CS3200: Computer Networks

Lecture 25

IIT Palakkad

14 Oct, 2019

Transport Layer

- Together with the network layer, the transport layer is the heart of the protocol hierarchy.
- The network layer provides end-to-end packet delivery using datagrams or virtual circuits.
- The transport layer builds on the network layer to provide data transport from a process on a source machine to a process on a destination machine with a desired level of reliability that is independent of the physical networks currently in use.

Services Provided to the Upper Layers

- The ultimate goal of the transport layer is to provide efficient, reliable, and cost-effective data transmission service to its users, normally processes in the application layer.

- To achieve this, the transport layer makes use of the services provided by the network layer. The software and/or hardware within the transport layer that does the work is called the **transport entity**. The transport entity can be located in the operating system kernel, in a library package bound into network applications, in a separate user process, or even on the network interface card.

- If the transport layer service is so similar to the network layer service, why are there two distinct layers? Why isn't one layer adequate?

The transport code runs entirely on the users' machines, but the network layer mostly runs on the routers, which are operated b

Berkeley Sockets

Sockets were first released as part of the Berkeley UNIX 4.2BSD software distribution in 1983. They quickly became popular. The primitives are now widely used for Internet programming on many operating systems, especially UNIX-based systems, and there is a socket-style API for Windows called "winsock."

Primitive	Meaning
SOCKET	Create a new communication endpoint
BIND	Associate a local address with a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Passively establish an incoming connection
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

Berkeley Sockets

- SOCKET primitive creates a new endpoint and allocates table space for it within the transport entity. A successful SOCKET call returns an ordinary file descriptor for use in succeeding calls, the same way an OPEN call on a file does.
- Newly created sockets do not have network addresses. These are assigned using the BIND primitive. Once a server has bound an address to a socket, remote clients can connect to it.
- LISTEN, a non-blocking call, allocates space to queue incoming calls for the case that several clients try to connect at the same time.

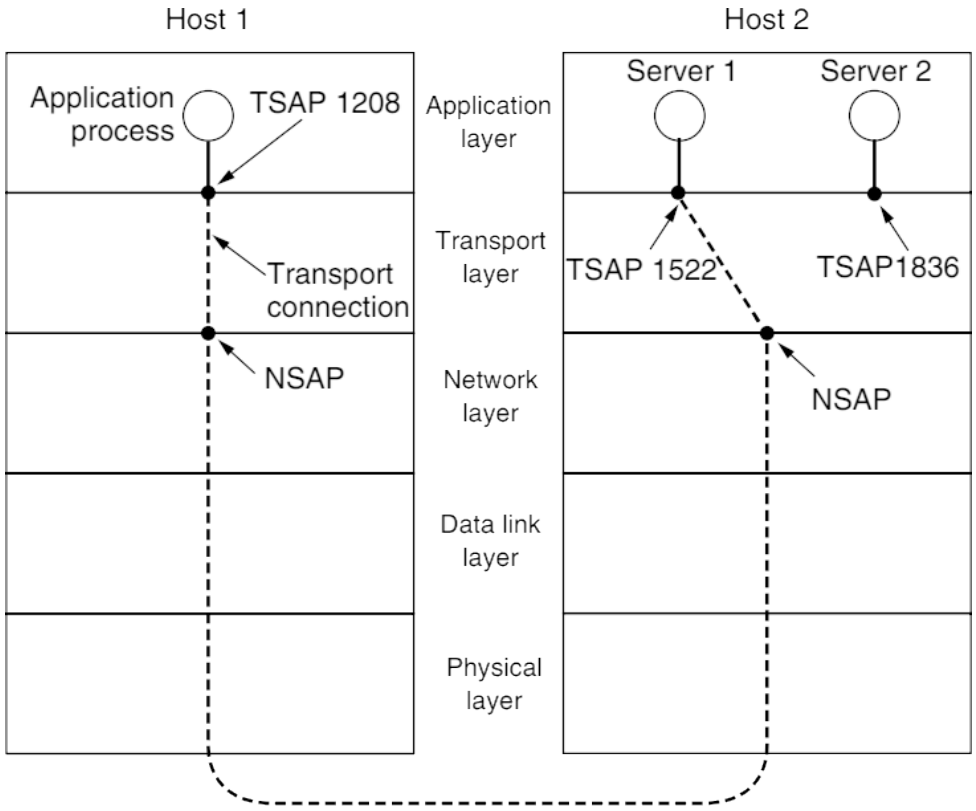
we need LISTEN as when we are in while loop, inside we might not be able to do instruction for the next connection which will

Berkeley Sockets

- To block waiting for an incoming connection, the server executes an ACCEPT primitive. When a segment asking for a connection arrives, the transport entity creates a new socket with the same properties as the original one and returns a file descriptor for it. The server can then fork off a process or thread to handle the connection on the new socket and go back to waiting for the next connection on the original socket.
- The CONNECT primitive blocks the caller and actively starts the connection process.
- Both sides can now use SEND and RECEIVE to transmit and receive data over the full-duplex connection.
- When both sides have executed a CLOSE primitive, the connection is released.

