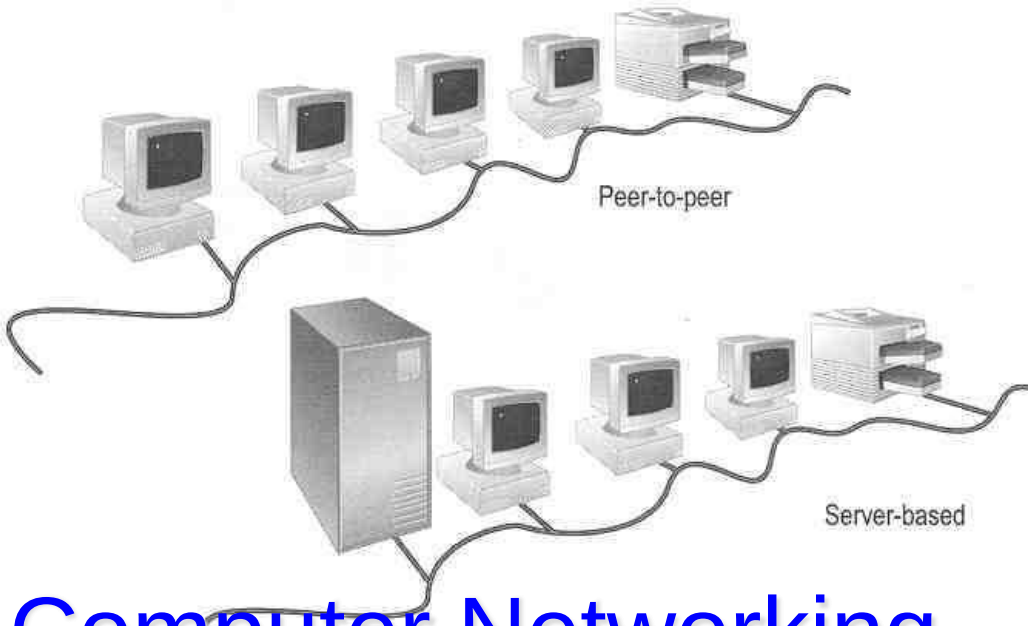


Computer Networking
Lecture 4
Hassan Alamri

Agenda:

- Peer to Peer Network
- Transport Layer
- Transport layer protocols:
  - TCP (Reliable data delivery)
  - UDP
- Congestion Control
- Conclusion

What is P2P systems?

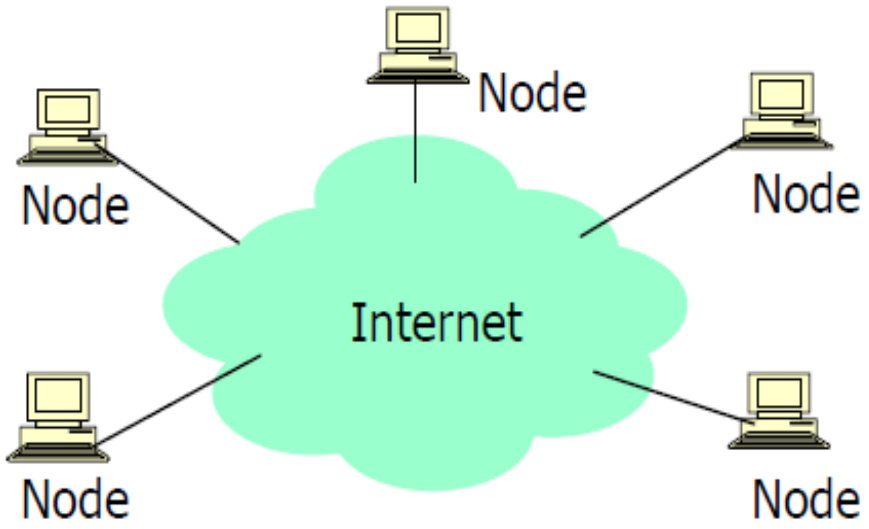
❖ P2P computing is the sharing of computer resources and services by direct exchange between systems.

