CS 181AG
Lecture 17

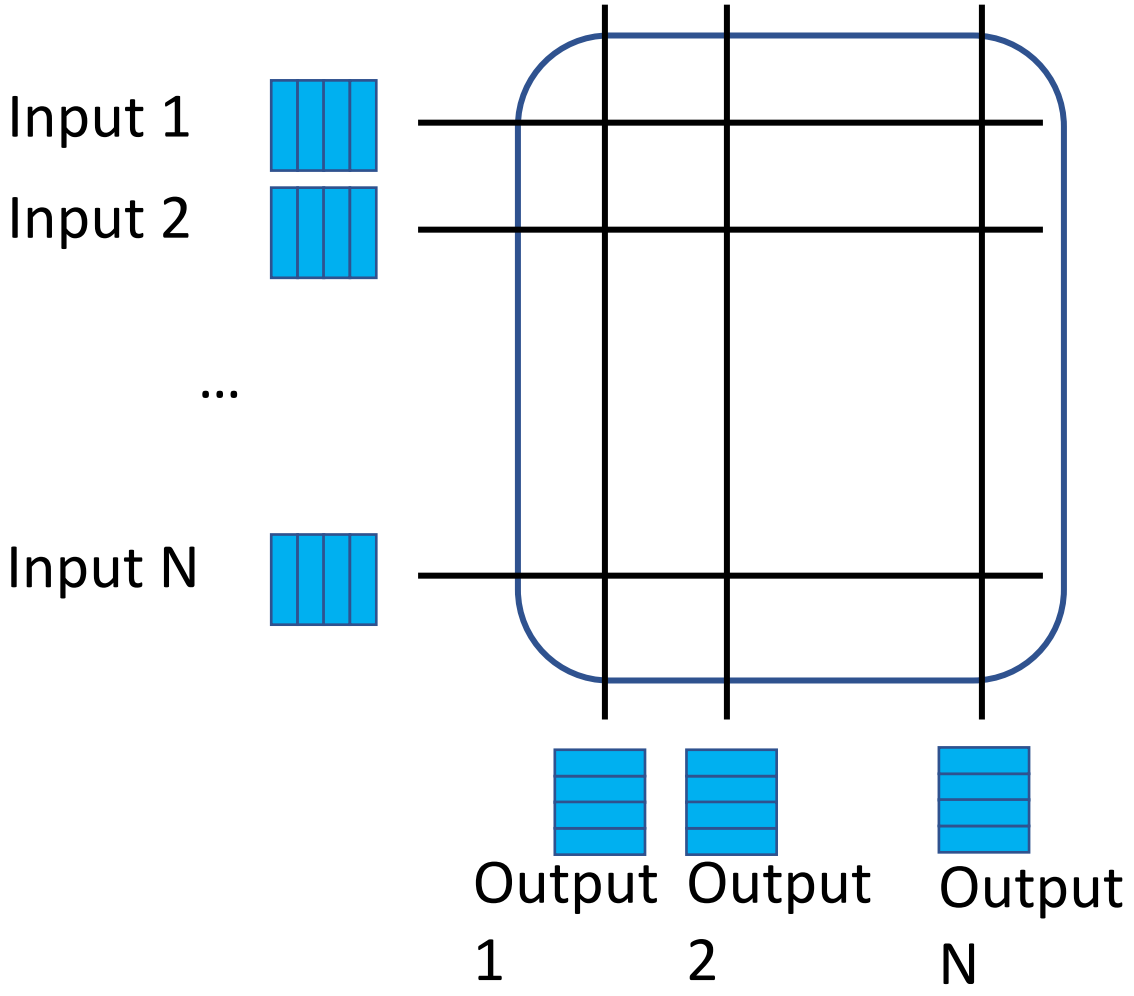
Intro to Transport Layer

Arthi Padmanabhan

Nov 2, 2022

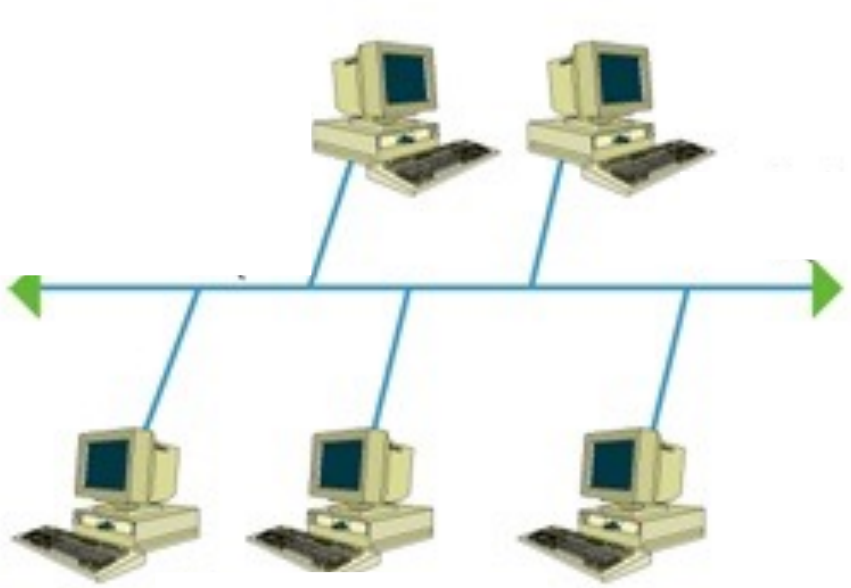
Recap from Monday

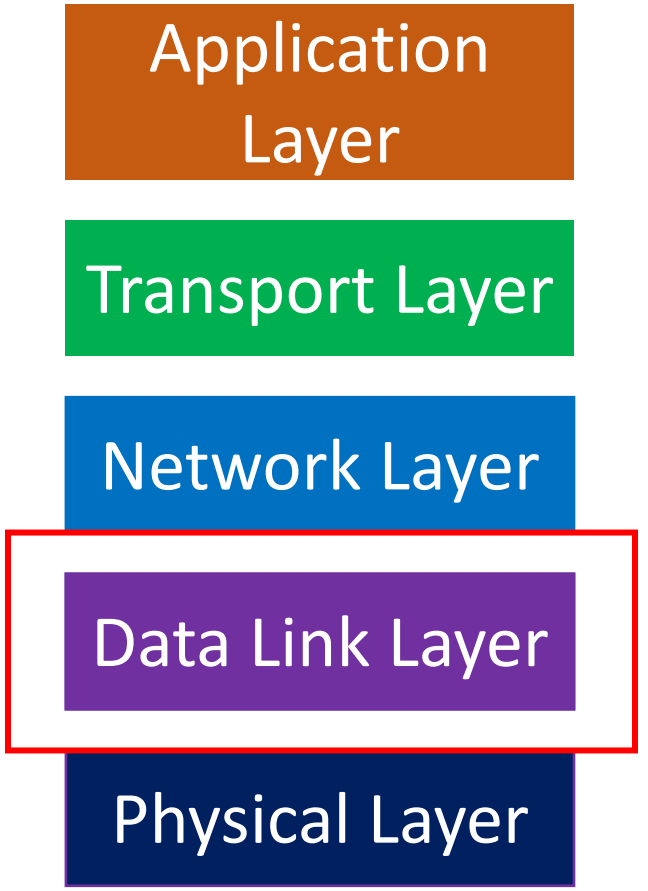
- Routers use a crossbar switch to move packets between input and output ports
- Routers keep queues for received packets AND queues for outgoing packets
- Virtual Output Queues (VOQs) help by avoiding HOL blocking
