CS 88: Security and Privacy

18: Network Security @ The Transport Layer

11-15-2022

slides adapted from UC Berkeley, Jim Kurose, Kevin Webb

SWARTHMORE COLLEGE

Reading Quiz

The Internet

Global network of networks that ..

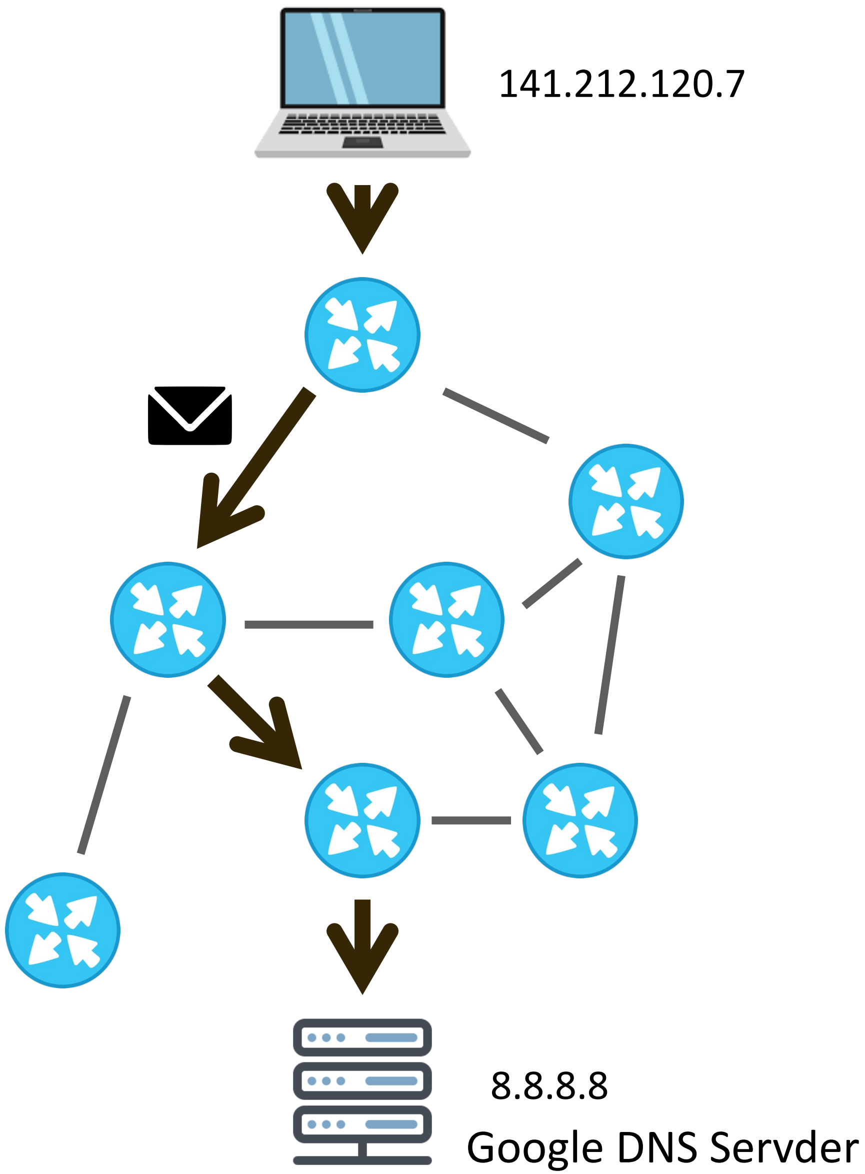
provides **best-effort** delivery of **packets** between connected hosts

Packet: a structured sequence of bytes

- Header: metadata used by network
- Payload: user data to be transported

Every host has a unique identifier — IP address

Series of routers receive packets, look at destination address on the header and send it one hop towards the destination IP address

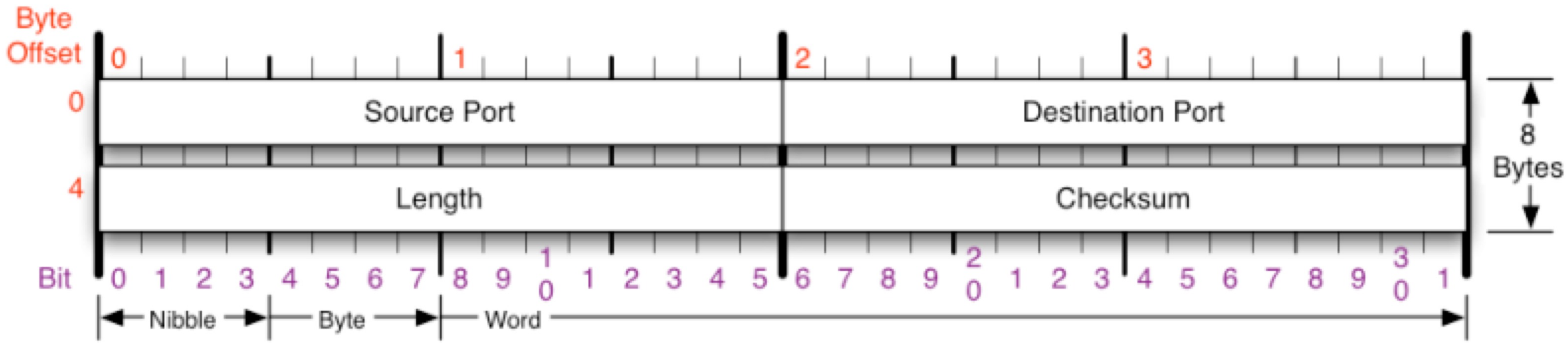


Network Protocols

We define how hosts communicate in published network protocols

Syntax: How communication is structured (e.g., format and order of messages)

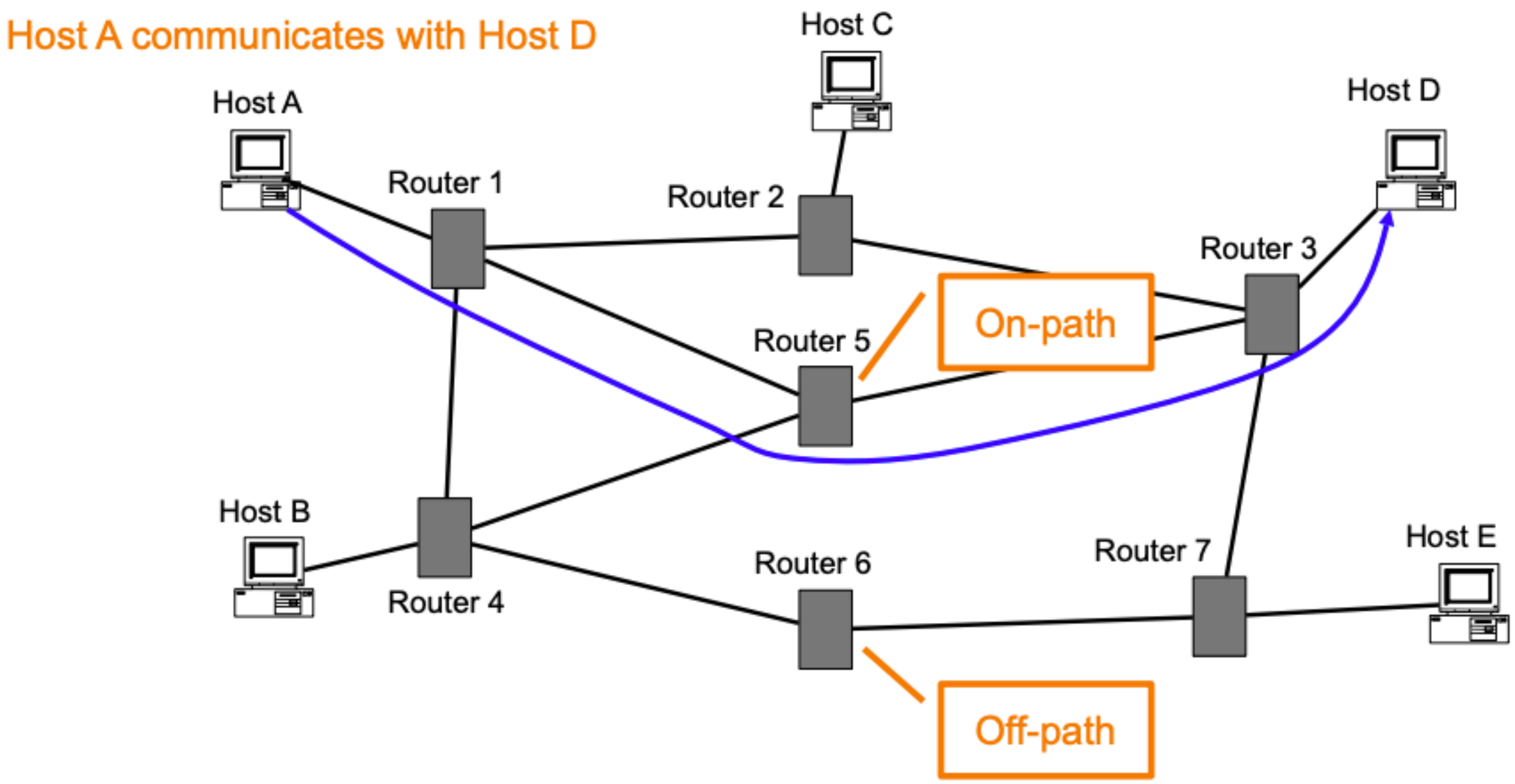
Semantics: What communication means. Actions taken on transmit or receipt of message, or when a timer expires. What assumptions can be made.