❖ These resources and services include the exchange of information, processing cycles, cache storage, and disk storage for files.

❖ P2P refers to applications that take advantage of resources (storage, cycles, content, human presence) available at the edges of the internet

P2P Architecture

- All nodes are both clients and servers
  - Provide and consume data
  - Any node can initiate a connection
- No centralized data source
  - "The ultimate form of democracy on the Internet"
  - "The ultimate threat to copy-right protection on the Internet"



P2P Network Characteristics

- Clients are also servers and routers
  - Nodes contribute content, storage, memory, CPU
- Nodes are autonomous (no administrative authority)
- Network is dynamic: nodes enter and leave the network "frequently"
- Nodes collaborate directly with each other (not through well-known servers)
- Nodes have widely varying capabilities

P2P Benefits

- Efficient use of resources
  - Unused bandwidth, storage, processing power at the edge of the network
- Scalability
  - Consumers of resources also donate resources
  - Aggregate resources grow naturally with utilization
- Reliability
  - Replicas
  - Geographic distribution
  - No single point of failure
- Ease of administration
  - Nodes self organize
  - No need to deploy servers to satisfy demand (c.f. scalability)
  - Built-in fault tolerance, replication, and load balancing

P2P Applications

- Are these P2P systems?
  - File sharing (Napster, Gnutella, Kazaa)
  - Multiplayer games (Unreal Tournament, DOOM)
  - Collaborative applications (ICQ, shared whiteboard)
  - Distributed computation (Seti@home)
  - Ad-hoc networks

