

Chapter 6

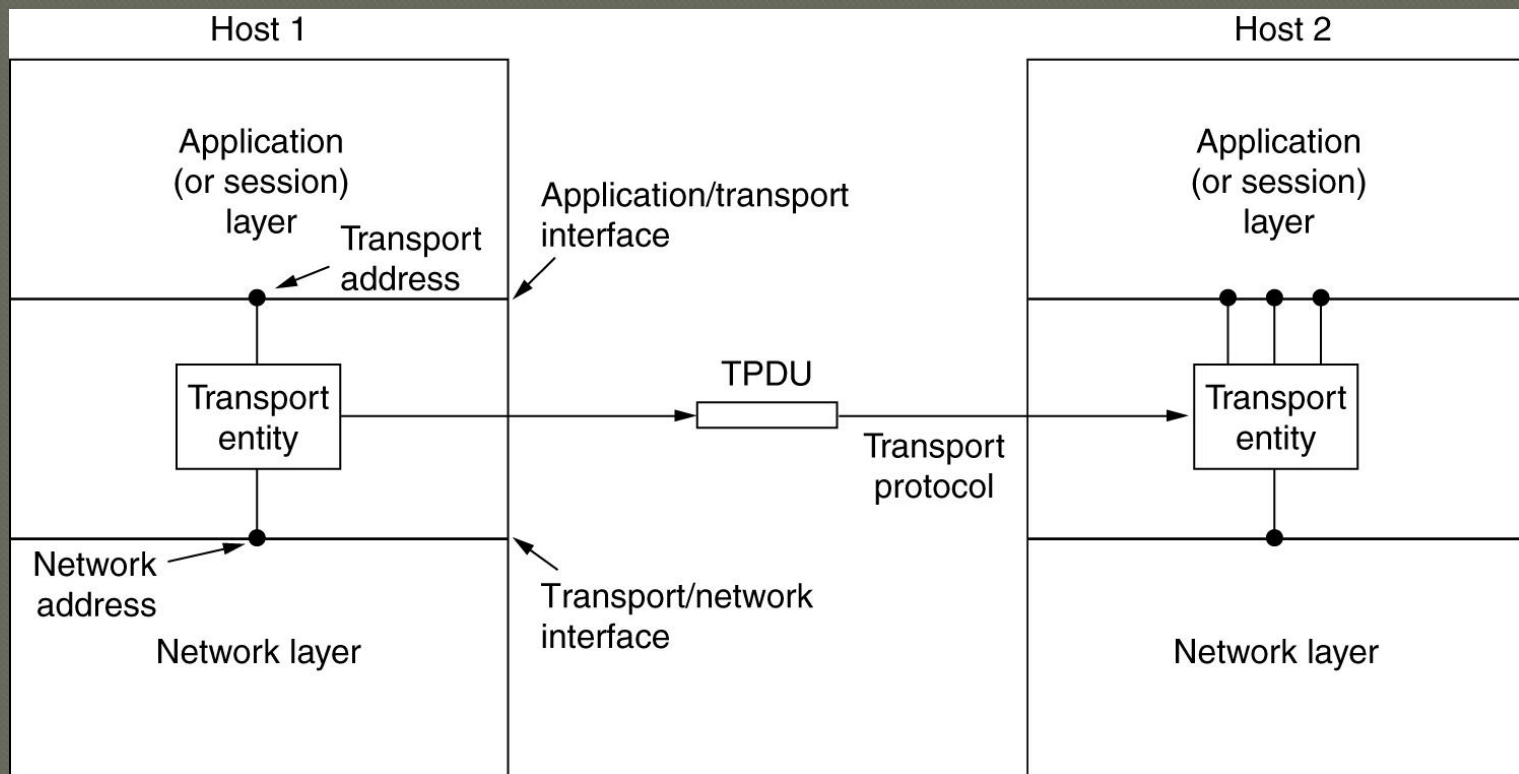
The Transport Layer

The Transport Service

- Services Provided to the Upper Layers
- Transport Service Primitives
- Berkeley Sockets
- An Example of Socket Programming:
 - An Internet File Server

Services Provided to the Upper Layers

The network, transport, and application layers.



