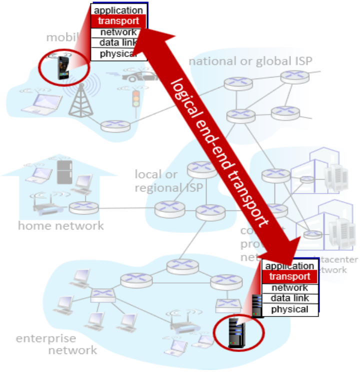
Transport Layer

Dr. G. Omprakash

Assistant Professor, ECE, KLEF

Transport Layer

- Transport Layer is located between the application layer and the network layer
    - It provides services to the application layer
    - It receives services from the network layer.



- if the transport layer service is so similar to the network layer service, why are there two distinct layers?

Process-to-Process Communication

- First duty of a transport-layer protocol is to provide **process-to-process** communication.
- **Process** is an application-layer entity (running program)
- Network layer is responsible for communication at the computer level (host-to-host communication).
    - Network-layer delivers the message only to the destination computer
        - The message still needs to be handed to the correct process
- A transport-layer protocol is responsible for delivery of the message to the appropriate process

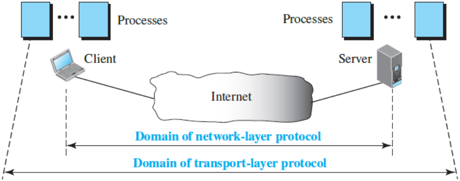

Processes
Processes
Client
Server
Internet
Domain of network-layer protocol
Domain of transport-layer protocol

Figure: Network layer versus transport layer

Analogy with Network Layer

For communication to take place

- Define local host ↔ remote host (using **IP address**)
  - Network layer largely runs on the routers
- Define local process ↔ remote process (using **port numbers**)
  - Transport code runs entirely on the users' machines

Port Numbers

- The port numbers are integers between 0 and 65,535 (16 bits)
- Client program defines itself with a port number (**ephemeral port number**)
  - *ephemeral* means ”short-lived”
- Server process must also define itself with a port number
  - The process at the client site that wants to access that server and use its services
    - Port number on server side cannot be random
- TCP/IP has decided to use universal port numbers for servers

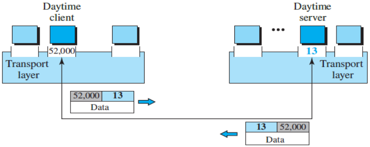


Figure: Port numbers

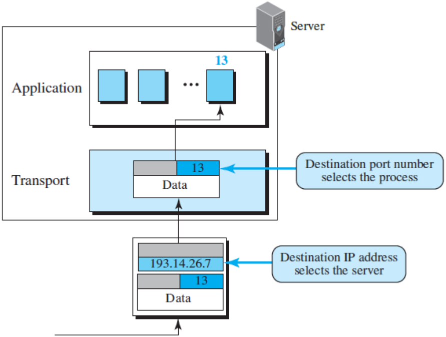

Server
Application
13
Transport
13
Data
Destination port number selects the process
193.14.26.7
13
Data
Destination IP address selects the server

Figure: IP addresses versus port numbers

- IP address defines the host among the different hosts in the world
- The port number defines one of the processes on this particular host

Multiplexing at sender

- Handle data from multiple sockets, add transport header

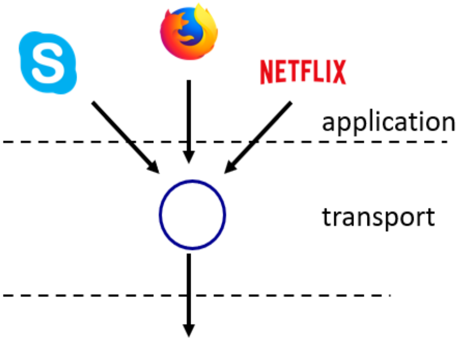


Figure: Multiplexing

Demultiplexing at receiver

- host uses IP addresses & port numbers to direct segment to appropriate socket

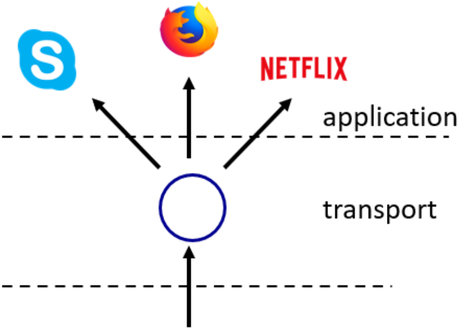
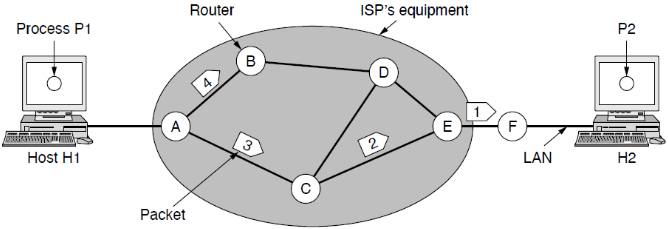


Figure: Demultiplexing

Recap -Network Layer

Connectionless service : Different paths for different datagrams belonging to the same message.