- PIM: inputs request to each output for which it has packets, outport port chooses which input to grant randomly, input chooses which output to accept randomly
- iSLIP: similar to PIM but choose with rotating pointer instead of randomly



Revisiting Layered Architecture

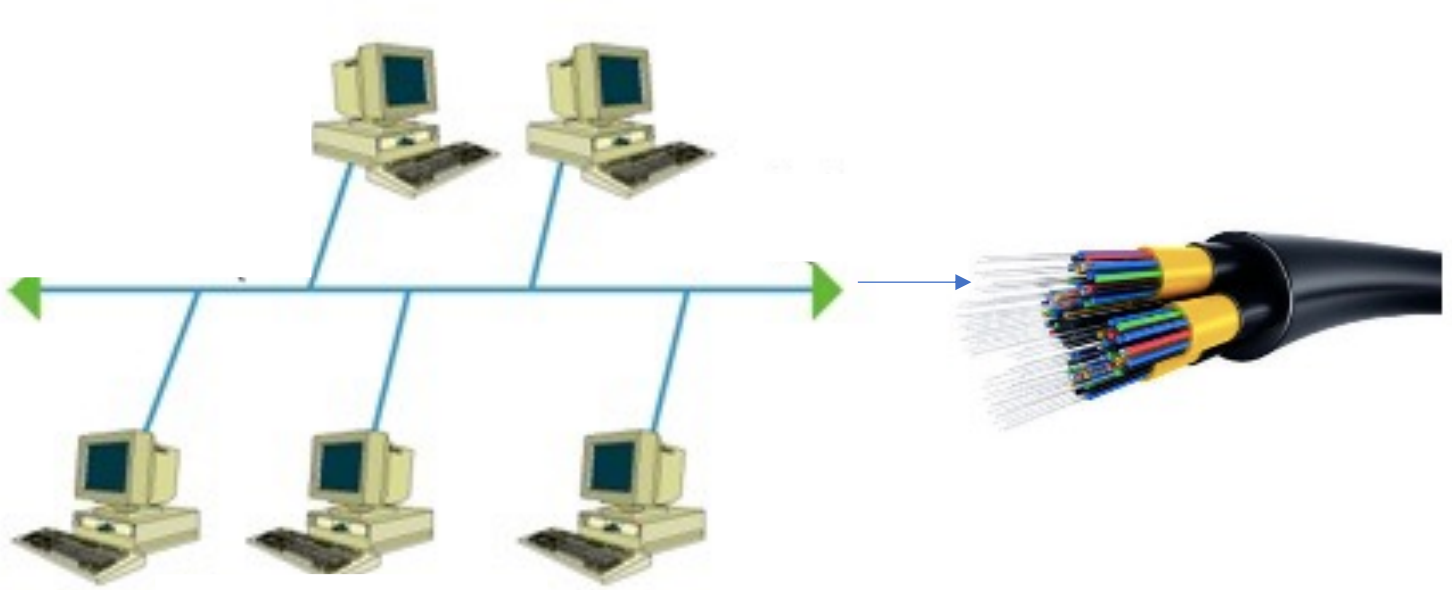
- **Single hop**: Start with one wire: how do multiple devices share a single wire to communicate with each other?