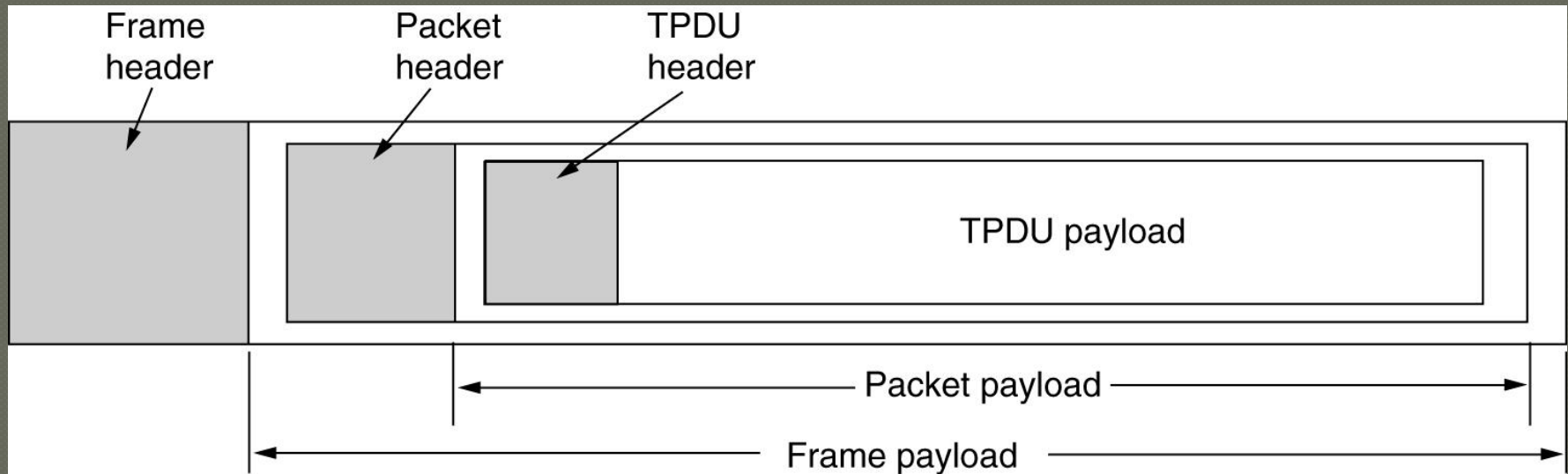
Transport Service Primitives

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

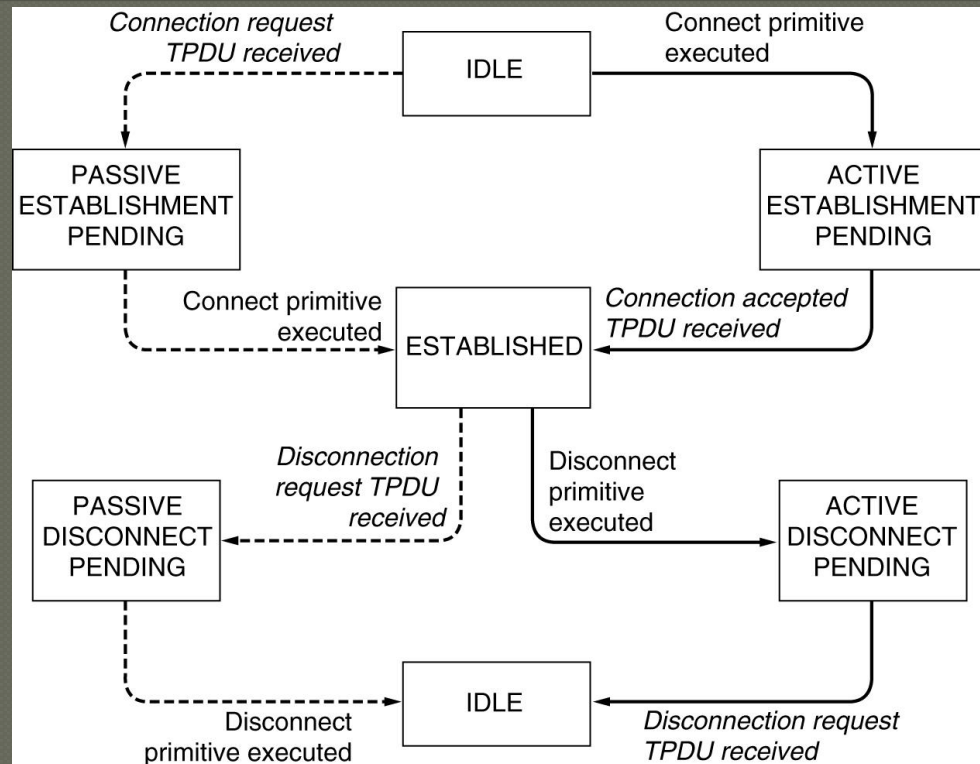
The primitives for a simple transport service.