Example: What bytes contain each field in a packet header

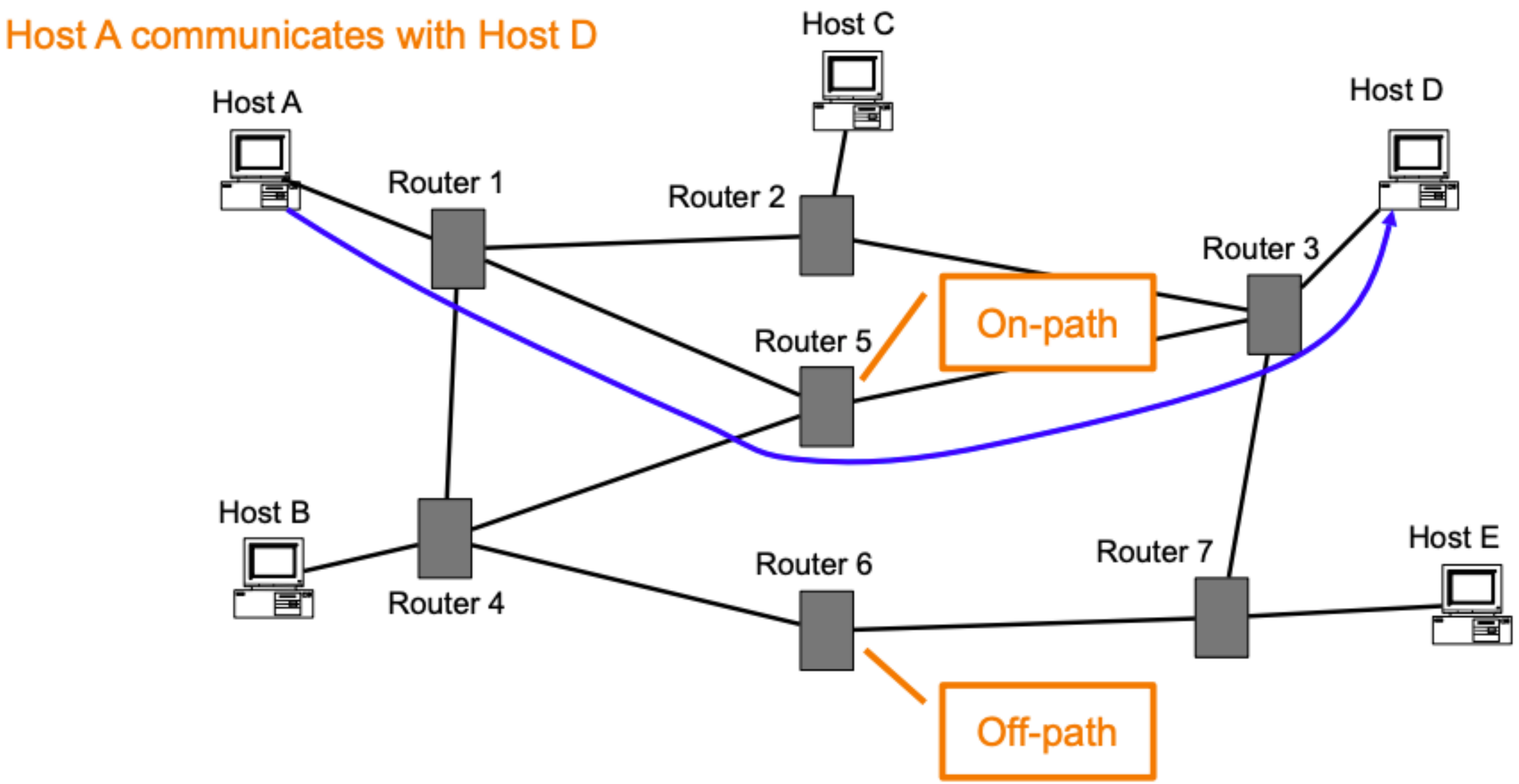
Network Attacks: Classes of Attackers

- MiTM: Can see packets, and can modify and drop packets
- On-path: Can see packets, but can't modify or drop packets
- Off-path: Can't see, modify, or drop packets



Network Attacks: Classes of Attackers

- MiTM: Can see packets, and can modify and drop packets
- On-path: Can see packets, but can't modify or drop packets
- Off-path: Can't see, modify, or drop packets



Which type of attacker is more powerful?

A. on-path
B. off-path
C. neither is strictly stronger than the other

Network Attacks: Classes of Attackers

- On-path: Can see packets, but can’t modify or drop packets
    - can see victim’s traffic: makes spoofing easy (creating a fake packet)
- Off-path: Can’t see, modify, or drop packets
    - resort to blind spoofing
    - guess/infer header values: sometimes brute-force succeeds!
        - 16 bit header field? only 2^16 possibilities
- Attacker can spoof translates to attacker has a reasonable chance of success

Protocol Layering

Networks use a stack of protocol layers

- Each layer has different responsibilities.
- Layers define abstraction boundaries

Lower layers provide services to layers above

- Don’t care what higher layers do

Higher layers use services of layers below

- Don’t worry about how the layer below works

Application Layer
Transport: end-to-end connections, reliability
Network: routing
Link (data-link): framing, error detection
Physical: 1’s and 0’s/bits across a medium (copper, the air, fiber)

Transport Layer perspective

Networks use a stack of protocol layers

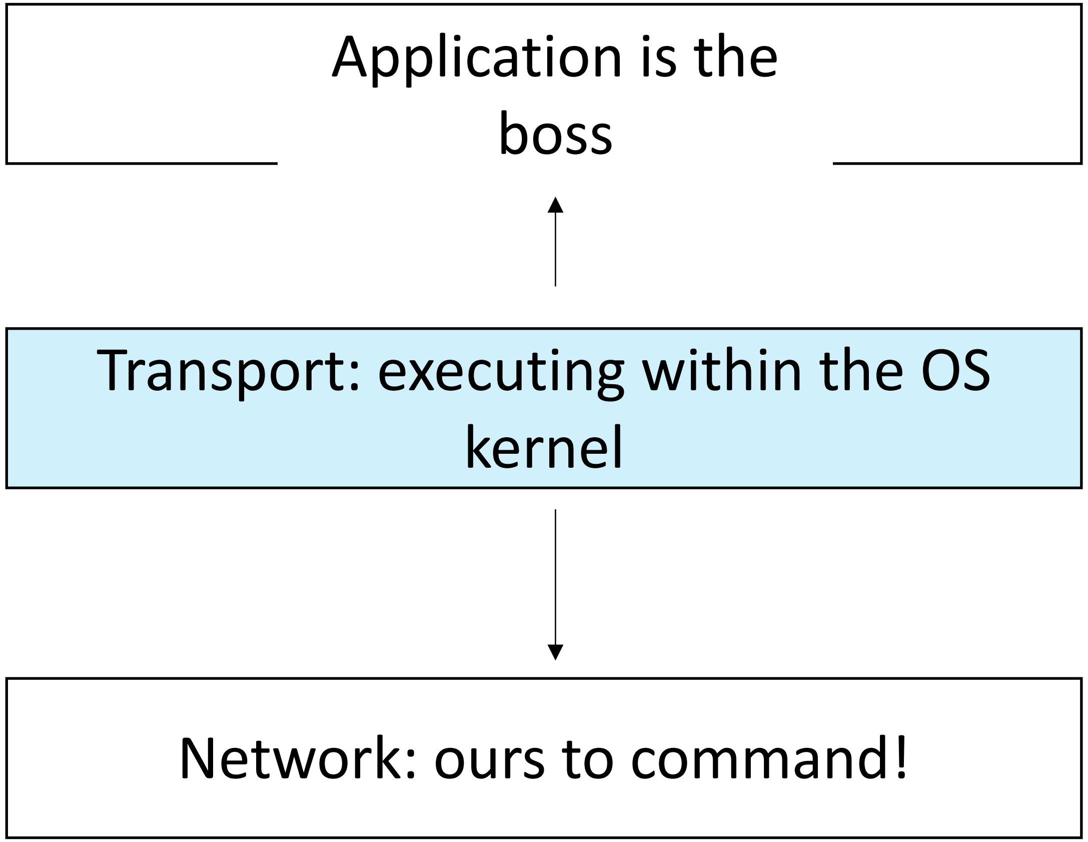
- Each layer has different responsibilities.
- Layers define abstraction boundaries

Lower layers provide services to layers above

- Don't care what higher layers do

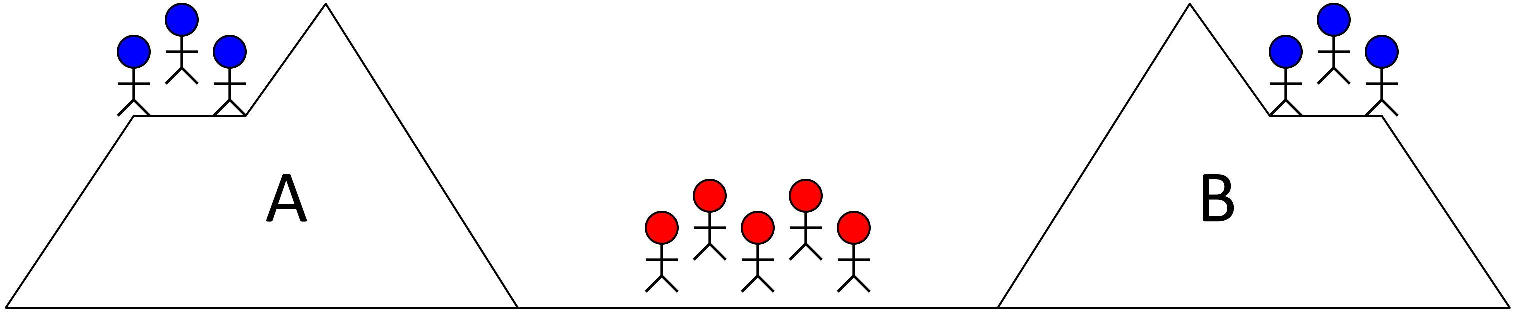
Higher layers use services of layers below

- Don't worry about how the layer below works



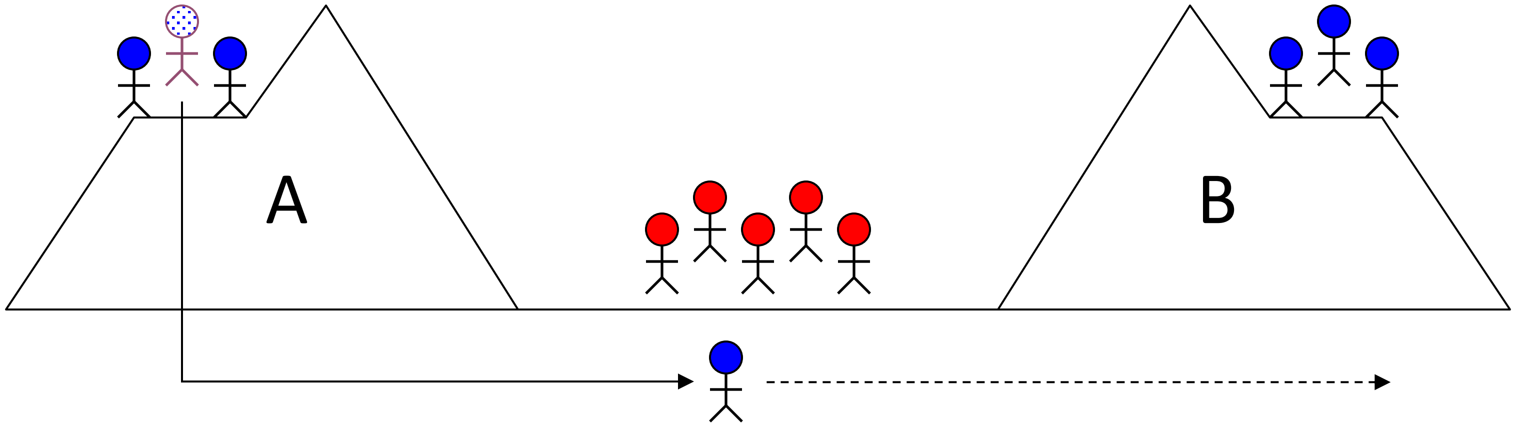
Transmission Control Protocol (TCP)

