ELEN 4017

Network Fundamentals

Lecture 19

Purpose of lecture

Chapter 3: Transport Layer

- TCP sequencing
- TCP handshaking

TCP sequencing

- TCP views data as unstructured, but ordered stream of bytes.
- Sequence number of a segment is the **byte stream number** of the **first byte** in that segment.
- Consider file of size 500 000 bytes. MSS = 1000 bytes.
- 1st segment has seq # = 0
- 2nd segment has seq # = 1000
- 3rd segment has seq# = 2000 …. 500 segments are created.

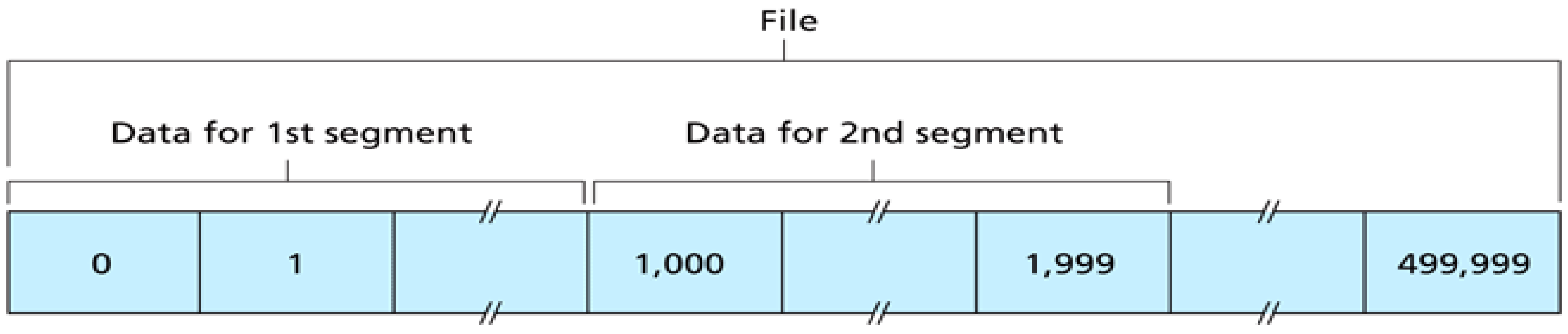
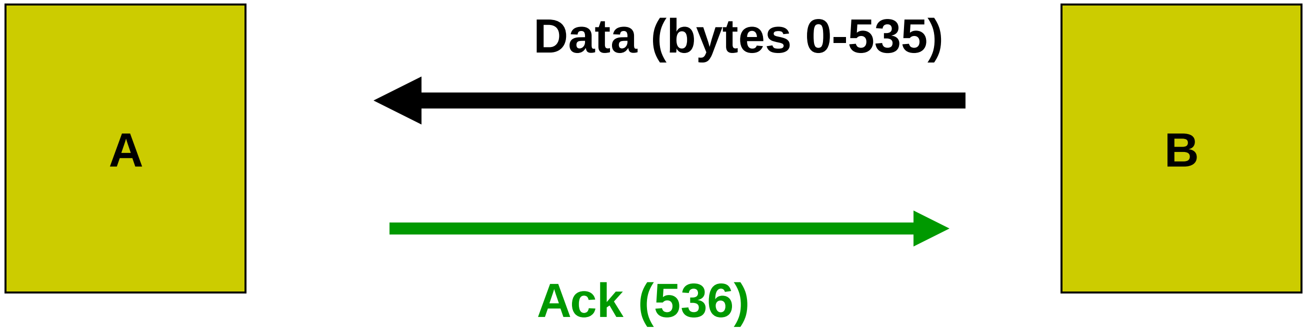


