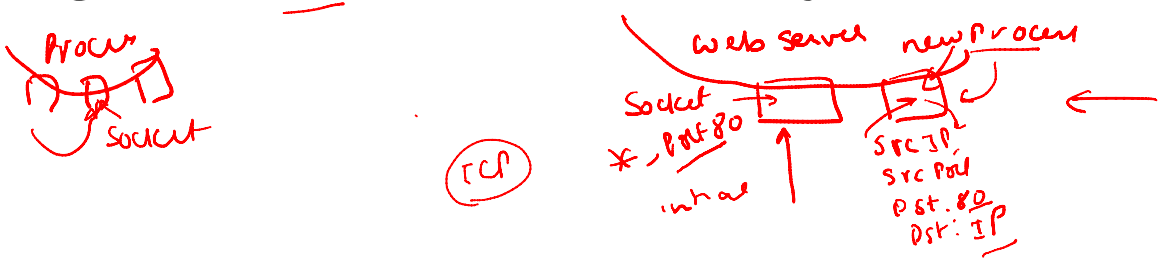
Obtaining Port Information

- Client contacts server
  - Client picks a random 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