Addressing

- When an application (e.g., a user) process wishes to set up a connection to a remote application process, it must specify which one to connect to.
- In the Internet, these endpoints are called **ports**. We will use the generic term **TSAP (Transport Service Access Point)**.

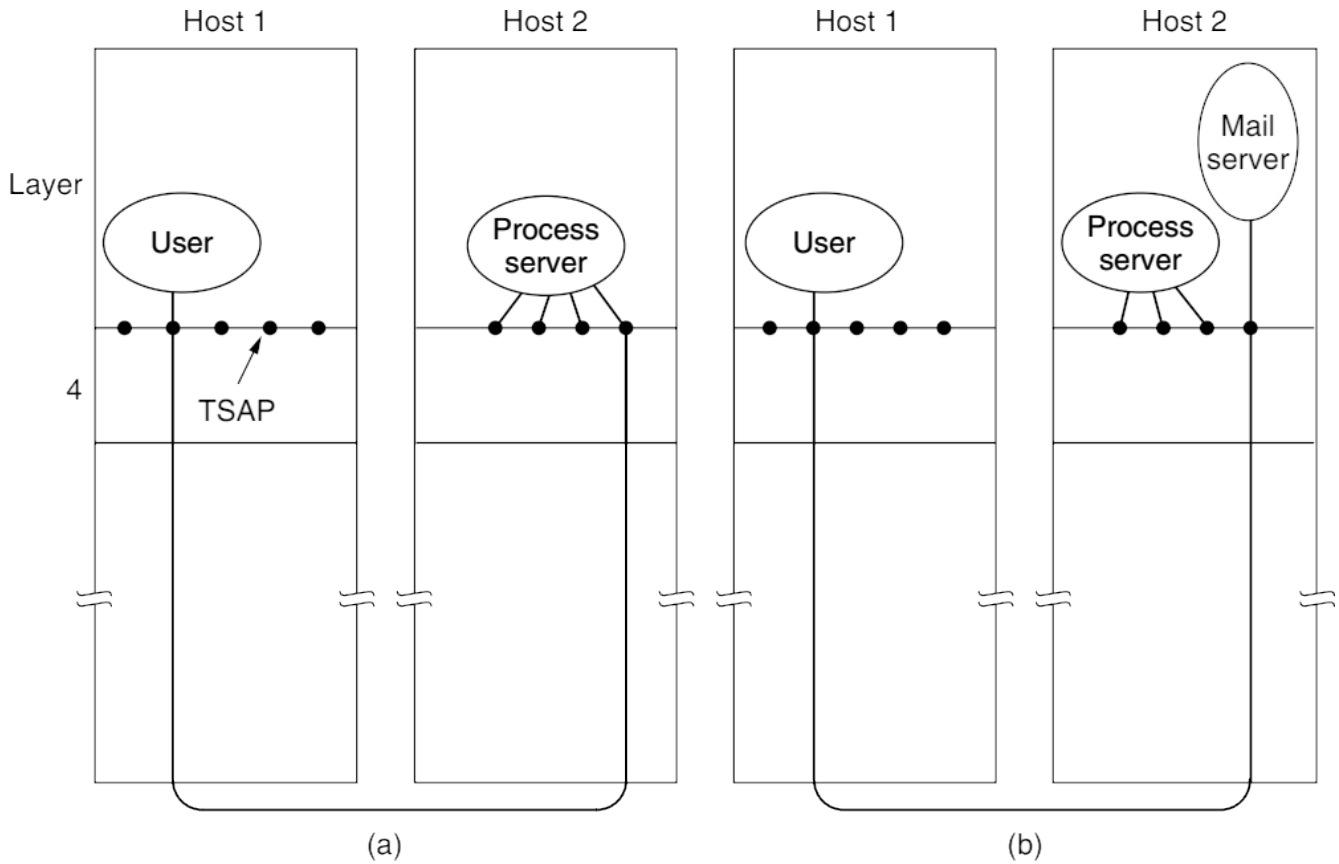


What if you do not know the address?