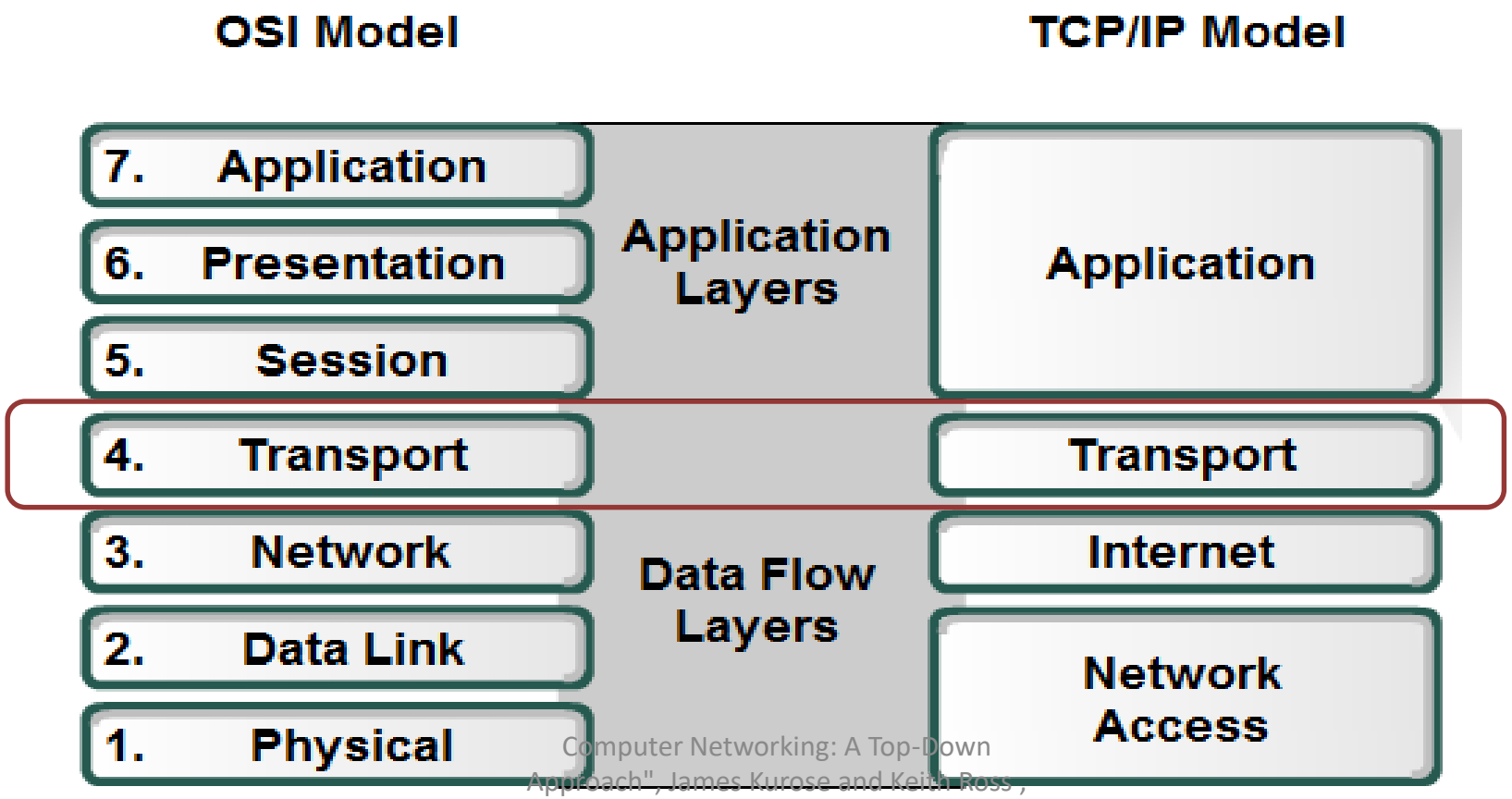
Popular P2P Systems

- Napster, Gnutella, Kazaa, Freenet

- Large scale sharing of files.
  - User A makes files (music, video, etc.) on their computer available to others
  - User B connects to the network, searches for files and downloads files directly from user A

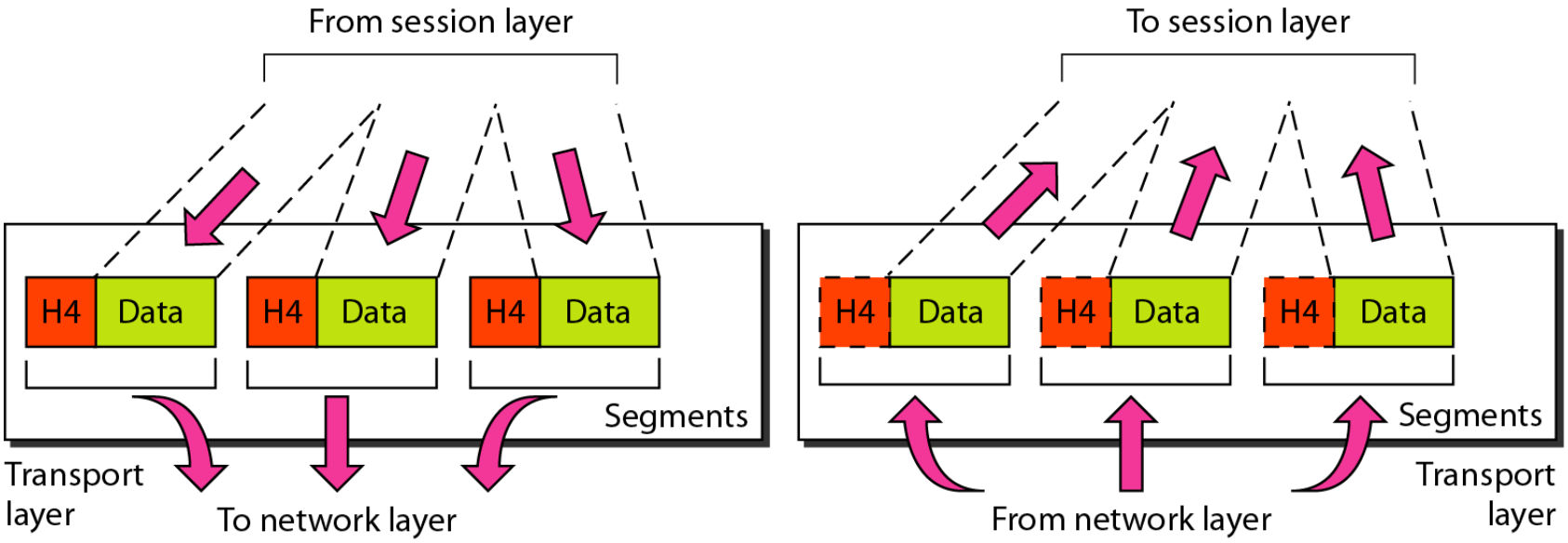
Transport(Layer 4)