Revisiting Layered Architecture

- **Physical Medium**: What is the wire itself made of? What about wireless mediums?



Application Layer

Transport Layer

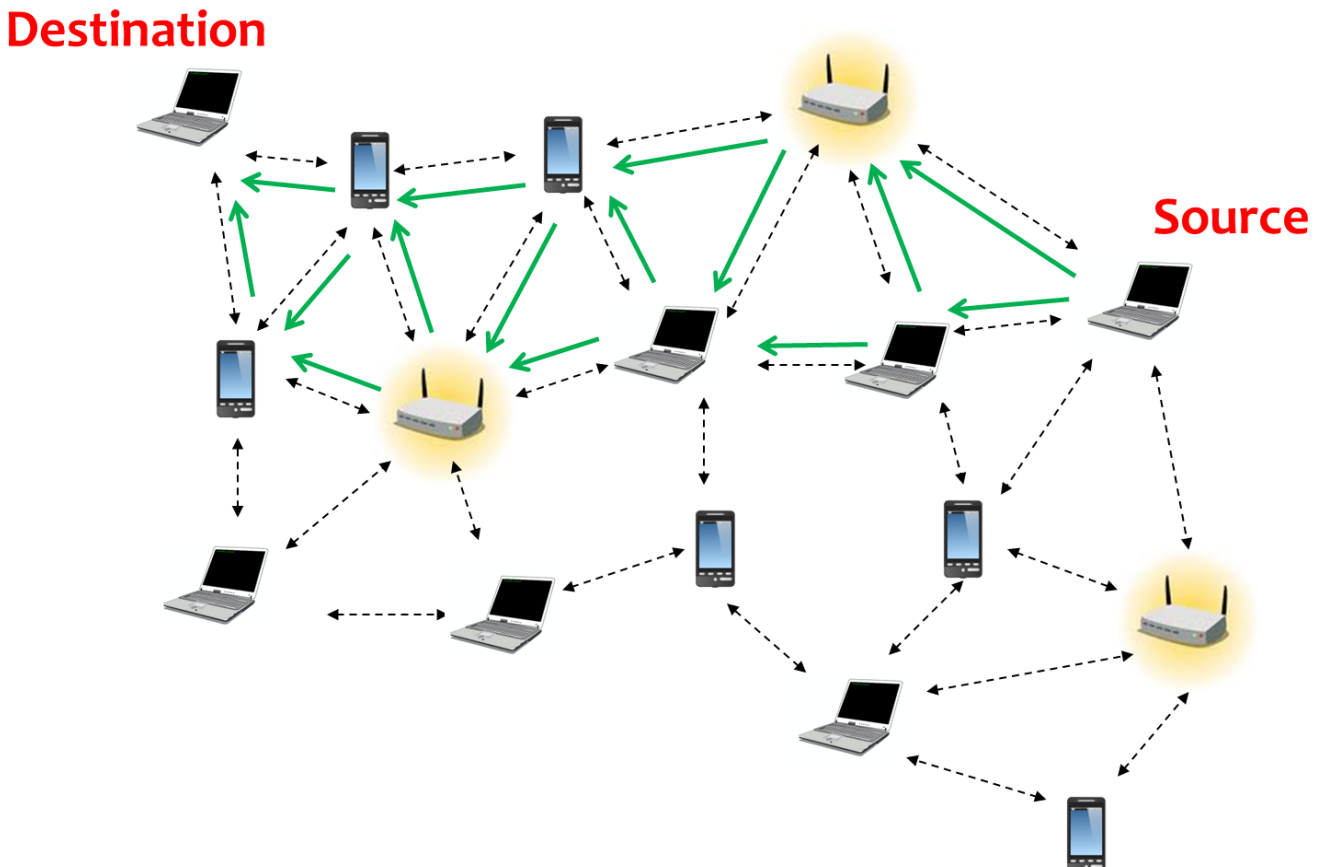
Network Layer

Data Link Layer

Physical Layer

Revisiting Layered Architecture

- **Routing**: What path should data take?



Application Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

Revisiting Layered Architecture

- **End-to-end delivery**: How do we know the packet reached?

Application Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

Revisiting Layered Architecture

- **User interaction**: How does an internet user interface with network services?