The Two Generals Problem



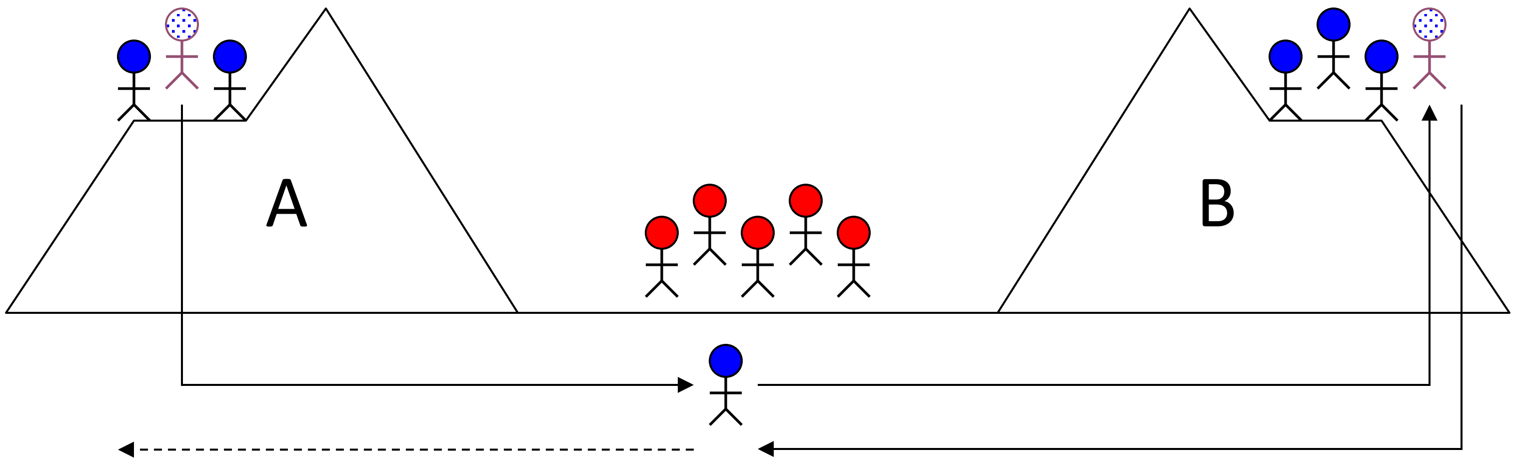
- Two army divisions (blue) surround enemy (red)
  - Each division led by a general
  - Both must agree when to simultaneously attack
  - If either side attacks alone, defeat
- Generals can only communicate via messengers
  - Messengers may get captured (unreliable channel)

The Two Generals Problem



- How to coordinate?
  - Send messenger: "Attack at dawn"
  - What if messenger doesn't make it?

The Two Generals Problem

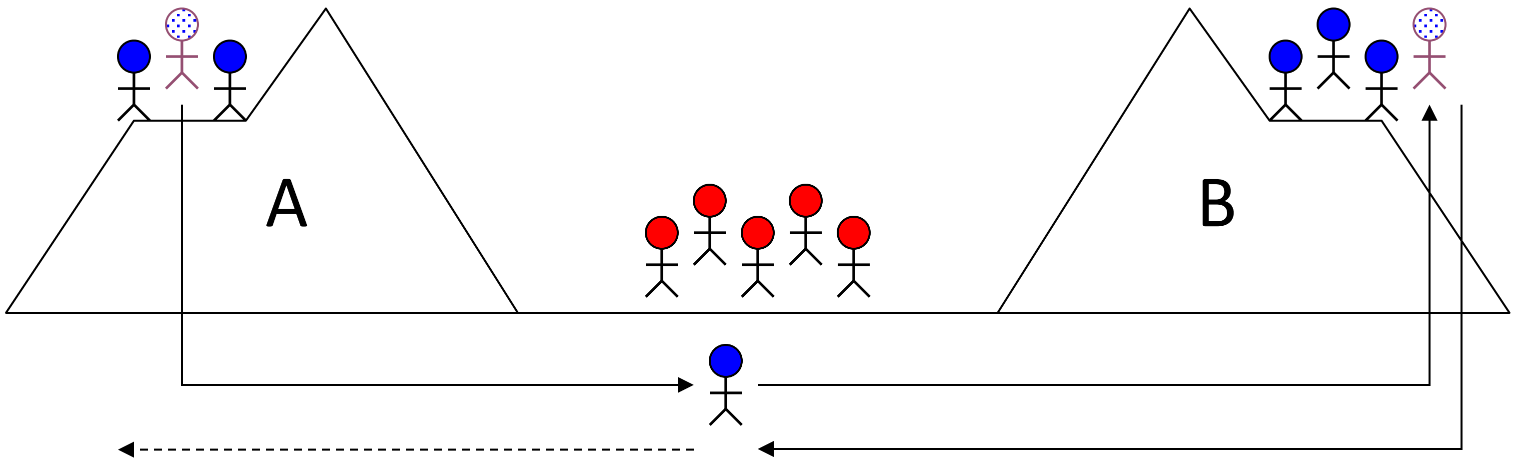


- How to be sure messenger made it?
  - Send acknowledgment: “I delivered message”

In the "two generals problem", can the two armies reliably coordinate their attack? (using what we just discussed)

- A. Yes (explain how)
- B. No (explain why not)

The Two Generals Problem



- Result
  - Can't create perfect channel out of faulty one
  - <u>Can only increase probability of success</u>

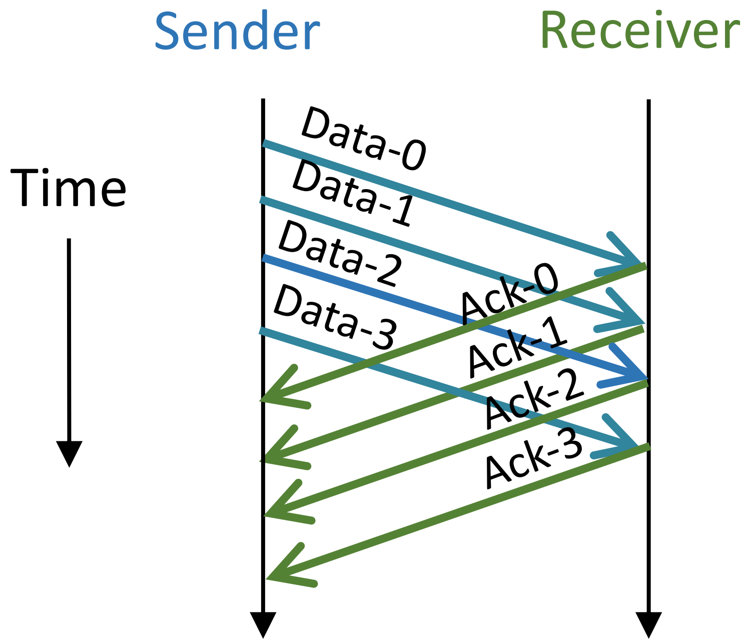
Designing reliability over an unreliable link. What can go wrong?

A. Packets can be dropped
B. Packets can arrive out or order
C. Acknowledgements can arrive out of order
D. All of the above
E. There are more issues….

Designing reliability over an unreliable link. What can go wrong?

- Problem: IP packets have a limited size. To send longer messages, we have to manually break messages into packets
    - When sending packets: TCP will automatically split up messages
    - When receiving packets: TCP will automatically reassemble the packets
    - Now the user doesn't need to manually split up messages!
- Problem: Packets can arrive out of order
    - When sending packets: TCP labels each byte of the message with increasing numbers
    - When receiving packets: TCP can use the numbers to rearrange bytes in the correct order
- Problem: Packets can be dropped
    - When receiving packets: TCP sends an extra message acknowledging that a packet has been received
    - When sending packets: If the acknowledgement doesn't arrive, re-send the packet

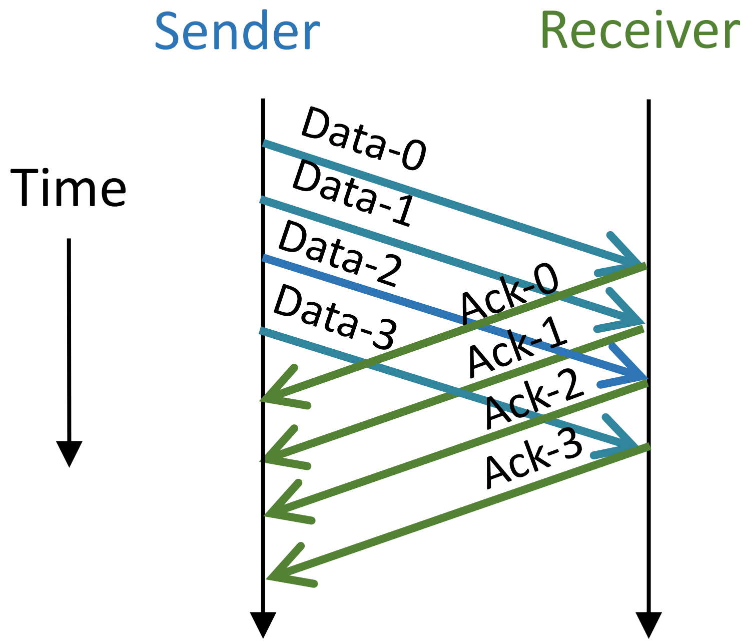
Pipelined Transmission

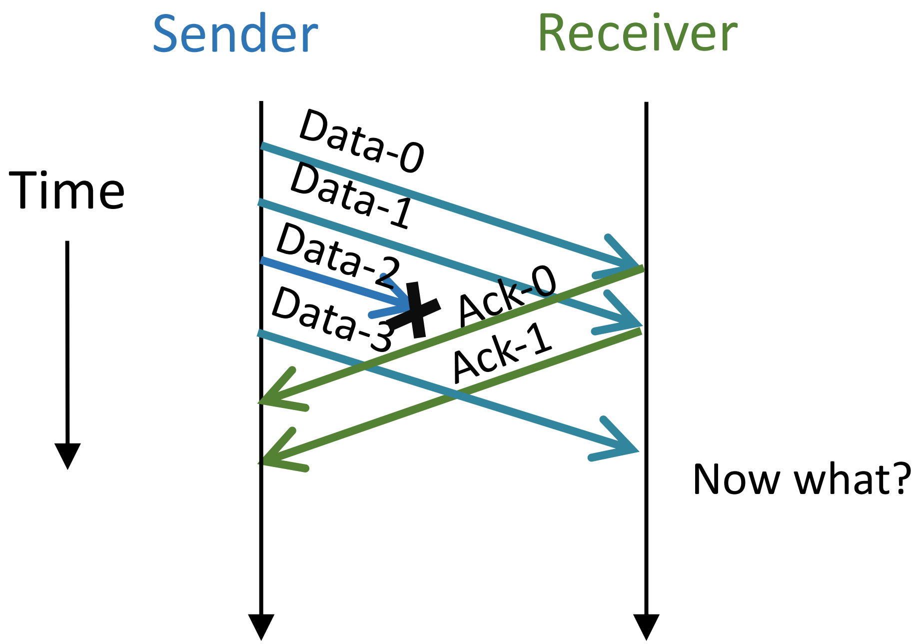


<u>Keep multiple segments "in flight"</u>

- Allows sender to make efficient use of the link
- Sequence numbers ensure receiver can distinguish segments