The transport layer is responsible for the delivery of a message from one process to another process.



Transport Layer responsibilities

- Segmentation and reassembly. A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.



- Connection control.

The transport layer can be either connectionless or connection oriented.

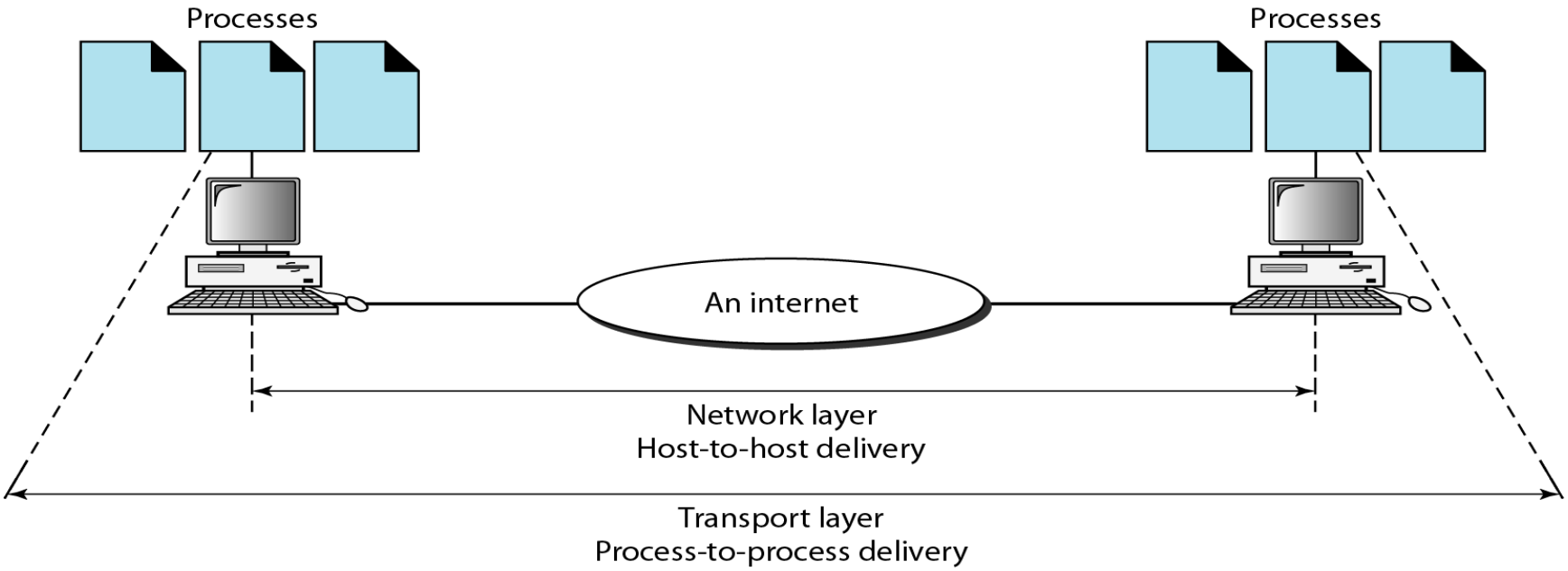
- ✓ A **connectionless transport layer** treats each segment independently and delivers it to the transport layer at the destination machine.
- ✓ A **connection oriented transport layer** makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred , the connection is terminated.