Application Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

Transport Layer

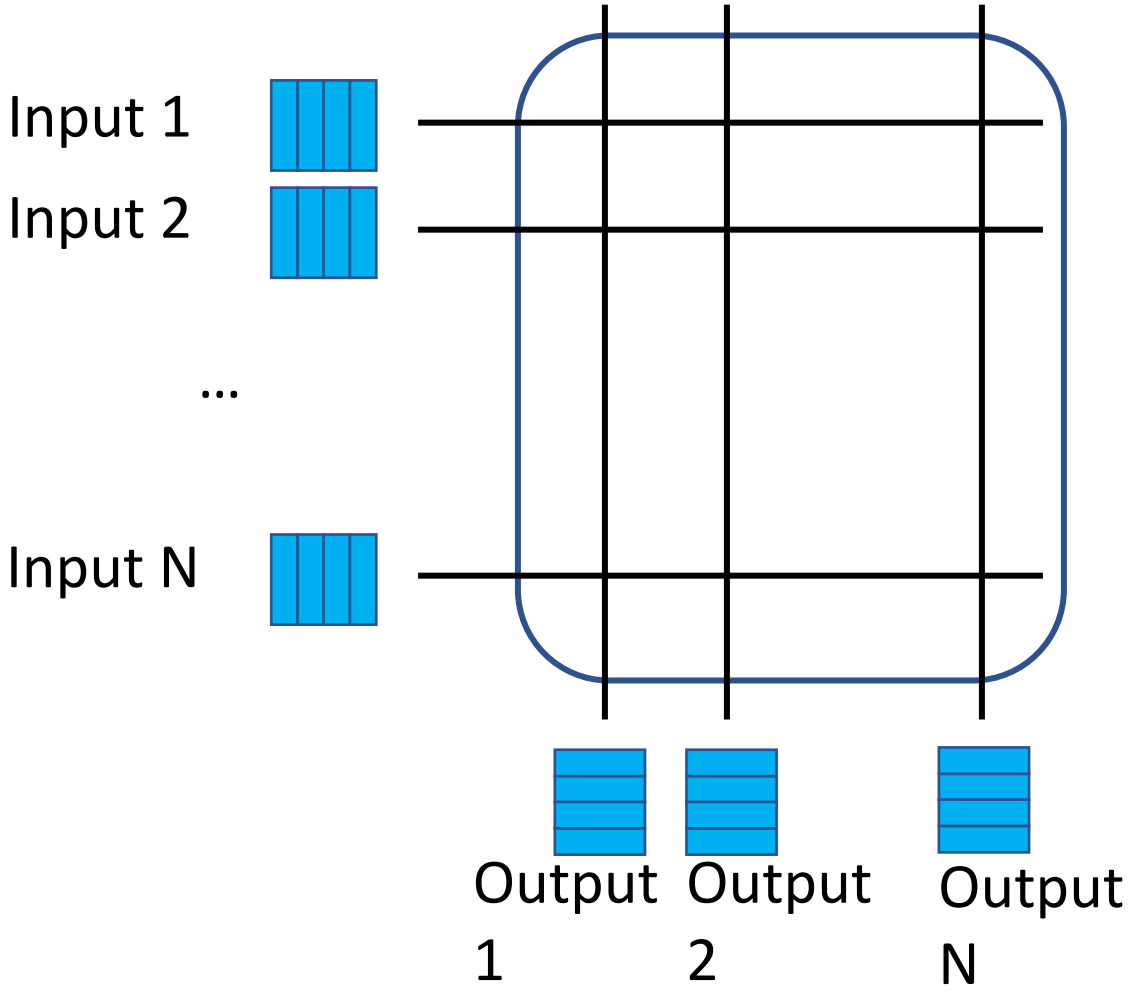
- Protocols:
  - Transmission Control Protocol (**TCP**): provides reliability in a multi-hop network
  - User Datagram Protocol (**UDP**): faster but less reliable; better for low-latency applications like video
- Today's focus: TCP

TCP

- Provides reliability across multi-hop network
- Each link in a network is quite reliable, so why do we worry about packets getting lost?

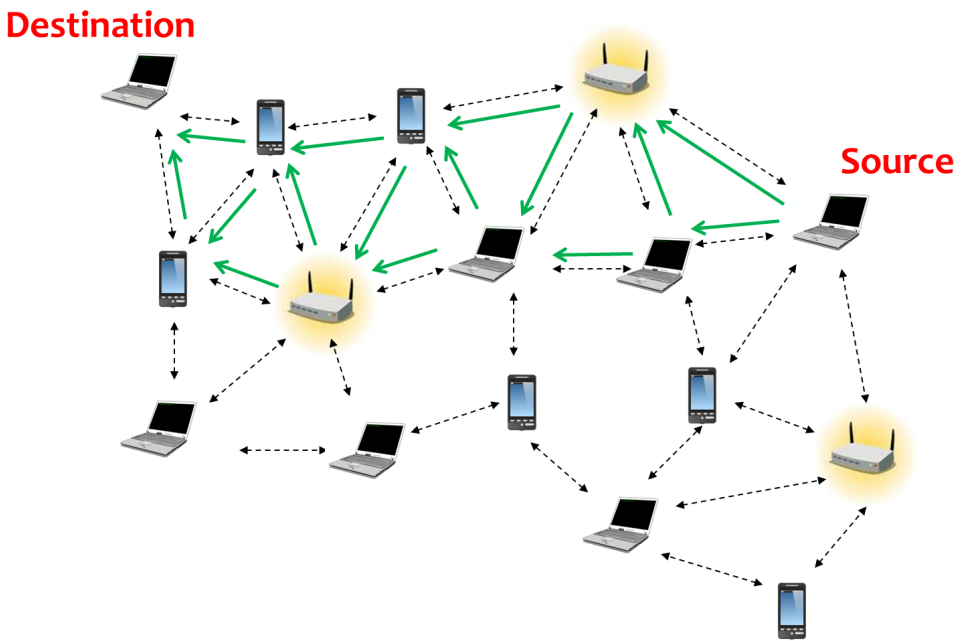
Congestion

- Congestion leads to packet loss!
- We added queues. We sped up algorithms to do lookup, classification, and moving packets. Sometimes, it's not enough
- When packets arrive faster than router can process them (queues fill up), packets are dropped