- To find the TSAP address corresponding to a given service name, such as "BitTorrent," a user sets up a connection to a special process known an **portmapper**.
- The user then sends a message specifying the service name, and the portmapper sends back the TSAP address.
- Then the user releases the connection with the portmapper and establishes a new one with the desired service.

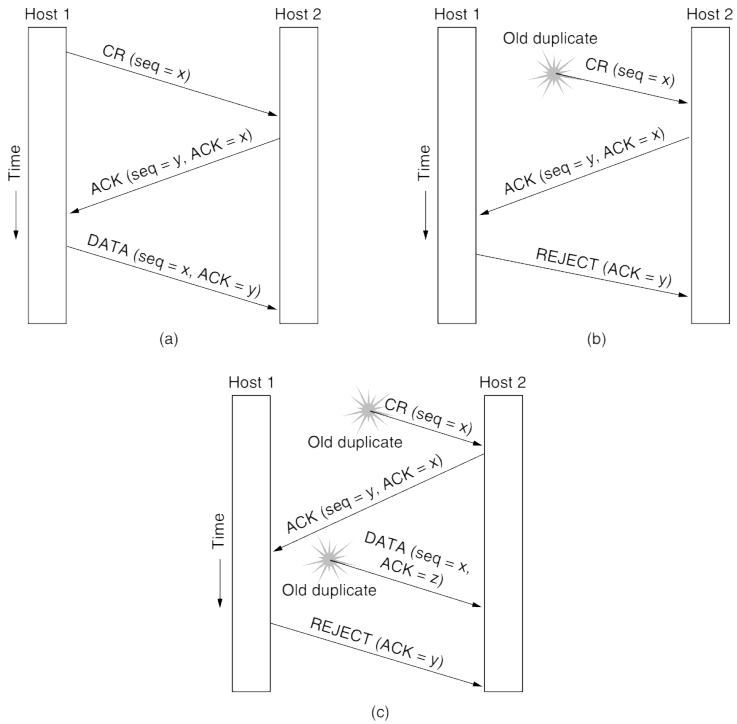
Initial Connection Protocol

Instead of every conceivable server listening at a well-known TSAP, each machine that wishes to offer services to remote users has a special process server that acts as a proxy for less heavily used servers. This server is called *inetd* on UNIX systems.



Connection Establishment

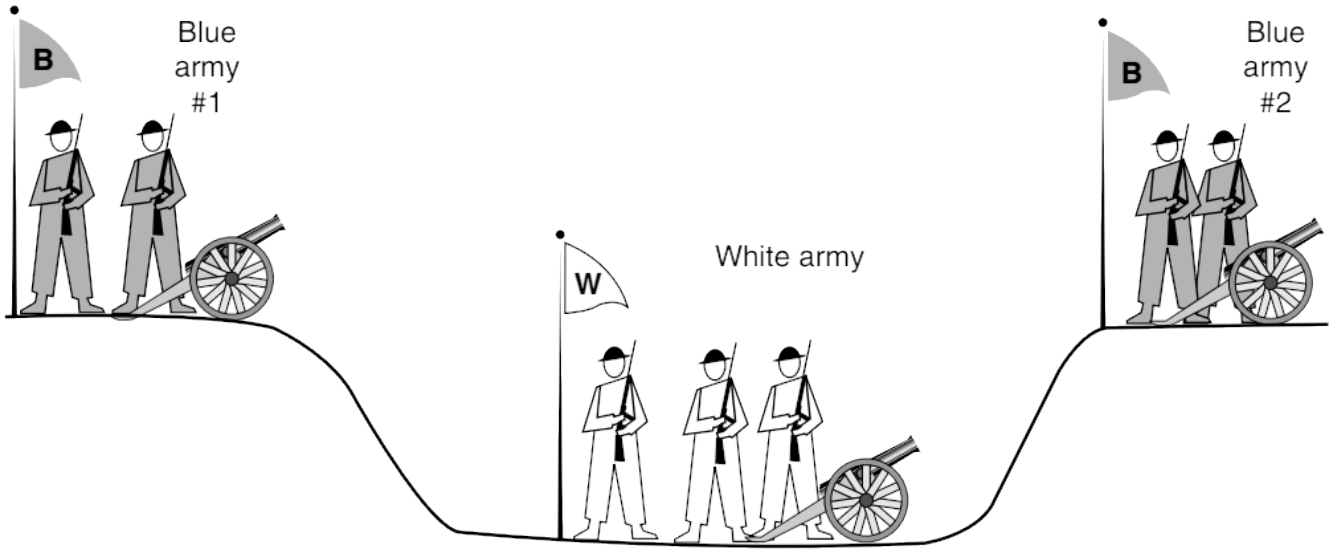
- Send CONNECTION REQUEST segment to the destination and wait for a CONNECTION ACCEPTED reply. What is the problem with this?
- Tomlinson (1975) introduced the **three-way handshake**. This establishment protocol involves one peer checking with the other that the connection request is indeed current.