- Flow control.

the transport layer is responsible for end to end flow control. However, flow control at this layer is performed end to end rather than across a single link.

➢ Error control.

the transport layer is responsible for error control. However, error control at this layer is performed process-to process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission.



Transport Layer Protocols

A. Transfer Control Protocol (TCP):

- Connection-Oriented delivery service
- Reliable connection (Acknowledgement)
- Full-duplex
- Handle flow control
- Costly, slower than UDP
- The data is transmitted in segments

Connection-oriented meaning

Connection-oriented means that a connection (Virtual Circuit) must be established before hosts can exchange data. Reliability is achieved by assigning a sequence number to each segment transmitted. An acknowledgement is used to verify that the data was received by the other host. For each segment sent, the receiving host must return an acknowledgement (ACK) within a specified period for bytes received. If an ACK is not received, the data is retransmitted. TCP uses byte-stream communications, wherein data within the TCP segment is treated as a sequence of bytes with no record or field boundaries.

TCP 3-way handshake

A TCP connection is initialized through a three-way handshake. The purpose of the three-way handshake is to synchronize the sequence number and acknowledgement numbers of both sides of the connection and exchange TCP Window sizes. The following steps outline the process:

TCP handshake step 1

The client sends a TCP segment to the server with an initial Sequence Number for the connection and a Window size indicating the size of a buffer on the client to store incoming segments from the server.

TCP handshake step 2

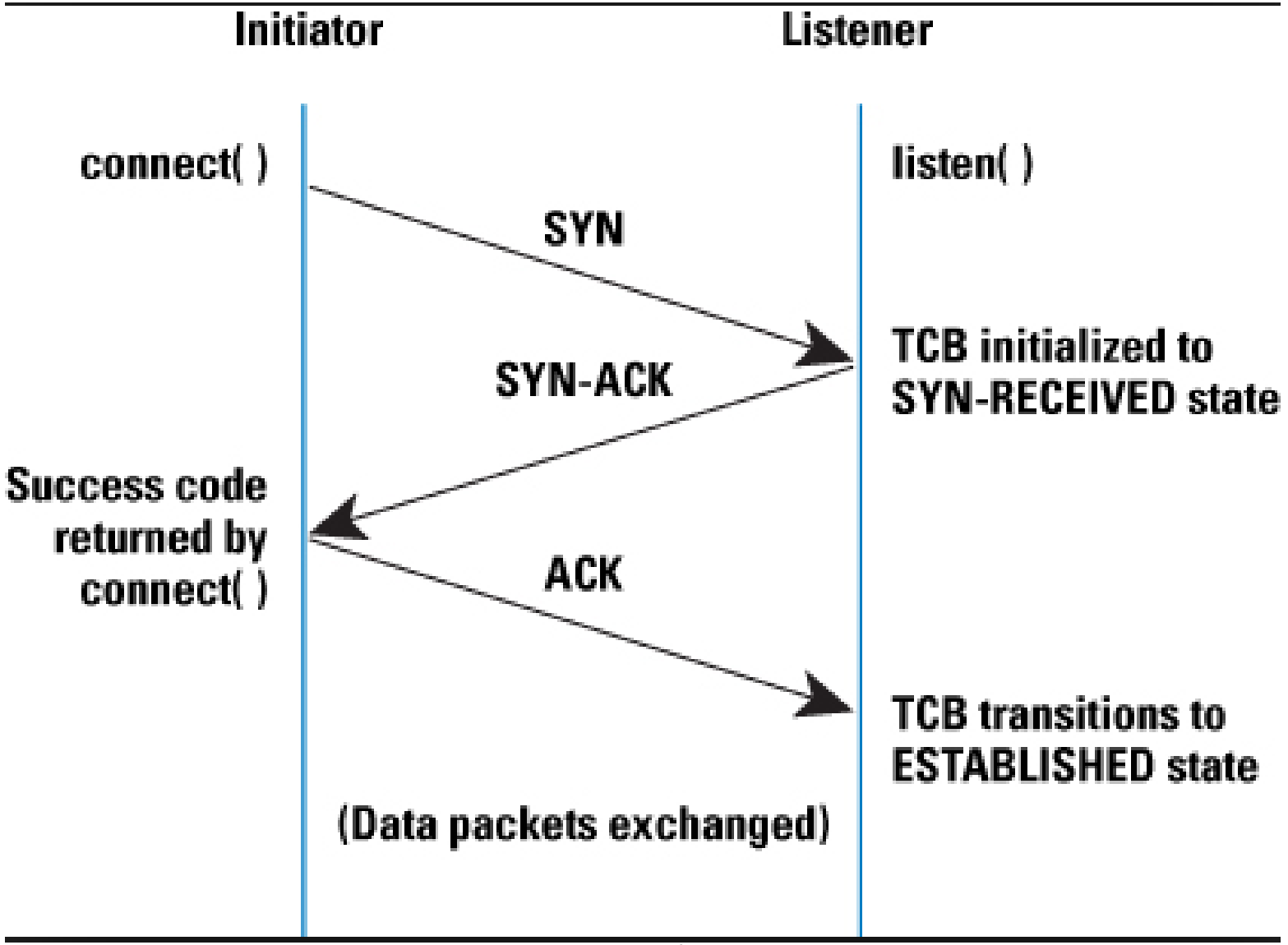
The server sends back a TCP segment containing its chosen initial Sequence Number, an acknowledgement of the client's Sequence Number, and a Window size indicating the size of a buffer on the server to store incoming segments from the client.