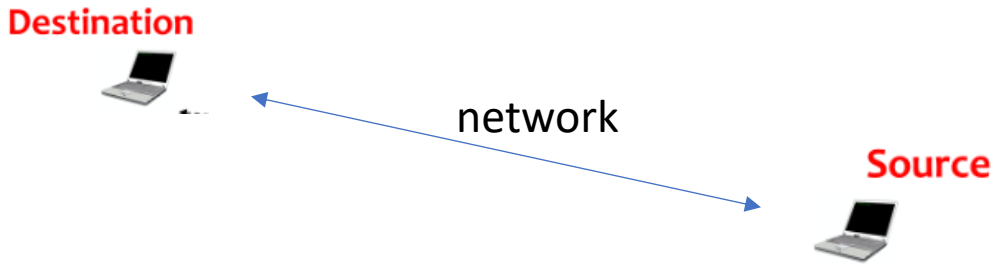
TCP

- Provides reliability across multi-hop network
- Each link in a network is quite reliable, but routers drop packets when they are too congested to process them all
- TCP treats mesh network as a black box



TCP

- Provides reliability across multi-hop network
- Each link in a network is quite reliable, but routers drop packets when they are too congested to process them all
- TCP treats mesh network as a black box



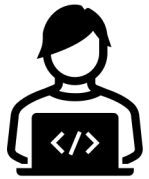
TCP

Sender

Data

Network

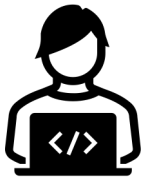
Receiver



TCP

Sender

Receiver

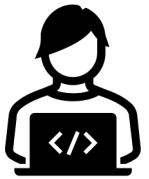


TCP

Sender

Network

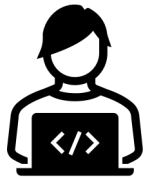
Receiver



TCP

Sender

Receiver



Network

- Two potential problems:
  - Packet loss
  - Out-of-order arrival

TCP Goals

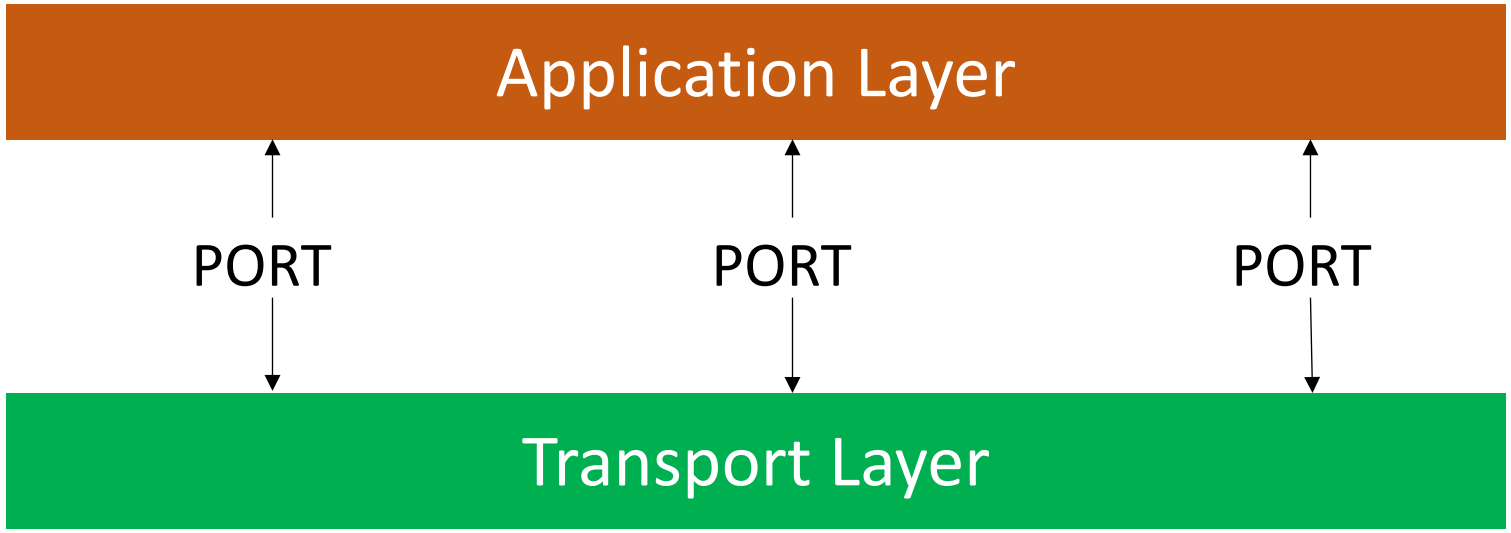
- Connection-oriented: it appears that there is a dedicated connection between two applications
- Reliable: packets are delivered error-free and in sequence
- Byte stream: an arbitrary number of bytes can be exchanged

TCP Goals

- **Connection-oriented:** it appears that there is a dedicated connection between two applications
  - How to identify a connection?
- Reliable: packets are delivered error-free and in sequence
- Byte stream: an arbitrary number of bytes can be exchanged