Figure 3.30 ♦ Dividing file data into TCP segments

Acknowledgement numbers

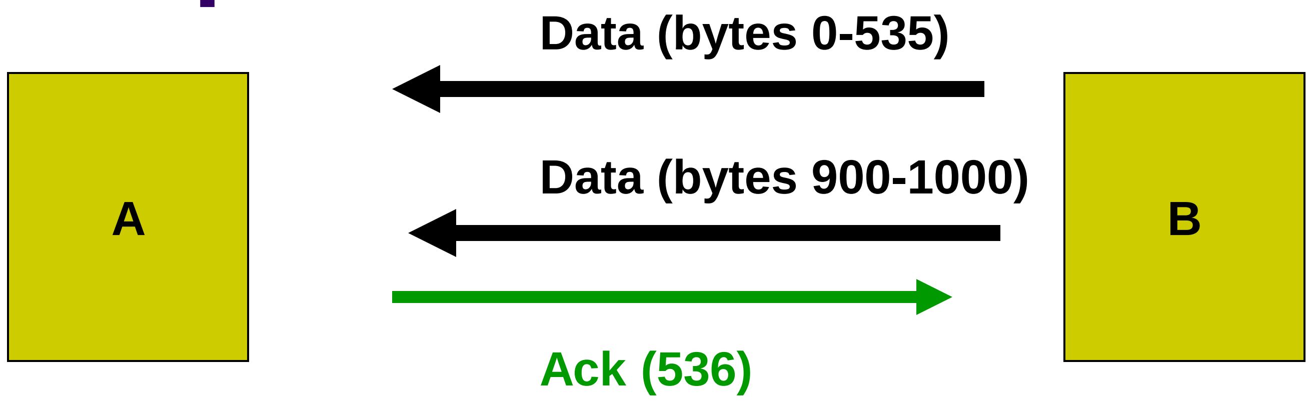
- TCP is full duplex.
- Consider host A and B are sending data to each other.
- Some data is sent from B to A and this data has a specific sequence number.
- The **ack number** used by Host A, is the **sequence number of the next byte** Host A is expecting **from Host B**.

Examples



- Host A has received bytes 0-535 in 1st segment successfully.
- The ack from A to B, is the next expected byte number → 536

Examples



- Host A has received bytes 0-535 in 1st segment and bytes 900-1000 in 2nd segment.
- For some reason bytes 536-899 not received.
- The ack from A to B, is the next expected byte number → 536.
- TCP only acknowledges bytes **up to the first missing byte in the stream**.
- Thus TCP is set to support **cumulative acknowledgements**.
- How could TCP support cumul. ack. but still not expect GBN behaviour from receiver ?

What does TCP do with out of order segment ?

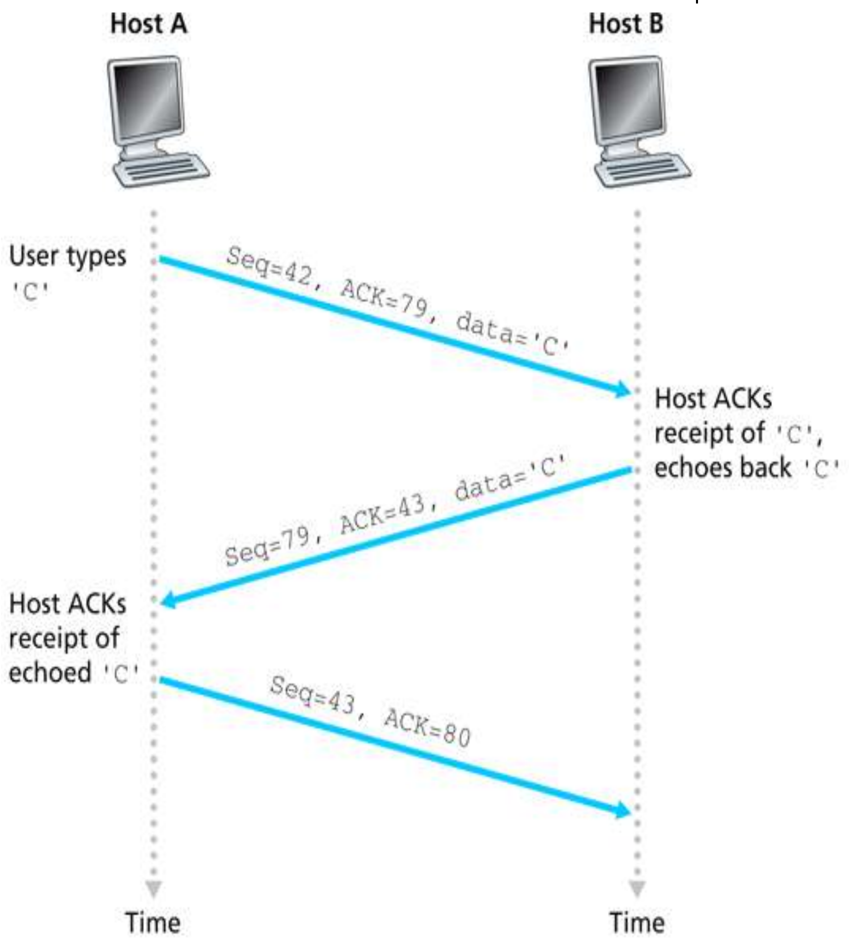
- TCP RFC does not impose any rules here. Its up to the implementer to decide.
- Either it can be discarded, or buffered.
- Most implementations store it and use later (similar to SR implementation).