Transport Service Primitives (2)

The nesting of TPDU, packets, and frames.



Transport Service Primitives (3)



A state diagram for a simple connection management scheme. Transitions labeled in italics are caused by packet arrivals. The solid lines show the client's state sequence. The dashed lines show the server's state sequence.

TCP/UDP sockets

Socket: a door between application process and end-end-transport protocol (UDP or TCP)

TCP service: reliable transfer of bytes from one process to another

Client must contact server

- server process must first be running
- server must have created socket (door) that welcomes client's contact

Client contacts server by:

- creating client-local TCP socket
- specifying IP address, port number of server process

When client creates socket: client TCP establishes connection to server TCP

When contacted by client, server TCP creates new socket for server process to communicate with client

- allows server to talk with multiple clients

UDP: no “connection” between client and server

- no handshaking
- sender explicitly attaches IP address and port of destination
- server must extract IP address, port of sender from received datagram

UDP: transmitted data may be received out of order, or lost

Client/server socket interaction: TCP

Server (running on `hostid`)

create socket,
port=`x`, for
incoming request:
`welcomeSocket =`
`ServerSocket()`

wait for incoming
connection request
`connectionSocket =`
`welcomeSocket.accept()`

read request from
`connectionSocket`

write reply to
`connectionSocket`

close
`connectionSocket`

Client

create socket,
connect to `hostid`, port=`x`
`clientSocket = Socket()`

send request using
`clientSocket`

read reply from
`clientSocket`

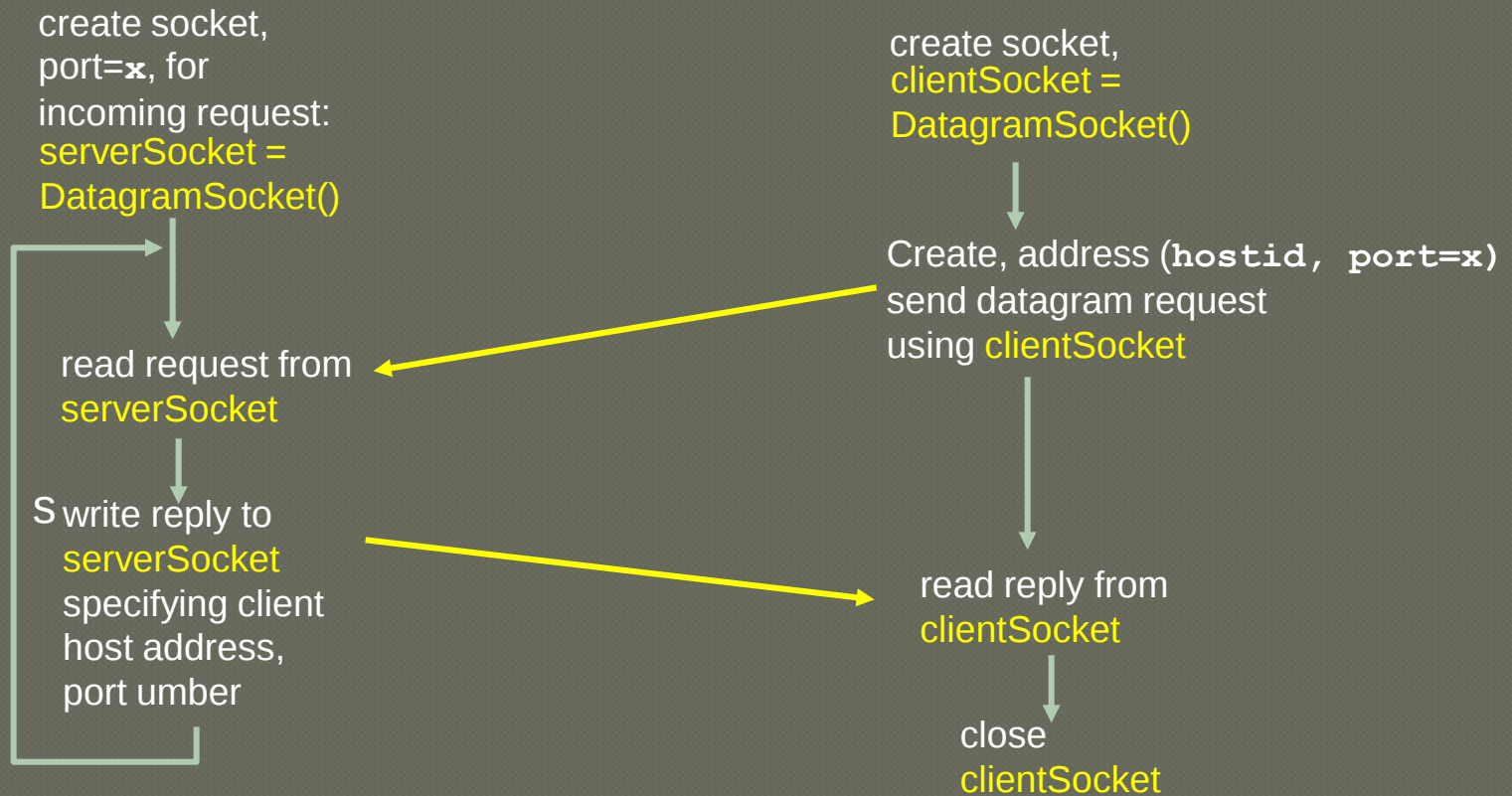
close
`clientSocket`

TCP
connection setup

Client/server socket interaction: UDP

Server (running on `hostid`)

Client



Berkeley UNIX Sockets for TCP

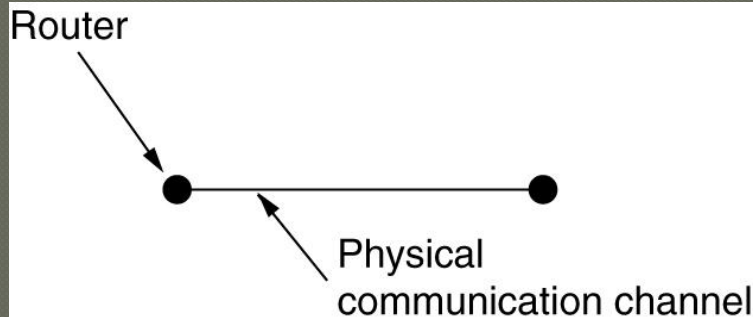
The socket primitives for TCP.

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

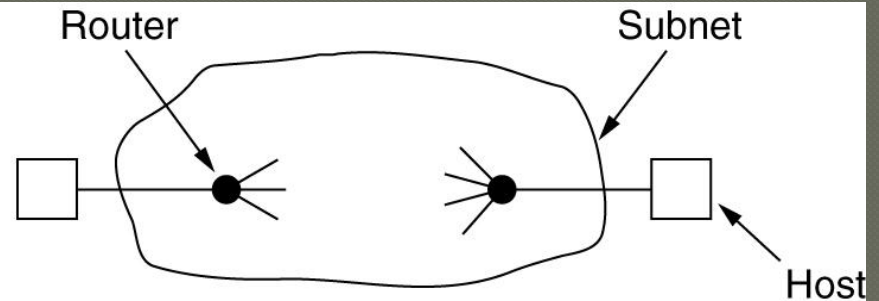
Elements of Transport Protocols

- Addressing
- Connection Establishment
- Connection Release
- Flow Control and Buffering
- Multiplexing
- Crash Recovery

Transport Protocol



(a)



(b)

(a) Environment of the data link layer.