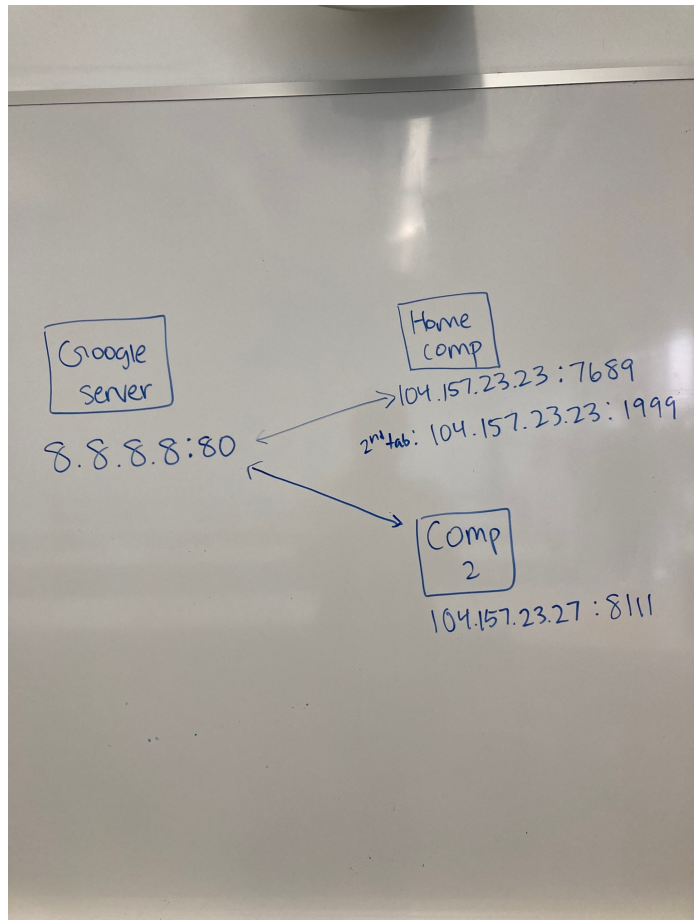
Port Numbers

- IP addresses only identify a device. Port numbers allow a single device to maintain multiple applications
- Connection is identified by pair of IP addresses (src and dst) and pair of ports (src and dst)



Port Numbers

- Port numbers are 16 bits (0 – 65535)
- Many ports numbers are standardized
  - If you want to initiate a TCP connection to browse a website and the server is located at 8.8.8.8 (Google), TCP session request is sent to (8.8.8.8:80) because port 80 is always for web traffic
- Numbers < 1024 are reserved; numbers > 1024 are assigned dynamically (by operating system) upon a new session

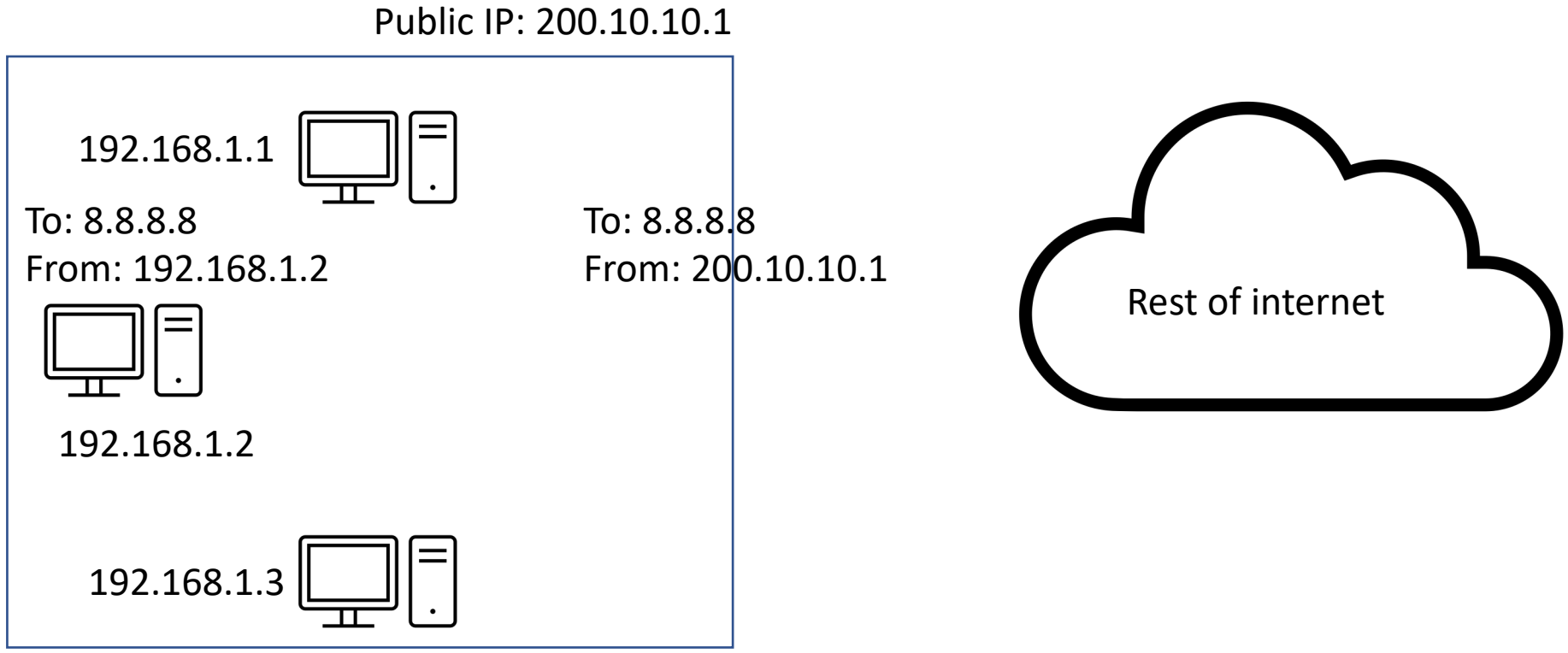

Google server
8.8.8.8:80
Home comp
104.157.23.23:7689
2nd tab: 104.157.23.23:1999
Comp 2
104.157.23.27:8111