Telnet example

- User starts a telnet session (remote login at console/command prompt)
- Telnet behaviour is the following:
- Each character entered by the user is echoed back to the terminal.
- Thus if you enter C, when you see displayed on the screen, its been received by the server and has been echoed back to the client.
- Thus 1 RTT occurs before character is displayed on screen.

Telnet example

- After login the **initial sequence number** of Host A is 42 and Host B is 79 (random assignment).
- The letter C is an ascii character, and thus is 1 byte in size.
- 3rd message sent has no data but specifies a sequence number.



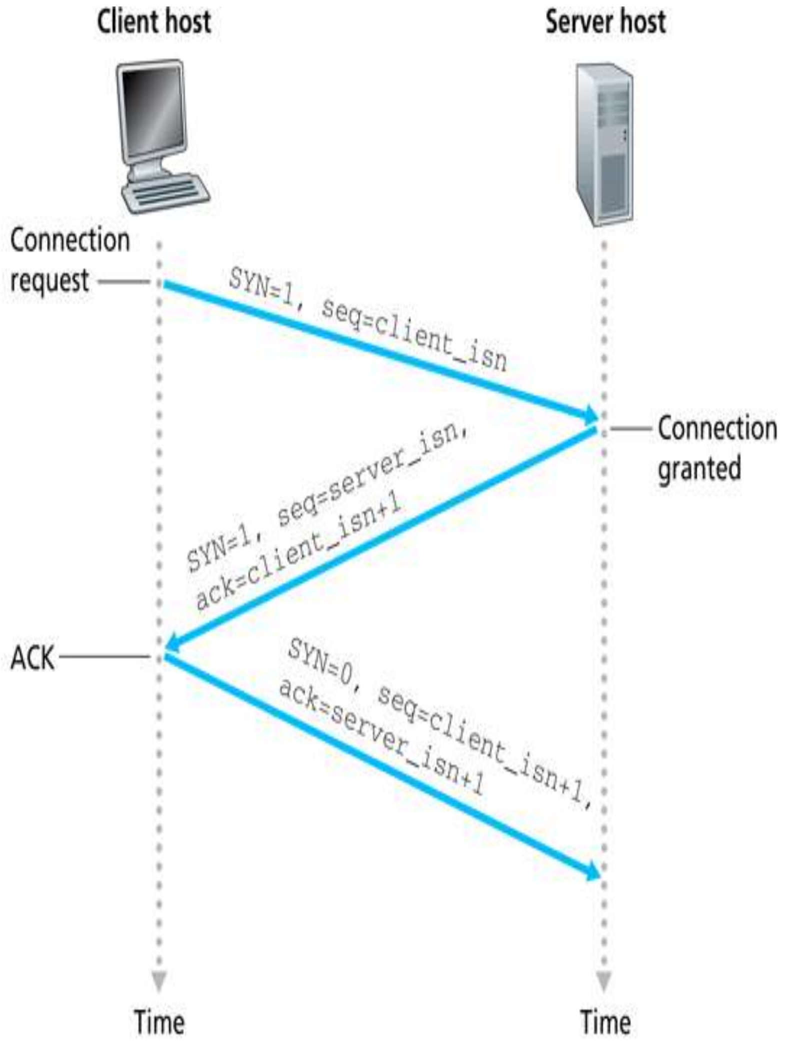
Purpose of lecture