TCP handshake step 3

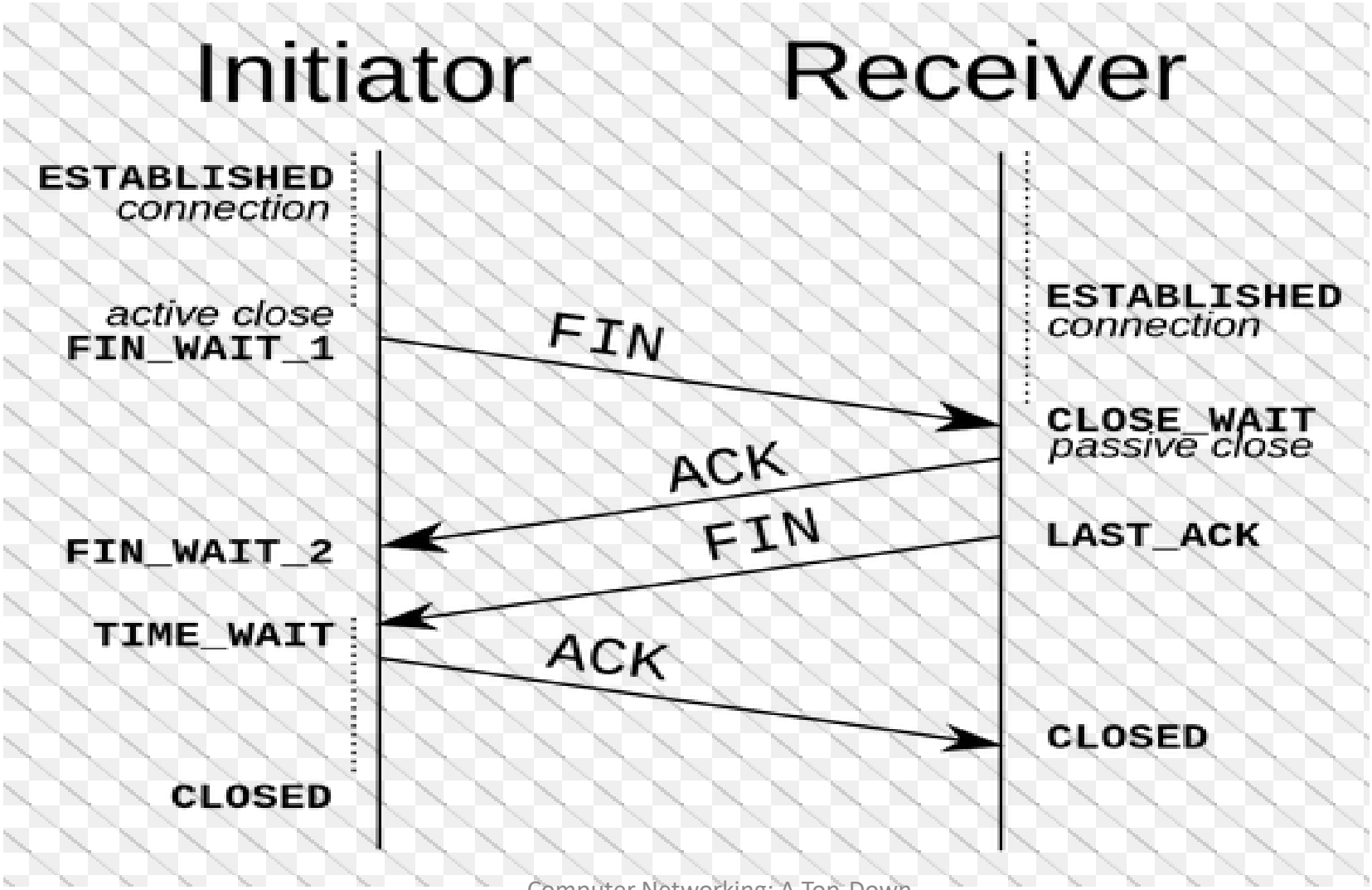
The client sends a TCP segment to the server containing an acknowledgement of the server's Sequence Number.