- Routers communicate via a physical channel

(b) Environment of the transport layer.

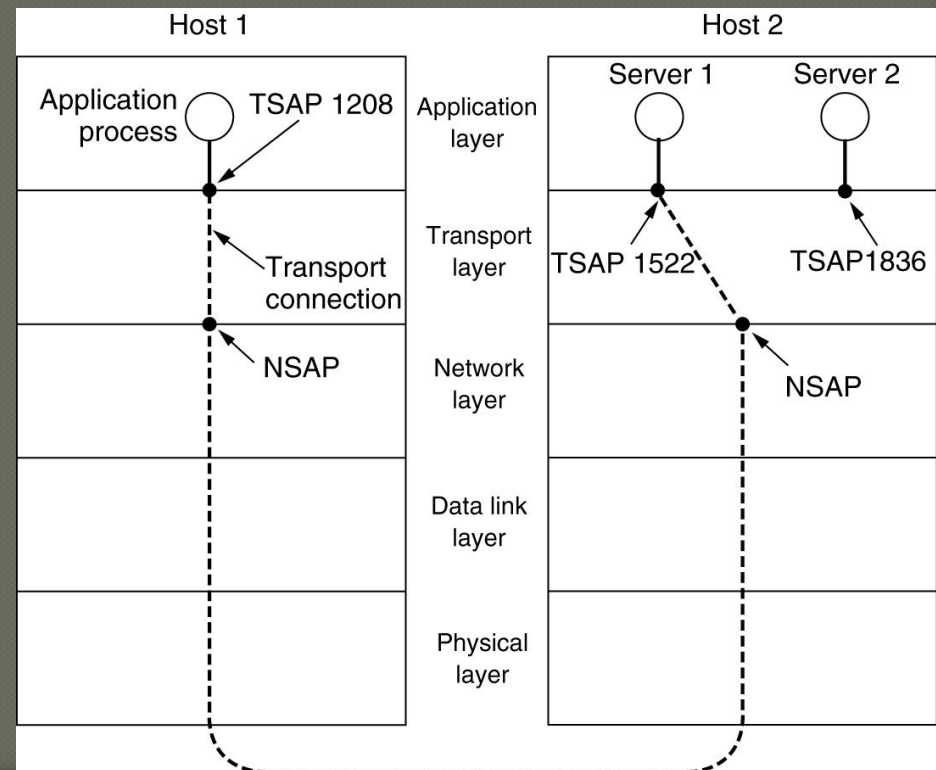
- Process communication through the subnet.

Addressing

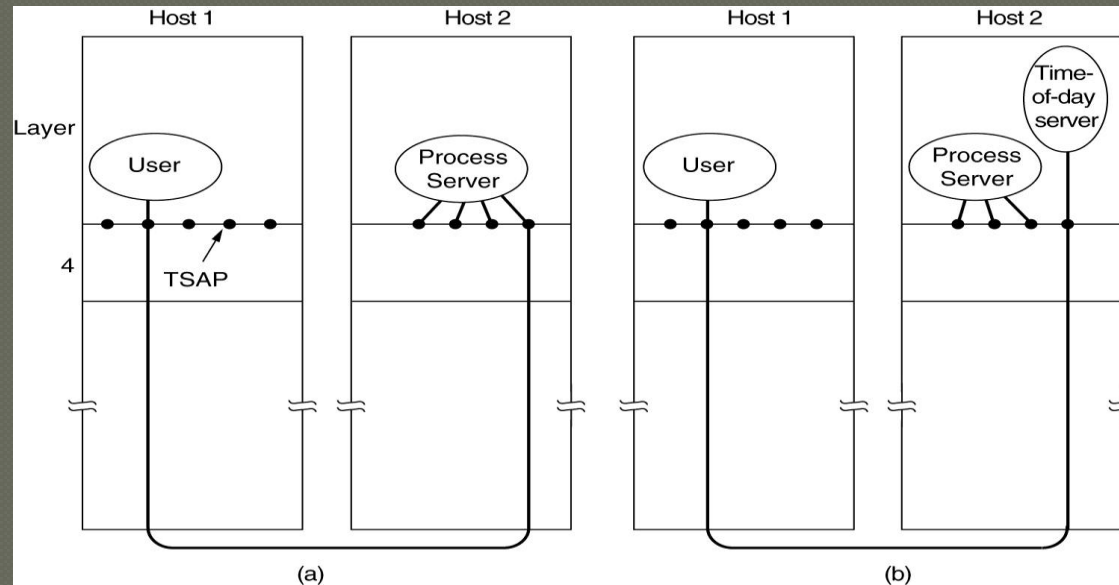
- TSAPs (Transport Service Access Point) or end points or simply ports