A's table (initially)		A's table (later)		C's table		E's table	
A	–	A	–	A	A	A	C
B	B	B	B	B	A	B	D
C	C	C	C	C	–	C	C
D	B	D	B	D	E	D	D
E	C	E	B	E	E	E	–
F	C	F	B	F	E	F	F

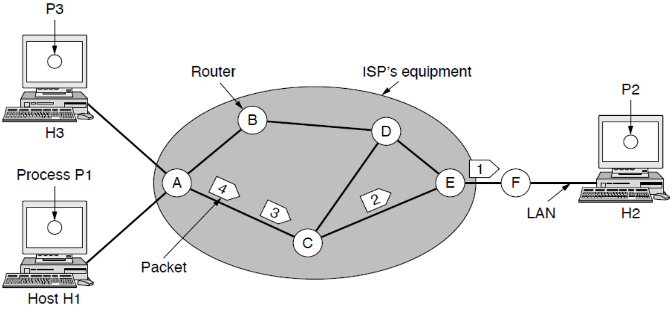
Dest. Line

Figure: Routing in datagram network

Recap -Network Layer

Connection-oriented service: Path is established first, then the packets sent.

In the network layer, connection-oriented service means a coordination between the two end hosts and all the routers in between



A's table

In		Out	
H1	1	C	1
H3	1	C	2

C's table

A	1	E	1
A	2	E	2

E's table

C	1	F	1
C	2	F	2

Connectionless Service

- At the transport layer, we are not concerned about the physical paths of packets
- Connectionless service at the transport layer means independency between packets;

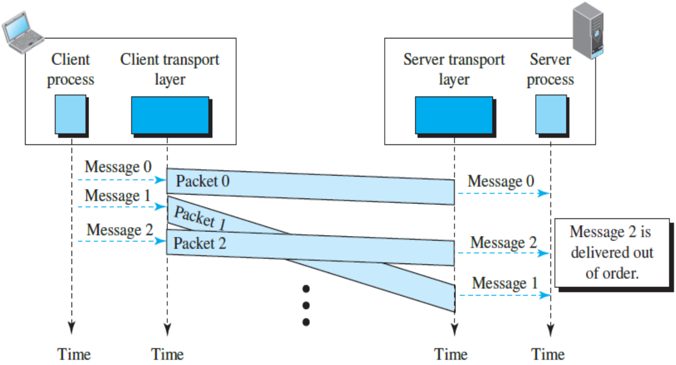


Figure: Connectionless Service

Connectionless Service

- Source process (application program) divides message into chunks of data of the size acceptable by the transport layer
- The transport layer treats each chunk as a single unit (no relation)
- Chunk is encapsulated as a packet and sent
- No relation between the packets at the transport layer $\implies$ the packets may arrive out of order at the destination
- Messages will be delivered out of order to the server process
- Extra delay in transportation of the second packet, the delivery of messages at the server is not in order (0, 2, 1).
- Since there is no numbering on the packets, if one packet is lost $\implies$ the receiving transport layer has no idea that one of the messages has been lost
- No flow control, error control, or congestion control can be effectively implemented in a connectionless service.

Connection-Oriented Service

- The client and the server first need to establish a logical connection between themselves.
    - The data exchange can only happen after the connection is established.
    - After the data exchange, the connection needs to be torn down
- connection-oriented service involves only the two hosts (not routers as in the N/W layer)

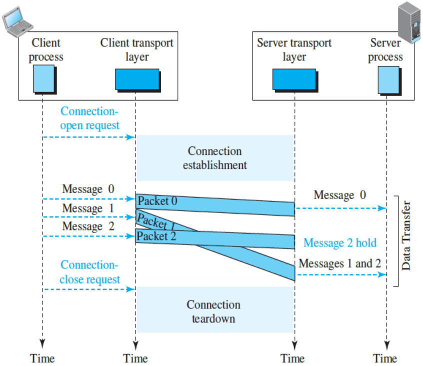


Figure: Connection-oriented Service

Transport-Layer protocols

- UDP: User Datagram Protocol
  - Unreliable connectionless transport-layer protocol
- TCP: Transmission Control Protocol
  - Reliable connection-oriented protocol
- SCTP: Stream Control Transmission Protocol
  - New transport-layer protocol: combines the features of UDP and TCP

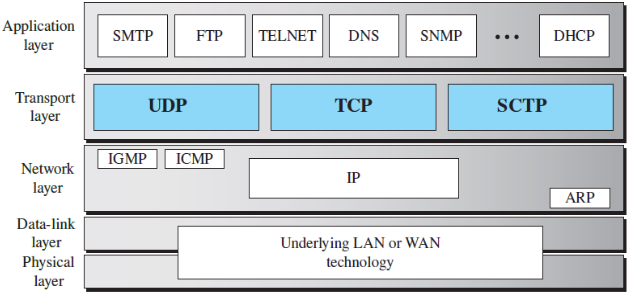


Figure: Position of transport-layer protocols

Acknowledge various sources for the images.
Thankyou