Pipelined Transmission

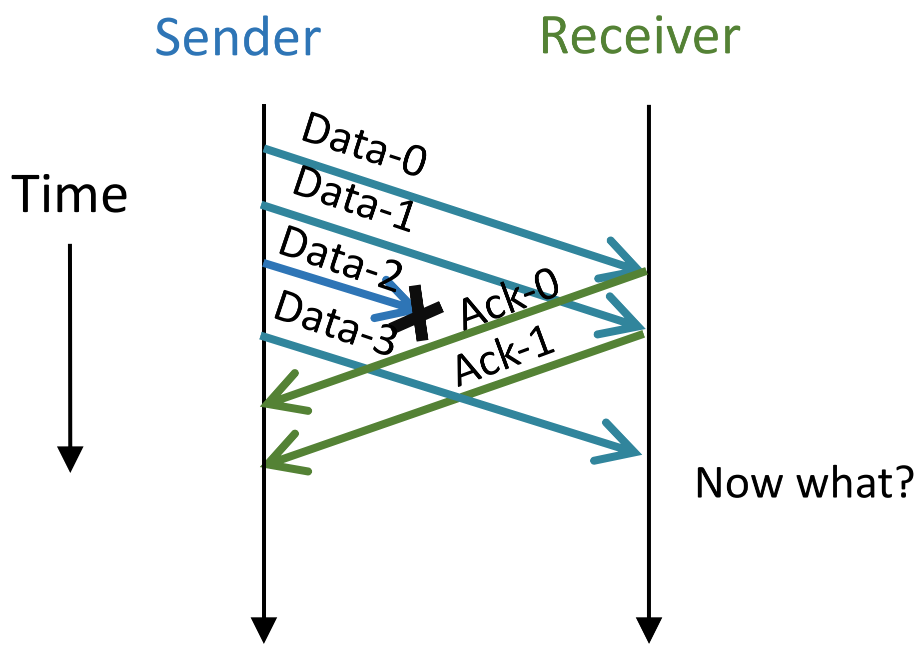




<u>Keep multiple segments "in flight"</u>

- Allows sender to make efficient use of the link
- Sequence numbers ensure receiver can distinguish segments

What should the sender do here?

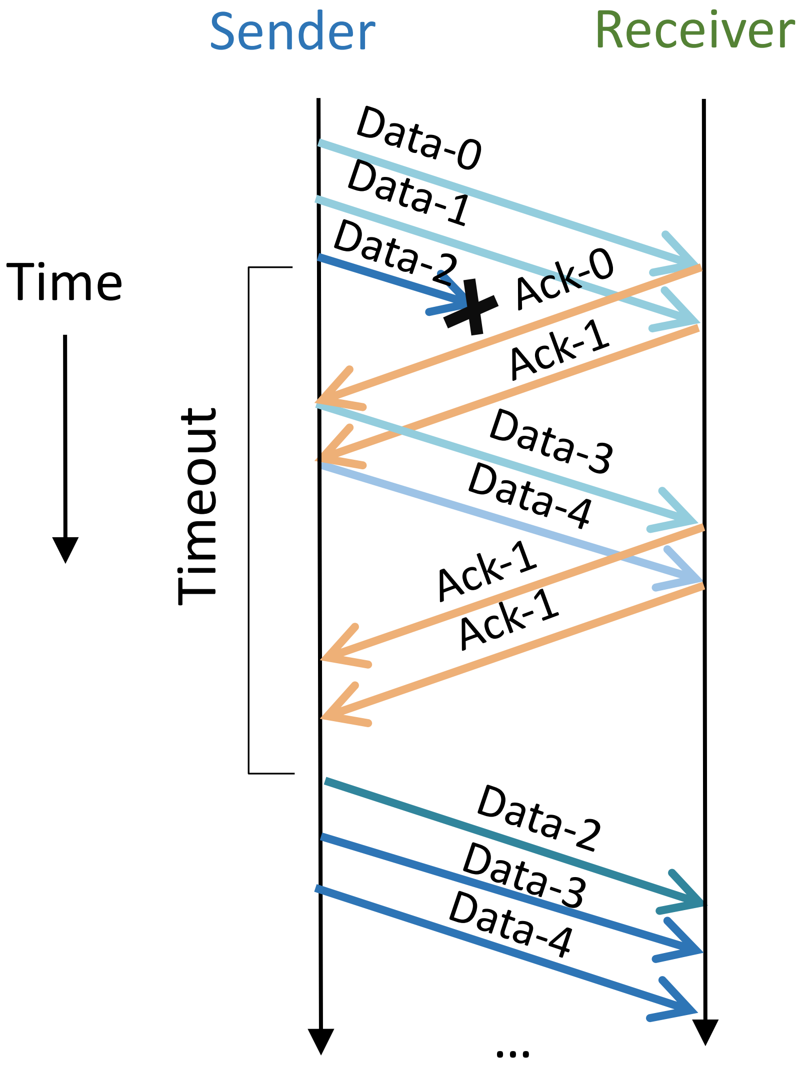


What information does the sender need to make that decision?

What is required by either party to keep track?

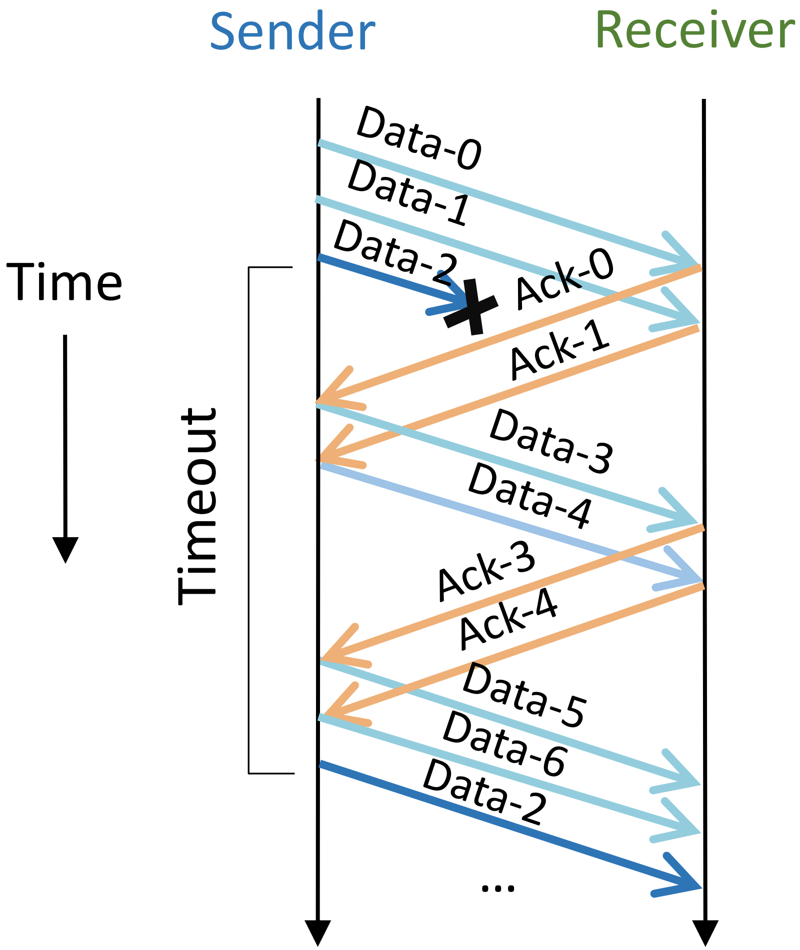
A. Start sending all data again from 0.
B. Start sending all data again from 2.
C. Resend just 2, then continue with 4 afterwards.

Go-Back-N



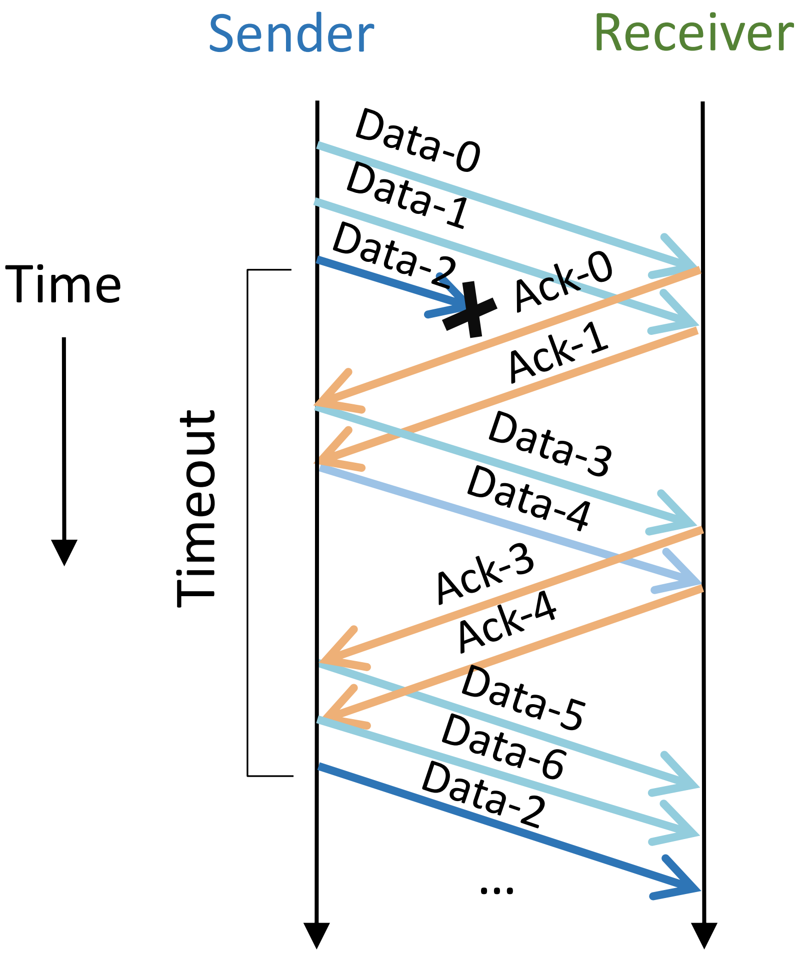
- Retransmit from point of loss
  - Segments between loss event and retransmission are ignored
  - "Go-back-N" if a timeout event occurs