Chapter 3: Transport Layer

- TCP sequencing
- TCP handshaking

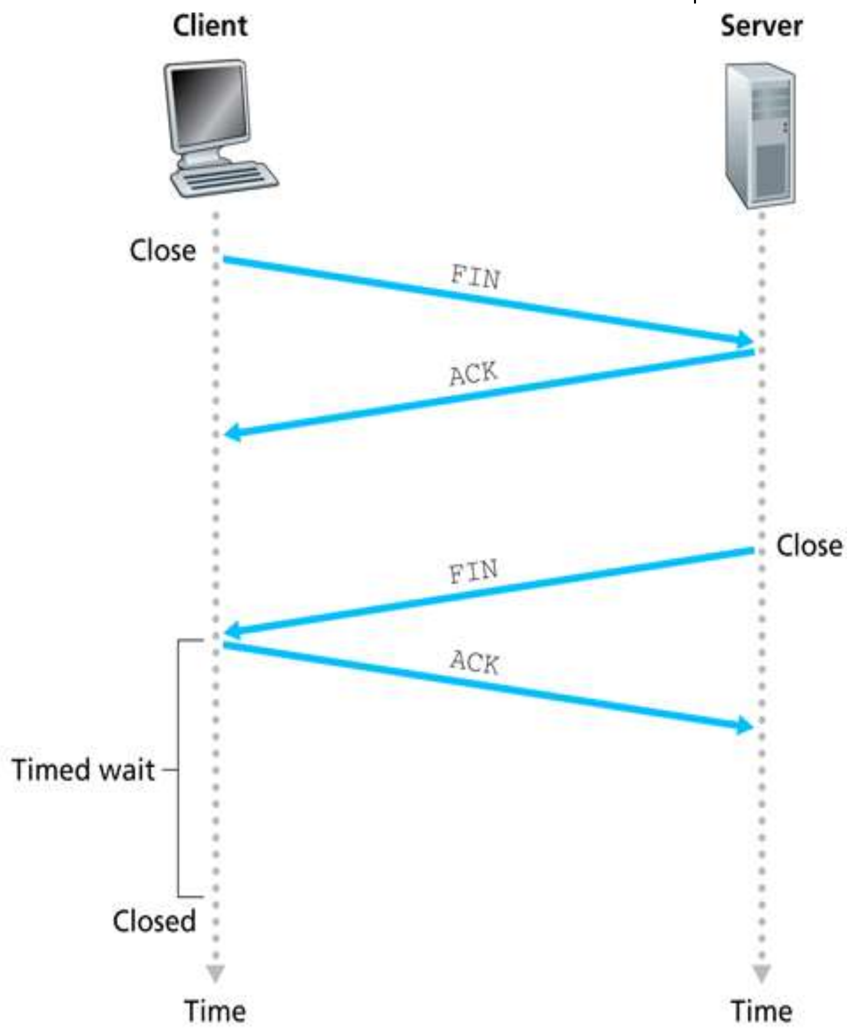
3 way handshake

- Client initiates connection by sending a segment with no payload. SYN bit = 1. **Random initial sequence number** is chosen (client_isn).
- Server receives TCP SYN, allocates buffers/variables and sends a connection granted segment (SYNACK) to Client. This also does not contain any payload.
- SYN=1, ACK field = client_isn+1. Server chooses its own initial sequence number (server_isn)
- Client receives SYNACK and allocates buffers/variables.
- Client then sends another segment to acknowledge the SYNACK with ACK=server_isn +1. SYN=0. This segment **can also contain client-to-server data.**