Network Address Translation

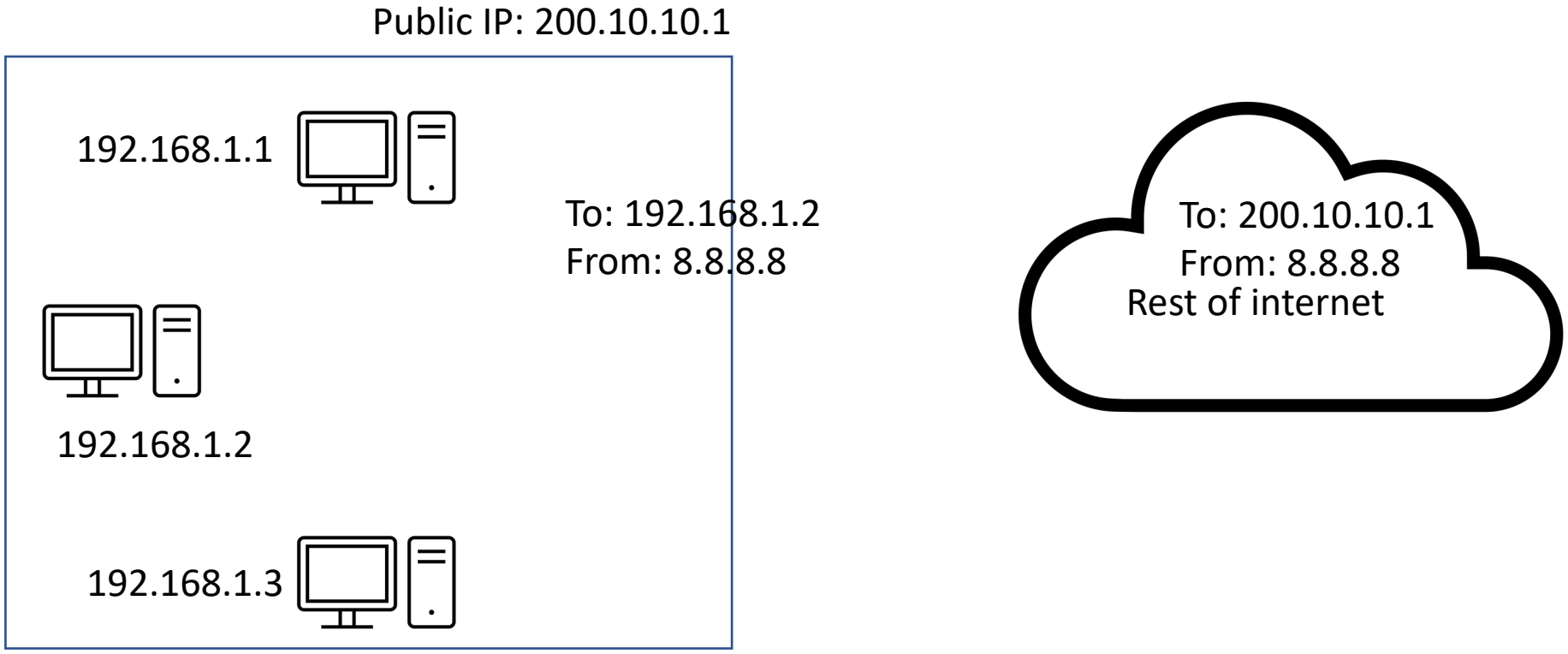
- Closing the loop on our discussion about public/private networks:
- When data address to public IP address arrives at your router, how does the router know that it’s for you?