Selective Repeat

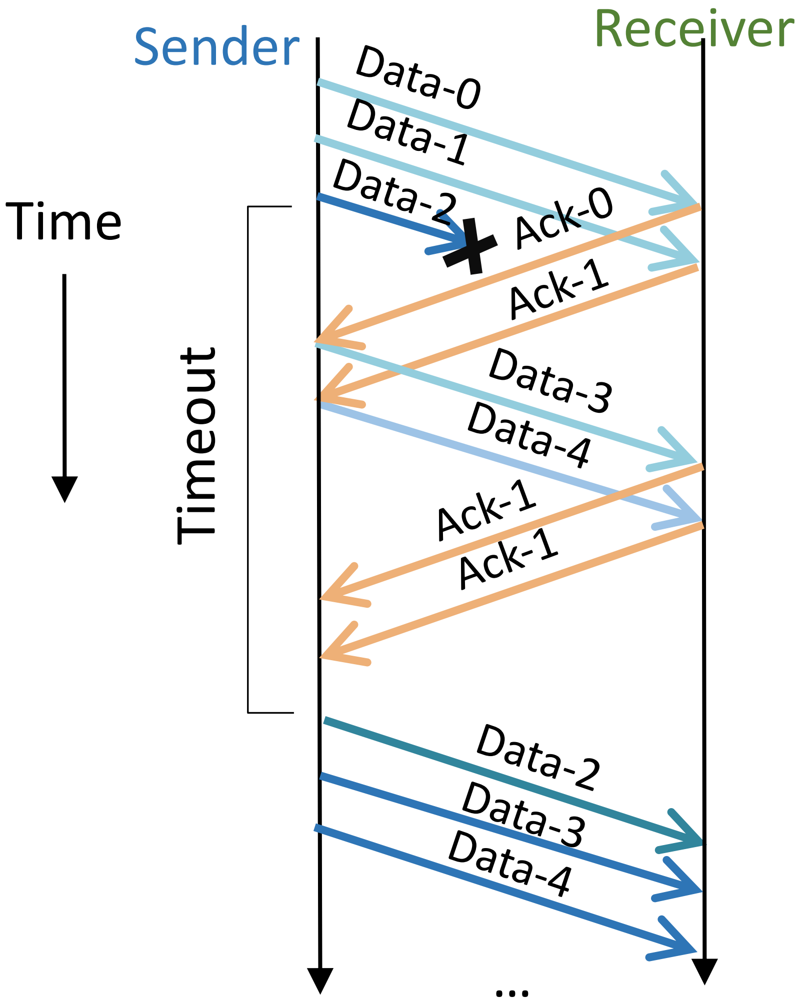


- Receiver ACKs each segment individually (not cumulative)
- Sender only resends those not ACKed

What should the sender do here?



Selective Repeat: Sender only resends those packets not ACKed



Go-Back-N: Retransmit from point of loss

What information does the sender need to make that decision?

What is required by either party to keep track?

A. Go-Back-N less work for the receiver
B. Selective Repeat less work for the network.
C. Some other combination, both are horrible.

Transmission Control Protocol (TCP)

- Provides a byte stream abstraction
  - Bytes go in one end of the stream at the source and come out at the other end at the destination
  - TCP automatically breaks streams into **segments**,
- Provides ordering
  - Segments contain sequence numbers, so the destination can reassemble the stream in order
- Provides reliability
  - The destination sends acknowledgements (ACKs) for each sequence number received
  - If the source doesn’t receive the ACK, the source sends the packet again
- Provides ports
  - Multiple services can share the same IP address by using different ports

Ports: An Analogy

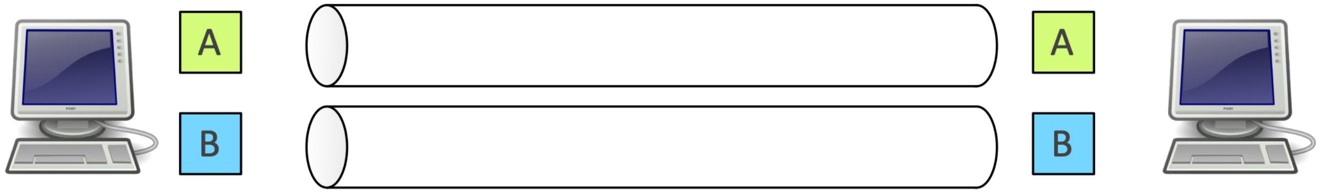
- Alice is pen pals with Bob. Alice’s roommate, Carol, is also pen pals with Bob
- Bob’s replies are addressed to the same global (IP) address
  - How can we tell which letters are for Alice and which are for Bob?
- Solution: Add a room number (port number) inside the letter
  - In private homes, usually a port number is meaningless
  - But, in public offices (servers), like Cory Hall, the port numbers are constant and known

Ports

Each application on a host is identified by a *port number*

TCP connection established between port *A* on host *X* to port *B* on host *Y* Ports are 1–65535 (16 bits)

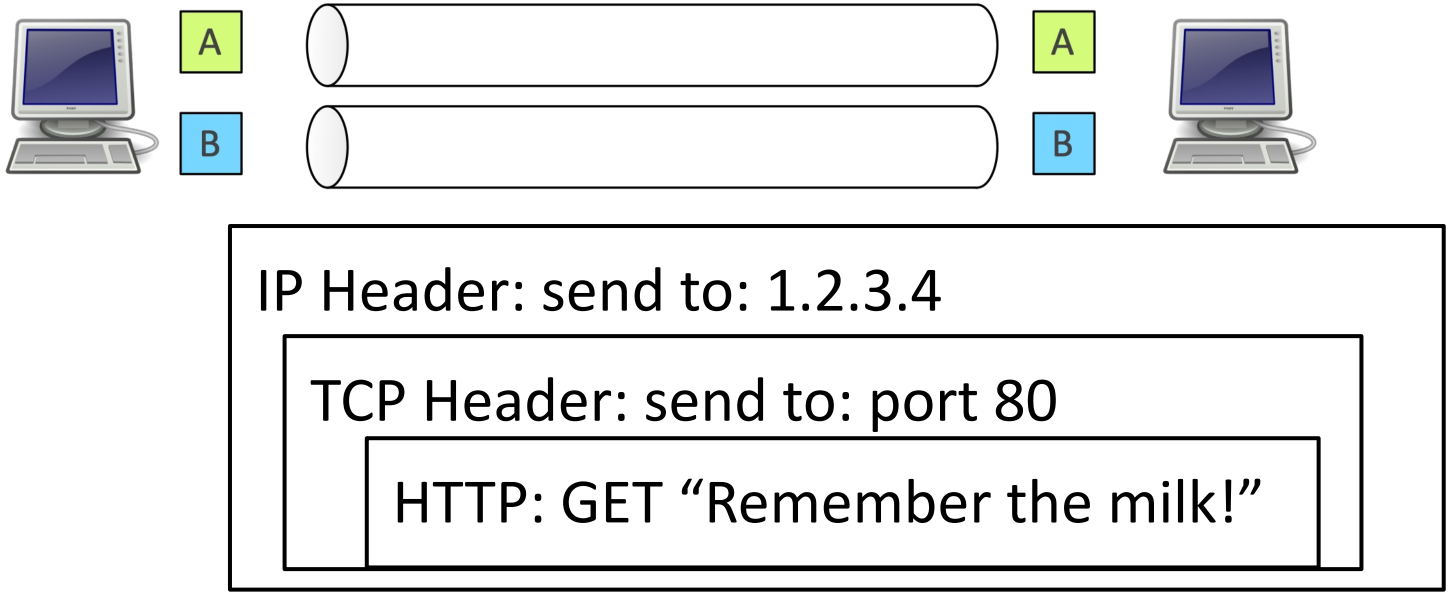
Some destination port numbers used for specific applications by convention



Ports

Ports help us distinguish between different applications on the same computer or server

- On private computers, port numbers can be random
- On public servers, port numbers should be constant and well-known (so users can access the right port)



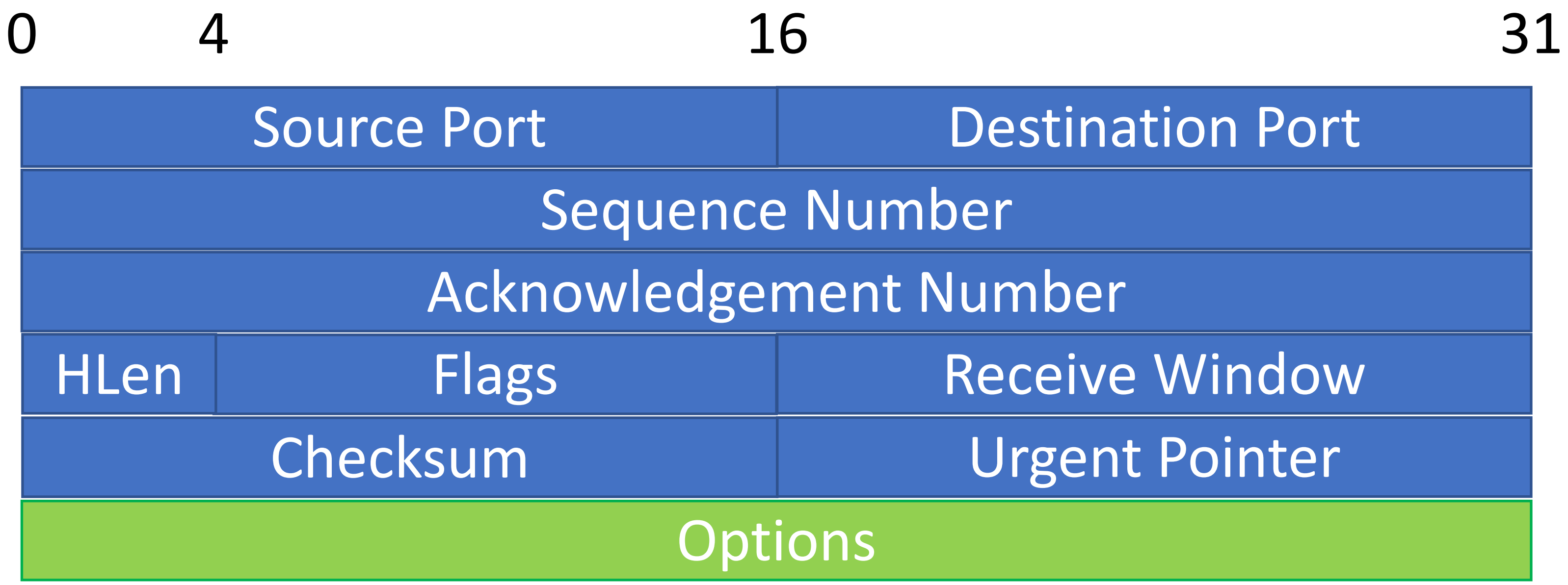
Common Ports

Port	Application
80	HTTP (Web)
443	HTTPS (E2E encrypted Web)
25	SMTP
22	SSH
23	Telnet
53	DNS