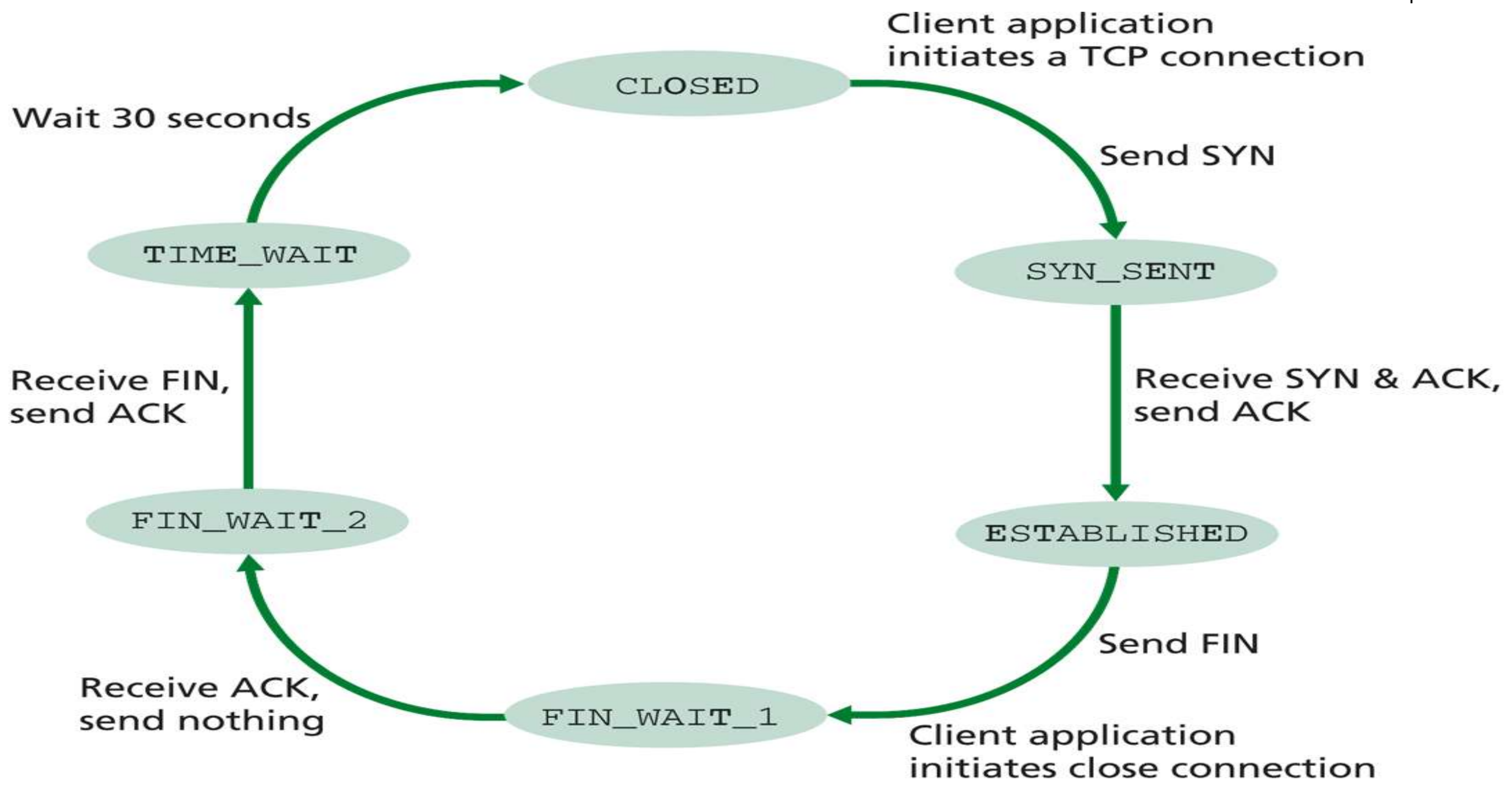


Closing a TCP connection

- Client closes connection by setting FIN=1
- Server responds with ACK for this segment.
- Server then sends its own shutdown segment with FIN=1.
- Finally the client acknowledges this and resources are de-allocated.
- Why is FIN needed on both sides?
- What could be the purpose of the TIMED_WAIT ?



States of TCP client side



Client application initiates a TCP connection
CLOSED
Send SYN
SYN_SENT
Receive SYN & ACK, send ACK
ESTABLISHED
Send FIN
Client application initiates close connection
FIN_WAIT_1
Receive ACK, send nothing
FIN_WAIT_2
Receive FIN, send ACK
TIME_WAIT
Wait 30 seconds