N.B. *TCP uses a similar handshake process to end a connection. This guarantees that both hosts have finished transmitting and that all data was received.*

TCP Open Connection



TCP close Connection



TCP header

Field	Function
Source Port	**TCP port of sending host.**
Destination Port	**TCP port of destination host.**
Sequence Number	**Sequence number of the first byte of data in the TCP segment.**
Acknowledgement Number	**Sequence number of the byte the sender expects to receive next from the other side of the connection.**
Window	**Current size of a TCP buffer on the host sending this TCP segment to store incoming segments.**
TCP Checksum	**Verifies the integrity of the TCP header and the TCP data.**

Well Known TCP ports 1-1023

TCP Port Number	Description
20	FTP (Data Channel)
21	FTP (Control Channel)
23	Telnet
80	HTTP used for the World Wide Web
139	NetBIOS session service

B. User Datagram Protocol (UDP):

- Connection less protocol
- Unreliable, give best effort delivery
- No error correction, no sequence
- Faster than TCP, less overhead
- Used by Video, audio
- Packet is forwarded using destination address inside it
- Different packets may take different paths
- UDP is used by applications that do not require an acknowledgement of receipt of data and that typically transmit small amounts of data at one time

UDP Header

Field	Function
Source Port	UDP port of sending host.
Destination Port	UDP port of destination host.
UDP Checksum	Verifies the integrity of the UDP header and the UDP data.