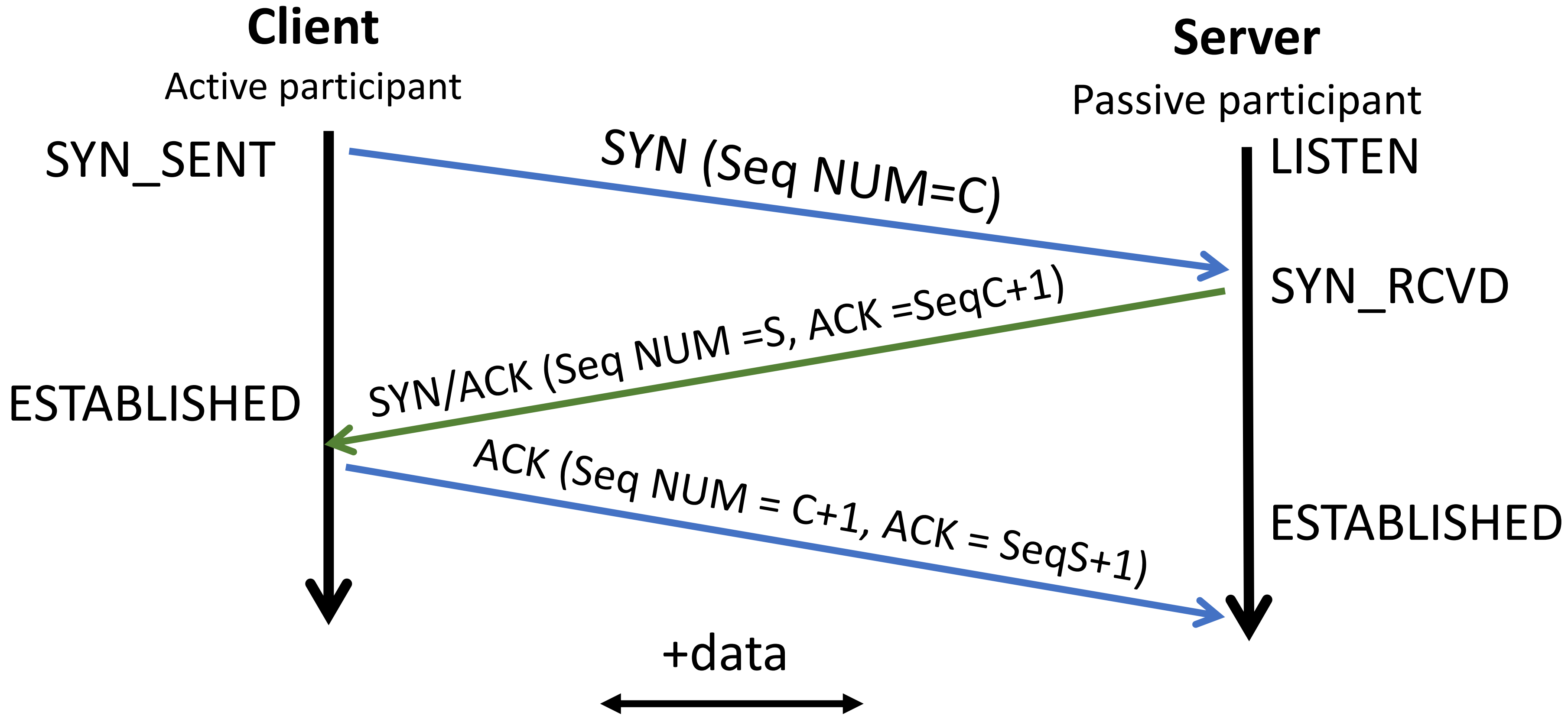
Transmission Control Protocol

Reliable, in-order, bi-directional byte streams

- Port numbers for demultiplexing
- Flow control
- Congestion control, approximate fairness

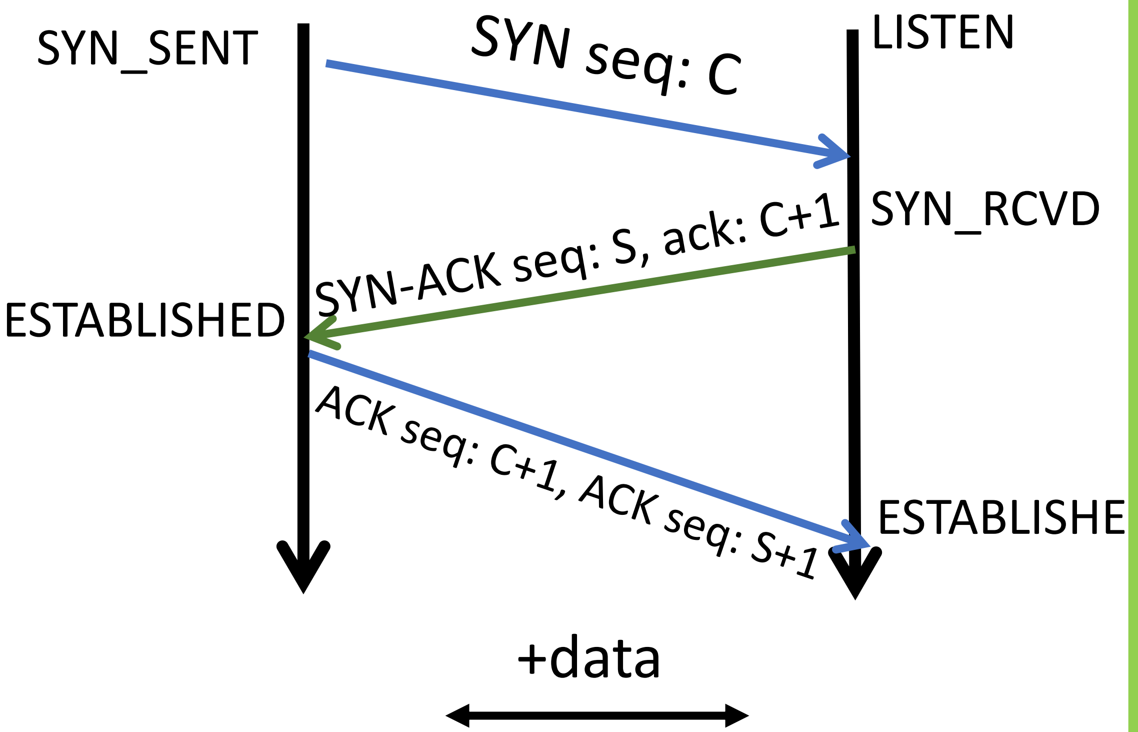


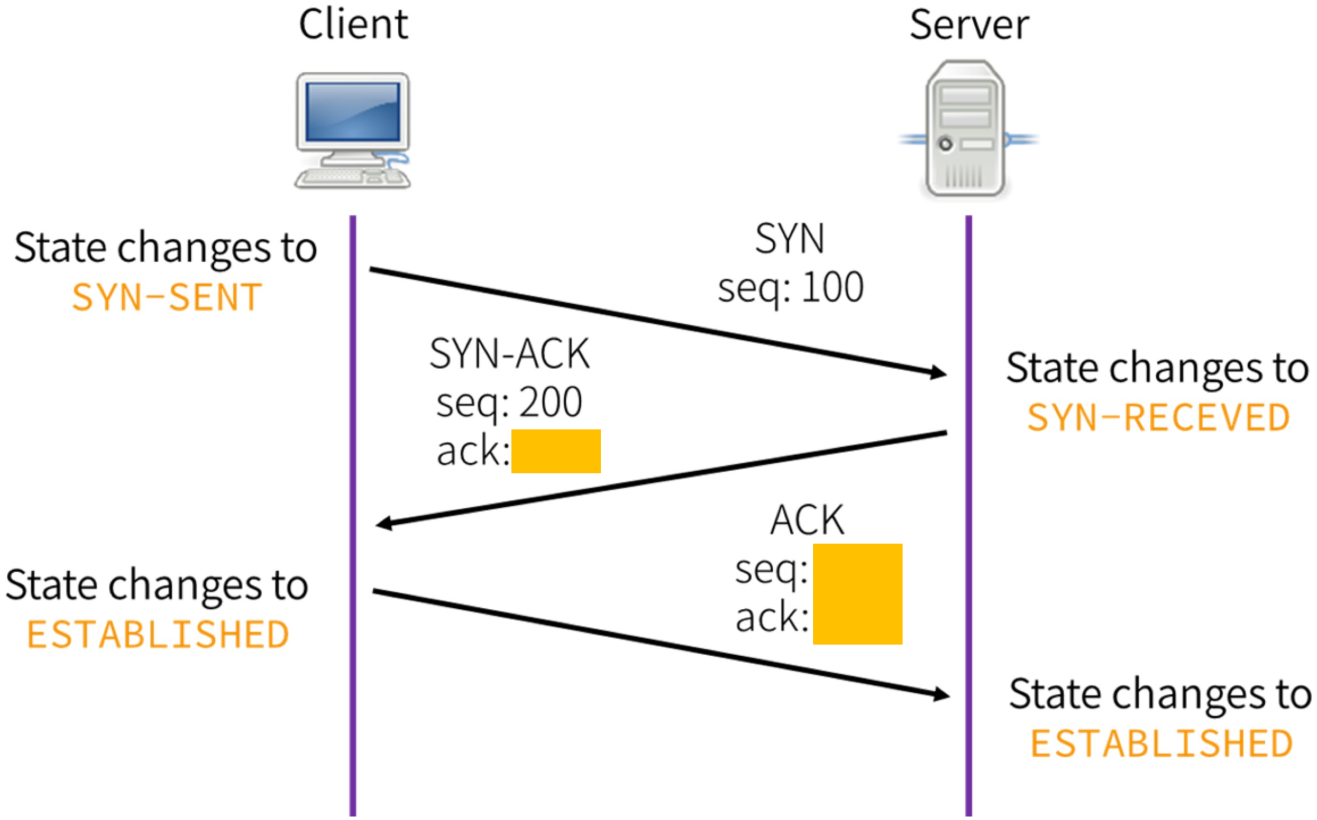
Three Way Handshake



- Each side:
  - Notifies the other of starting sequence number
  - ACKs the other side's starting sequence number

TCP Three Way Handshake





A. SYN-ACK: ack:200, ACK: seq: 300, ack: 400

B. SYN-ACK: ack:201, ACK: seq: 301, ack: 401

C. SYN-ACK: ack:101, ACK: seq: 101, ack: 201

D. SYN-ACK: ack:101, ACK: seq: 201, ack: 101

How should we choose the initial sequence number?