Figure 3.40 ◆ A typical sequence of TCP states visited by a client TCP

States of TCP server side

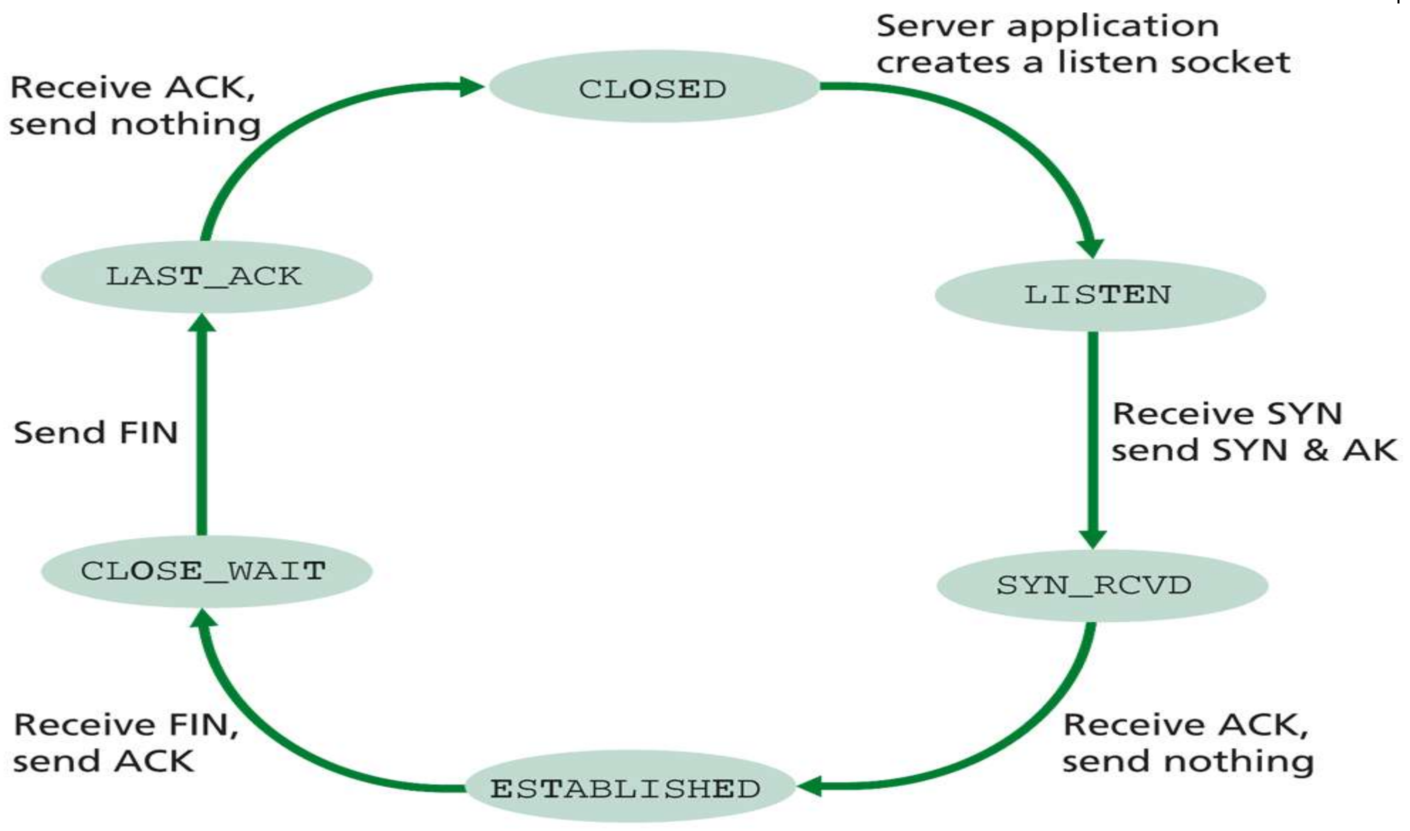

CLOSED
Server application creates a listen socket
LISTEN
Receive SYN send SYN & AK
SYN_RCVD
Receive ACK, send nothing
ESTABLISHED
Receive FIN, send ACK
CLOSE_WAIT
Send FIN
LAST_ACK
Receive ACK, send nothing

Figure 3.41 ◆ A typical sequence of TCP states visited by a server-side TCP

Accessing invalid ports

- What is the behaviour if the client attempts to establish a TCP connection with a server port for which there is no ongoing socket.
- The server will then send a special TCP segment with RST flag = 1.
- RST indicates “I don’t have a socket for that segment. Please don’t resend it”

Port scanning

- Consider a port scanner sends a TCP segment to server port 6789.
- 3 possible outcomes:
  - Server sends SYNACK → port is open and an app is running.
  - TCP RST segment is received → Segment reached target host but the host is not running an application on that port. Indicates that the port is open (i.e. not blocked by a firewall)
  - No response – SYN was blocked probably by a firewall.

SYN flood attack

- As we have seen, the server allocates variables and buffers when a SYN is received (1st part of 3 way handshake).
- This fact has been abused to create Denial of Service (DoS) attacks.
- DoS is an attempt to make a resource unavailable for its intended users.
- A Distributed DoS is where the attack originates from multiple clients.
- With the SYN flood attack, a large number of SYN messages are sent, without completing the handshake. Since the server allocates resources at this time, a server can very quickly become overloaded and will then deny connections from legitimate users.

SYN cookies

- To prevent such attacks SYN cookies were introduced.
- With SYN cookies, when a SYN is received no resources are assigned. Instead an initial TCP sequence number is created that is a function of source & destination IP addresses and ports, as well as a secret number.
- This carefully crafted number is called the cookie.
- The server responds with the sequence number to the client. **Importantly the cookie is not stored.**
- In the case of a legitimate client, an ACK is sent with the sequence number = SYNACK +1
- The server can then re-run the algorithm to regenerate the sequence number, and validate the session as legitimate.
- Of course, the next logical step in the attackers arsenal is to complete the handshake.