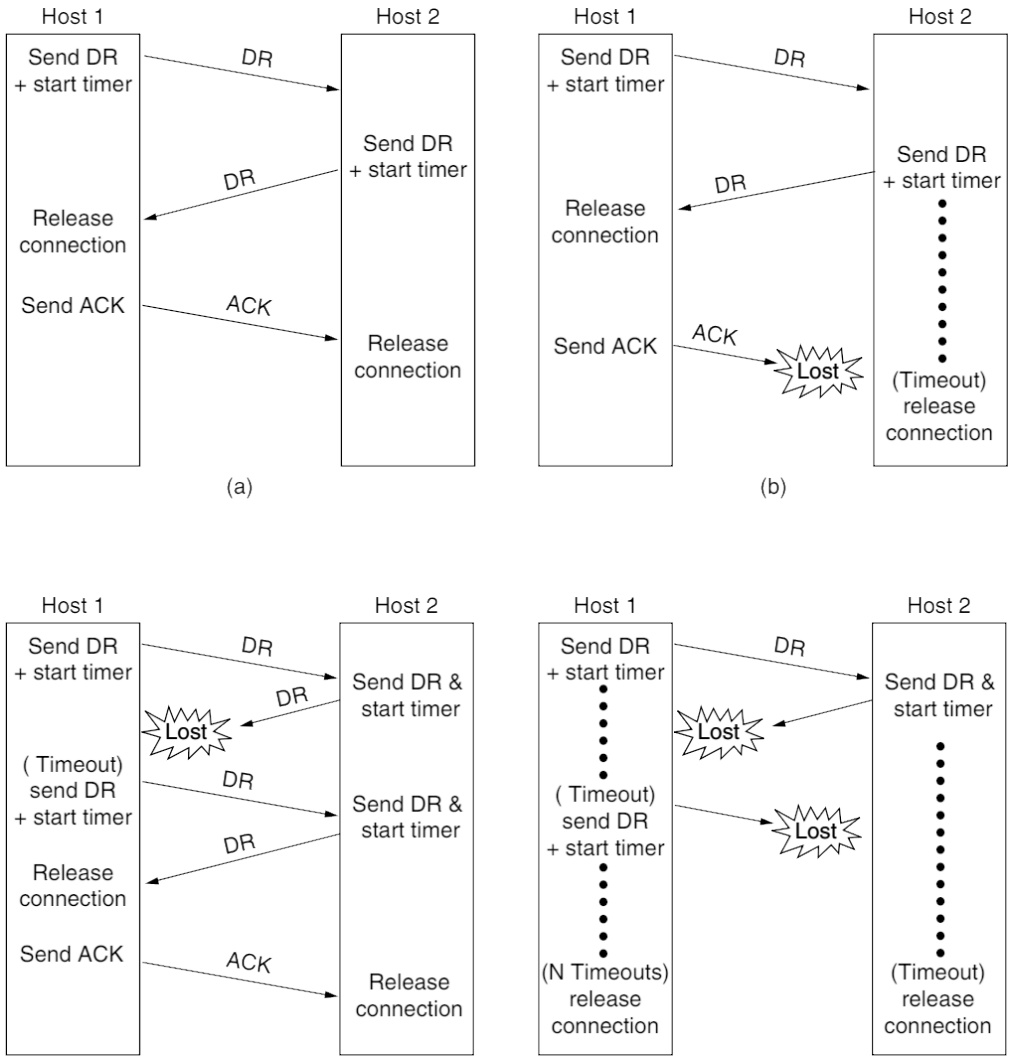
Connection Release

- Releasing a connection is easier than establishing one. There are two styles of terminating a connection: asymmetric release and symmetric release.
- Asymmetric release is the way the telephone system works: when one party hangs up, the connection is broken.
- Symmetric release treats the connection as two separate unidirectional connections and requires each one to be released separately.

The two-army problem



A Disconnection Protocol



Reading Assignments

- Section 6.2.4 of Tanenbaum on Error and Flow Control
- Section 6.2.5 of Tanenbaum on Multiplexing
- Section 6.2.6 of Tanenbaum on Crash Recovery