A. Start from zero

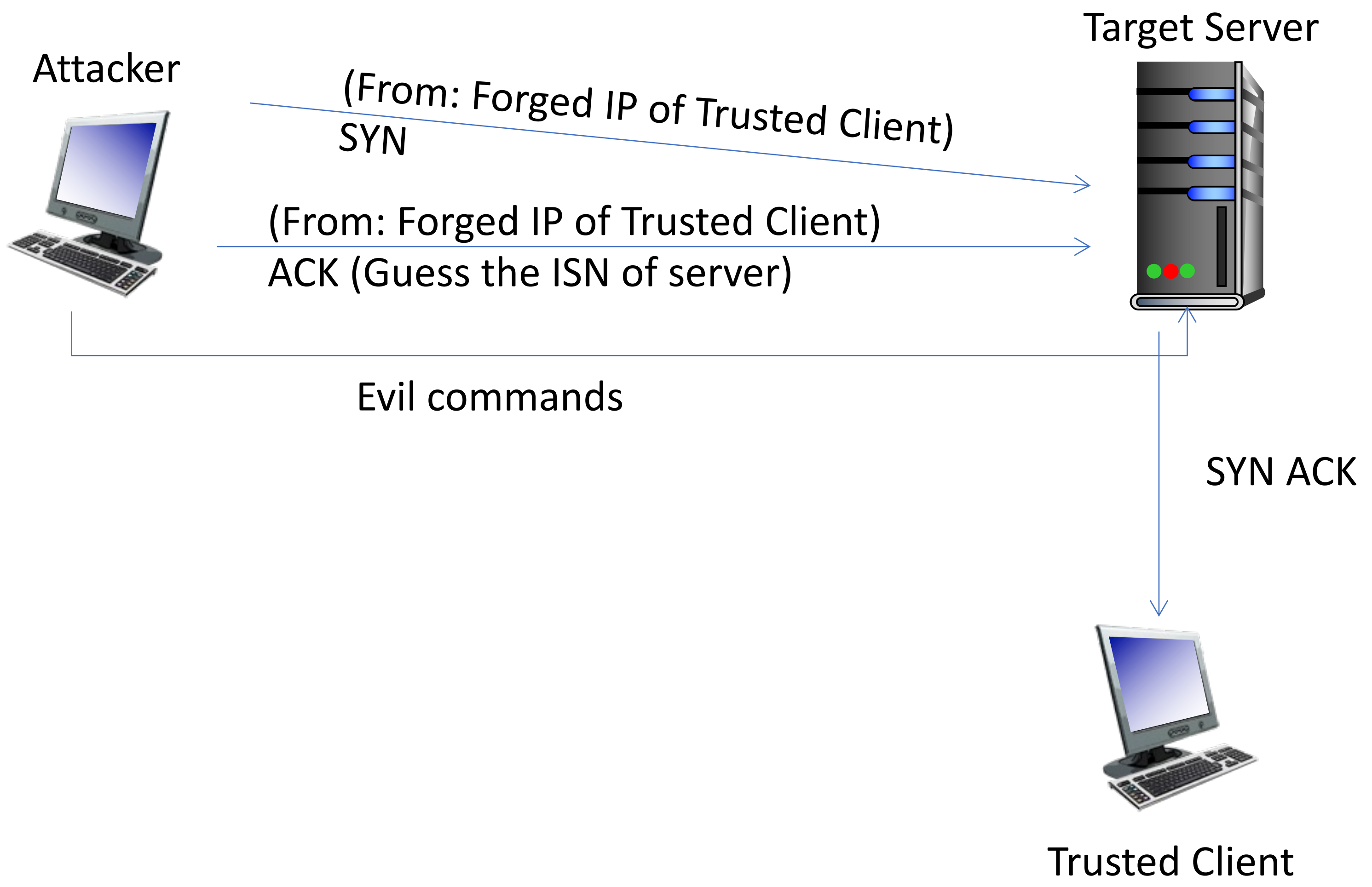
B. Start from one

C. Start from a random number

D. Start from some other value (such as…?)

What can go wrong with sequence numbers?
-How they're chosen?
-In the course of using them?

TCP Connection Spoofing: Sequence Prediction Attack

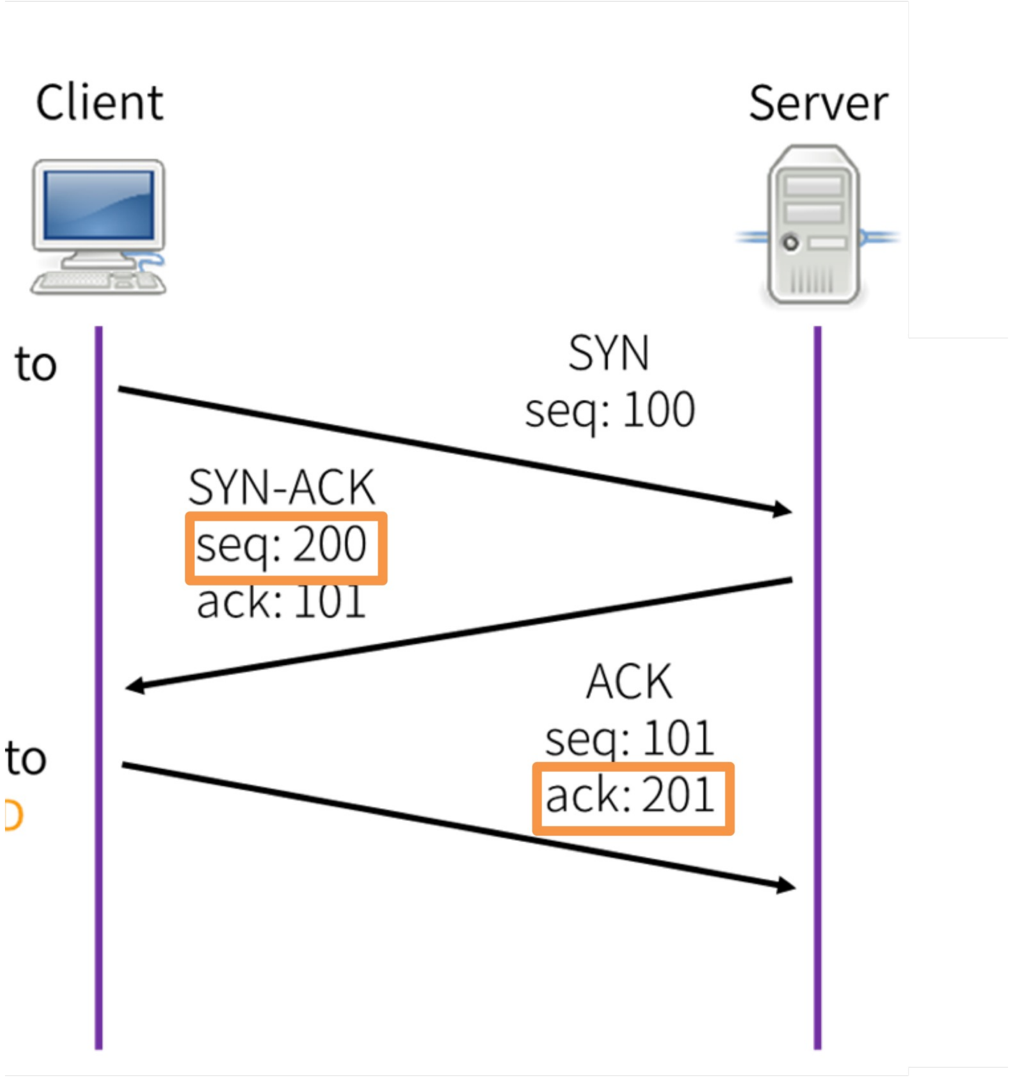


TCP Connection Spoofing

Can we impersonate another host when *initiating* a connection?

Off-path attacker can send initial SYN to server …
… but cannot complete three-way handshake without seeing the server's sequence number

1 in 2^{32} chance to guess right if initial sequence number chosen uniformly at random



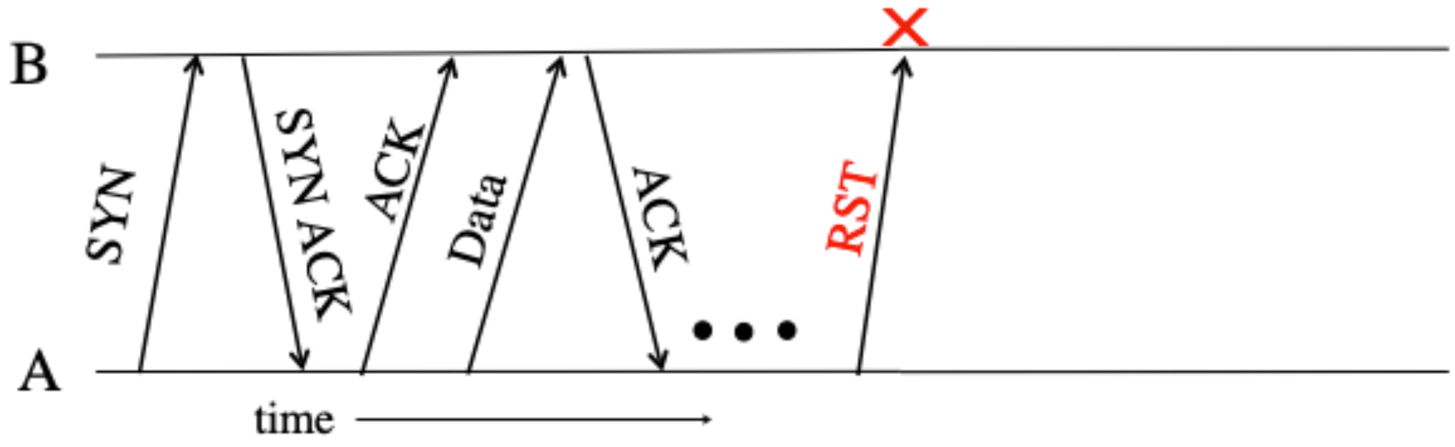
TCP Flags: Ending/Aborting a Connection

- ACK
  - Indicator that the user is acknowledging the receipt of something (in the ack number)
  - Pretty much always set except the very first packet
- SYN
  - Indicator of the beginning of the connection
- FIN
  - One way to end the connection
  - Requires an acknowledgement
  - No longer sending packets, but will continue to receive
- RST
  - One way to end a connection
  - Does not require an acknowledgement
  - No longer sending or receiving packets