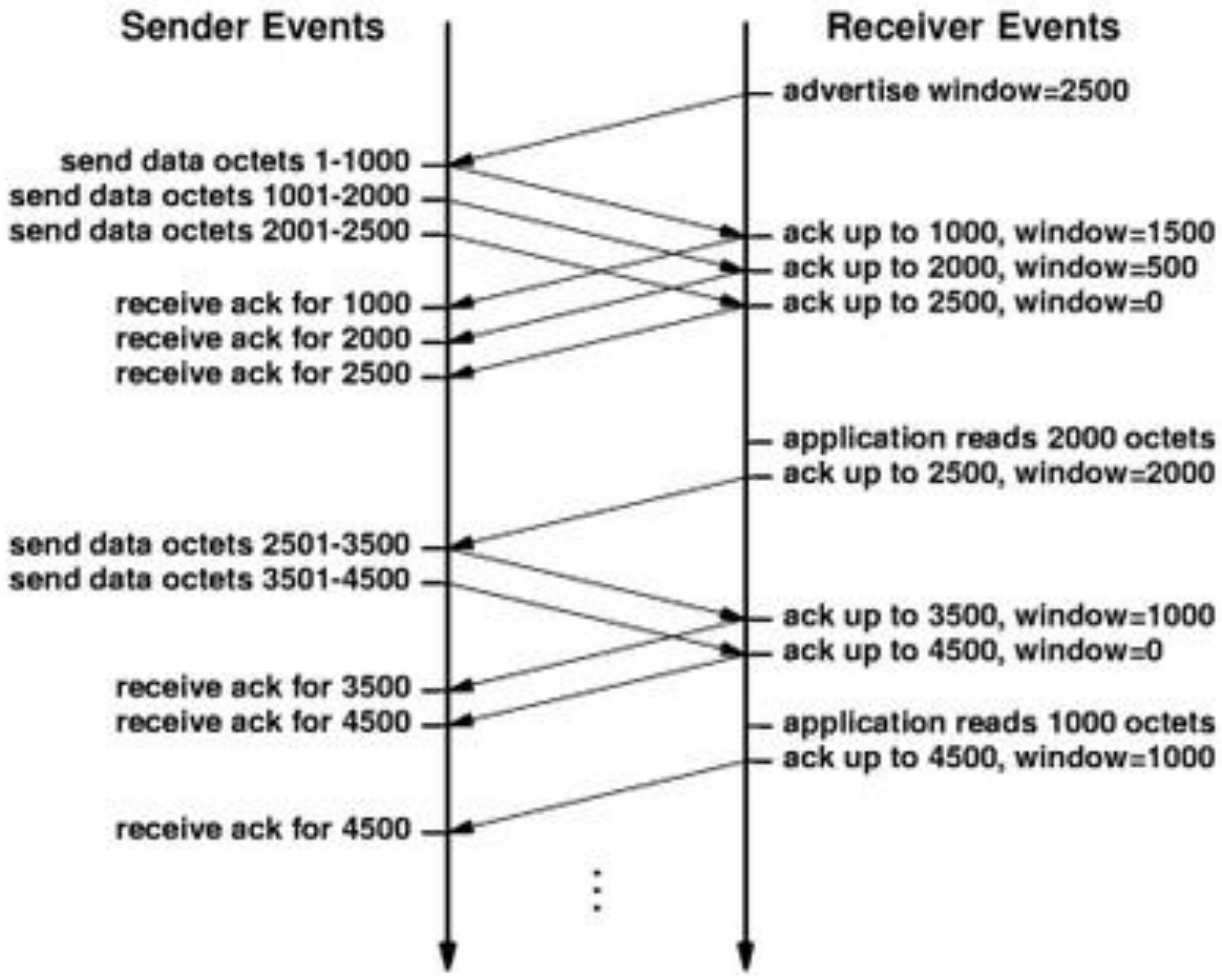
Well known UDP ports

UDP Port Number	Description
53	Domain Name System (DNS) Name Queries
69	Trivial File Transfer Protocol (TFTP)
137	NetBIOS name service
138	NetBIOS datagram service
161	SNMP

Congestion Control

TCP uses a congestion window in the sender side to do congestion avoidance. The congestion window indicates the maximum amount of data that can be sent out on a connection without being acknowledged. TCP detects congestion when it fails to receive an acknowledgement for a packet within the estimated timeout. In such a situation, it decreases the congestion window to one maximum segment size (MSS), and under other cases it increases the congestion window by one MSS. There also exists a congestion window threshold, which is set to half the congestion window size at the time when a re-transmit was required.

Congestion Control Window



Reference

- Computer Networking: A Top-Down Approach", James Kurose and Keith Ross , 5th edition