- NSAPs (Network Service Access Point) in the network layer
 - i.e. IP address is an example of NSAPc

- Application processes, both client and server, can attach themselves to a TSAP to establish a connection to a remote TSAP.
- 'Attach' is implemented in the OS, (ex. call LISTEN in Unix)
- Connections run through NSAPS on each host.
- Multiple TSAPs share the same NSAP on a host.

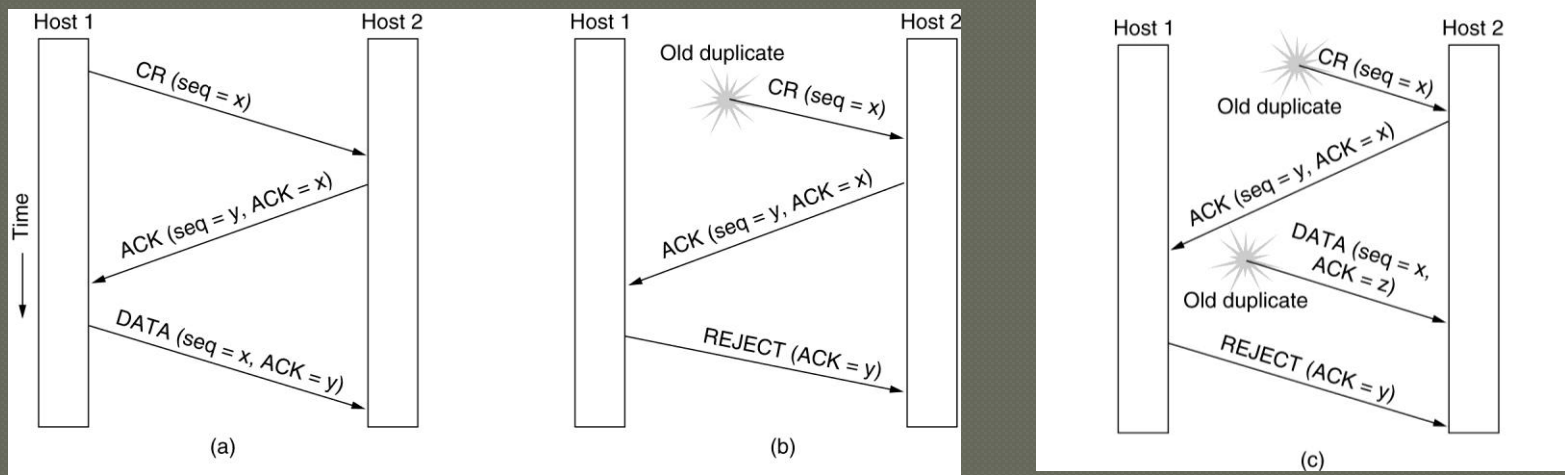


Connection Establishment



How a user process in host 1 establishes a connection with a time-of-day server in host 2.

Connection Establishment (2)



Sequence Number and clock time are used to avoid that old requests might be processed as new.

Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST.

(a) Normal operation,

(b) Old CONNECTION REQUEST appearing out of nowhere.

(c) Duplicate CONNECTION REQUEST and duplicate ACK.

Next Time

- ◉ Connection Release
- ◉ Flow Control
- ◉ Multiplexing

- ◉ UDP Protocol