TCP: Ending/Aborting a Connection

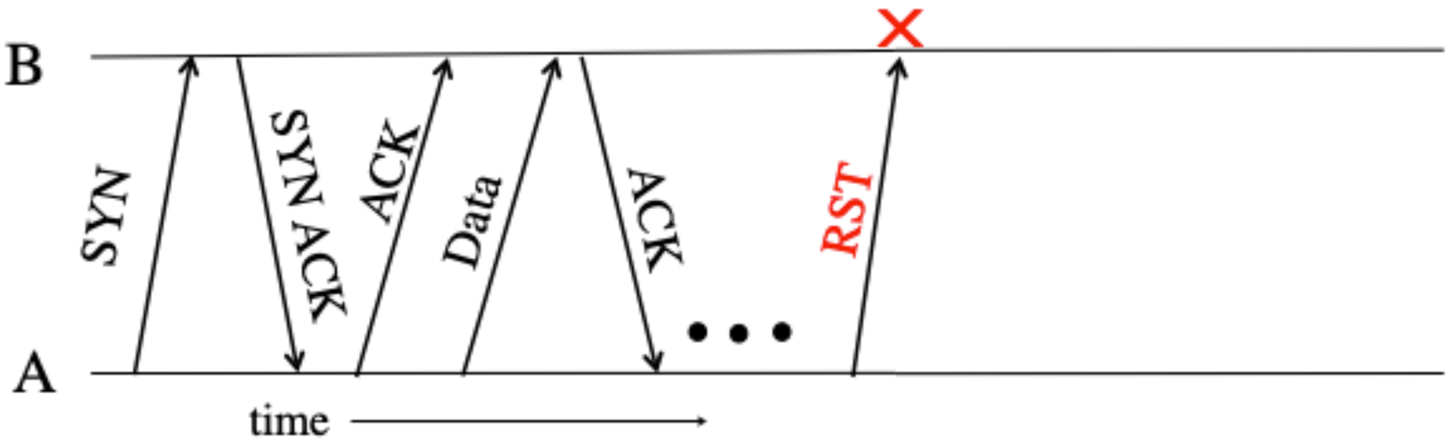
- To **end** a connection, one side sends a packet with the FIN (finish) flag set, which should then be acknowledged
  - This means “I will no longer be sending any more packets, but I will continue to receive packets”
  - Once the other side is no longer sending packets, it sends a packet with the FIN flag set
- To **abort** a connection, one side sends a packet with the RST (reset) flag set
  - This means “I will no longer be sending nor receiving packets on this connection”
  - RST packets are not acknowledged since they usually mean that something went wrong

TCP RST Injection



- If A sends a TCP packet with RST flag to B and sequence number fits, connection is terminated
  - Unilateral, and takes effect immediately

TCP RST Injection Attack



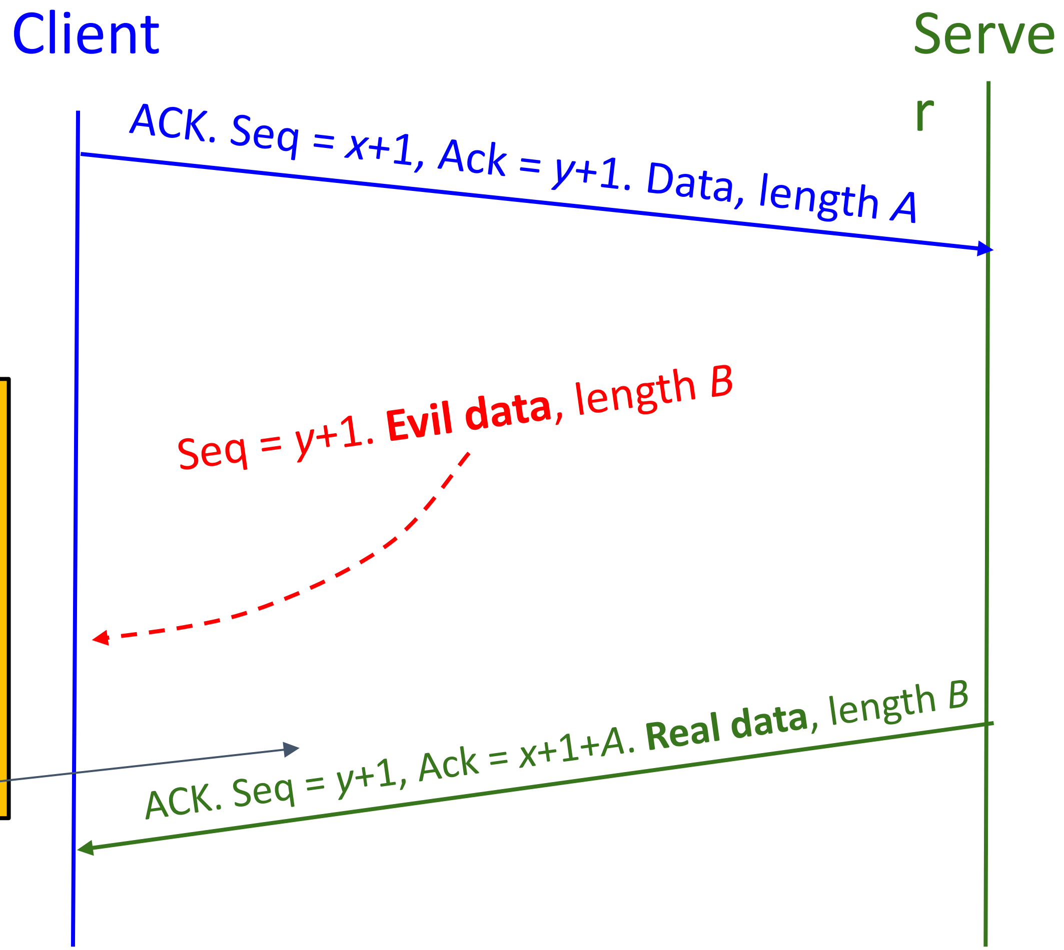
Who can do RST injection?
A. off-path attacker
B. on-path attacker
C. man-in-the-middle

The attacker can inject RST packets and block connection
TCP clients must respect RST packets and stop all communication

Who uses this? Historically..

- China: The Great Firewall does this to TCP requests
- A long time ago: Comcast, to block BitTorrent uploads
- Some intrusion detection systems: To hopefully mitigate an attack in progress

TCP Data Injection: Tampering with an existing session to modify or inject data into a connection

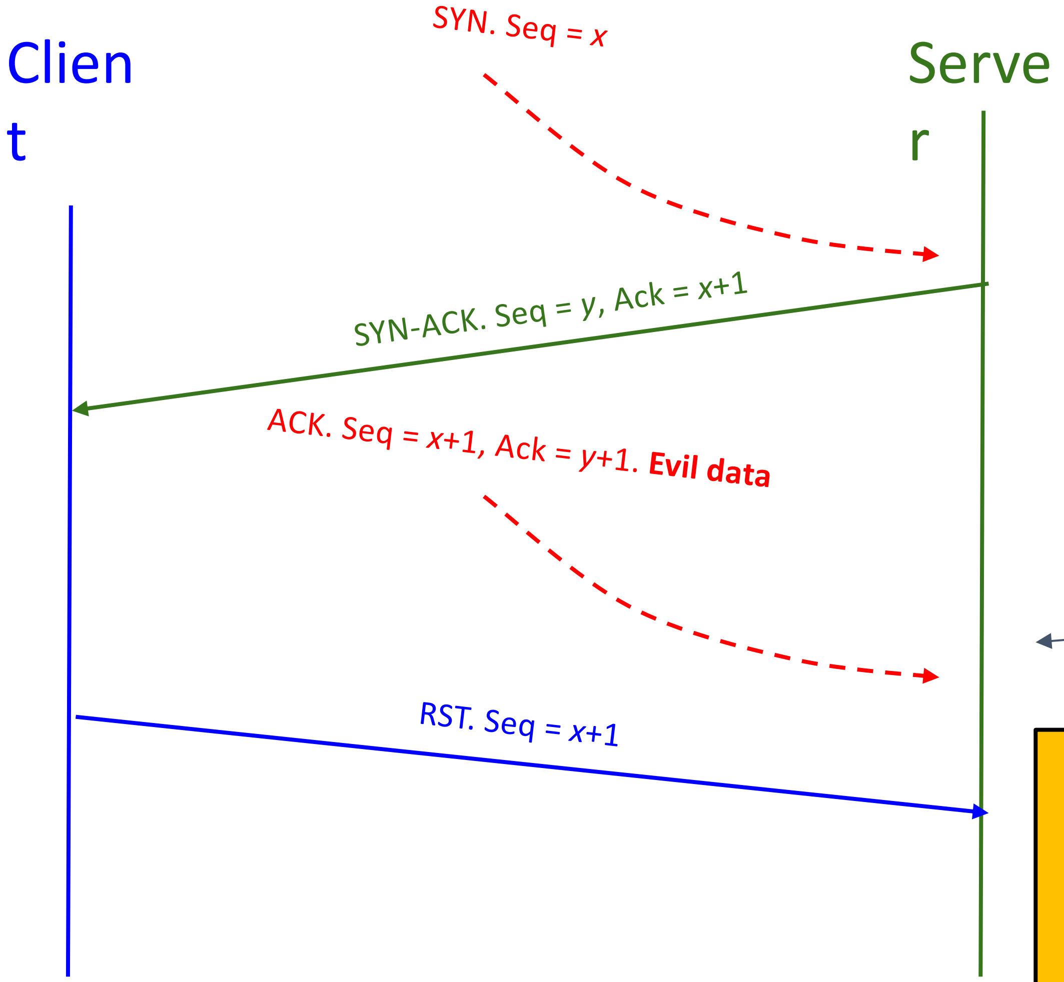


This packet will be ignored by the client since the client already processed the malicious packet!

TCP Attacks

- **TCP hijacking**: Tampering with an existing session to modify or inject data into a connection
  - **Data injection**: Spoofing packets to inject malicious data into a connection
    - Need to know: The sender’s sequence number
  - Easy for MITM and on-path attackers, but off-path attackers must guess 32-bit sequence number (called **blind injection/hijacking**, considered difficult)
  - For on-path attackers, this becomes a race condition since they must beat the server’s legitimate response

TCP Spoofing



An on-path attacker must send the evil data before the server receives the

A MITM attack could just drop the client's packets, however

TCP Provides..

A. Confidentiality
B. Availability
C. Integrity
D. None of the above

TCP Provides..

- TCP provides no confidentiality or integrity
  - Instead, we rely on higher layers (like TLS, more on this next time) to prevent those kind of attacks
- Defense against off-path attackers rely on choosing random sequence numbers
  - Bad randomness can lead to trivial off-path attacks: TCP sequence numbers used to be based on the system clock!