Network Address Translation

- Use private IP addresses within a network and a single public IP outside

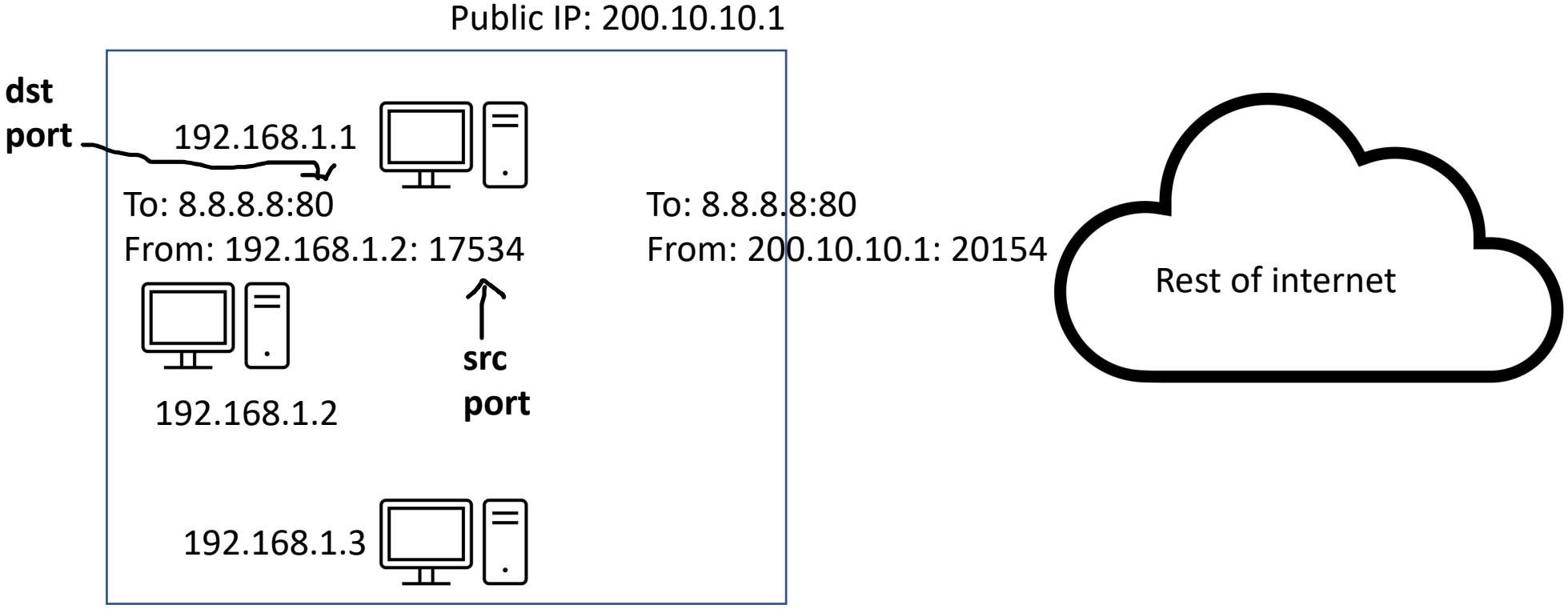


Network Address Translation



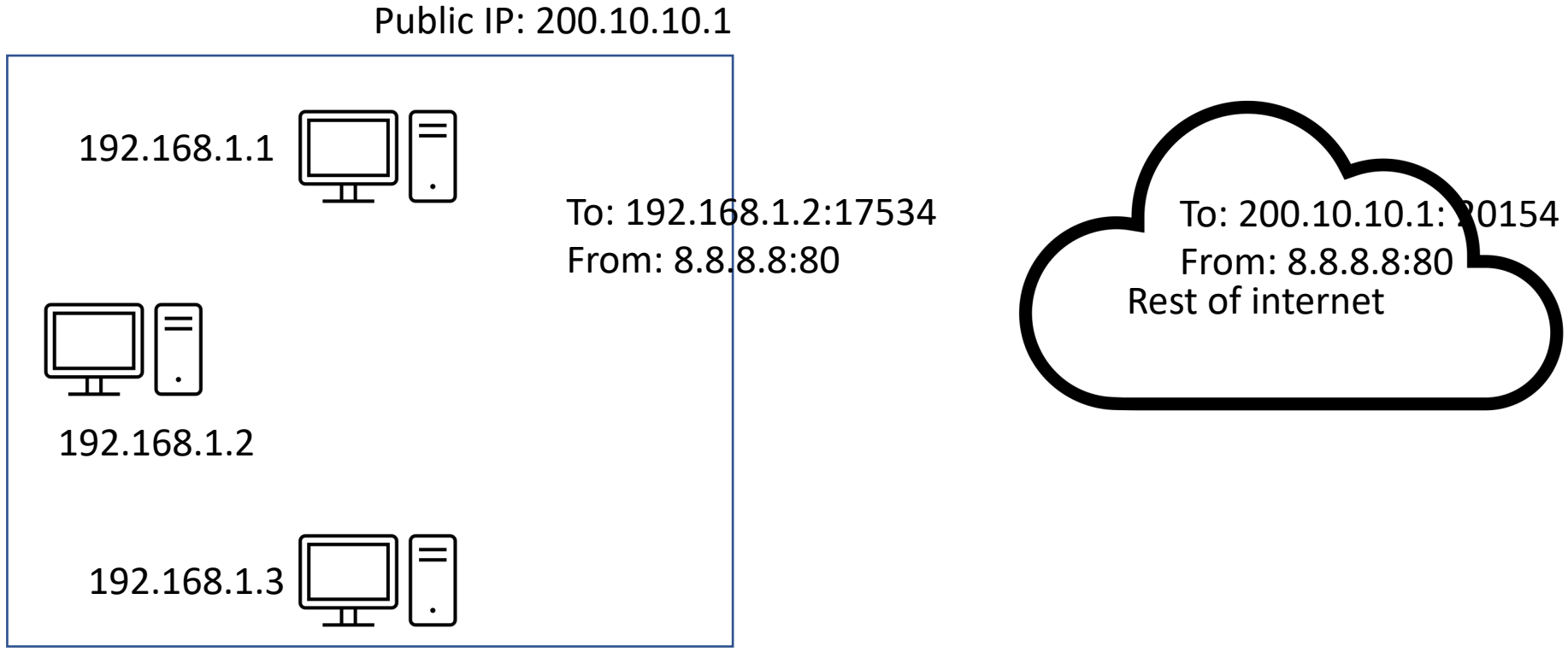
Network Address Translation (in more detail)

- Use IP address and port number to identify the connection
- Router translates IP address AND source port and send out to the internet



Network Address Translation

- Use IP address and port number to identify the connection
- Router translates IP address AND source port and send out to the internet



TCP: End-to-End Reliability

- TCP divides data into packets and then handles:
  - Packet loss
  - Out-of-order arrival

TCP: End-to-End Reliability

- TCP divides data into packets and then handles:
  - Packet loss -> Automatic Repeat Request (ARQ)
  - Out-of-order arrival -> Sequence Numbers

Segments and Sequence Numbers

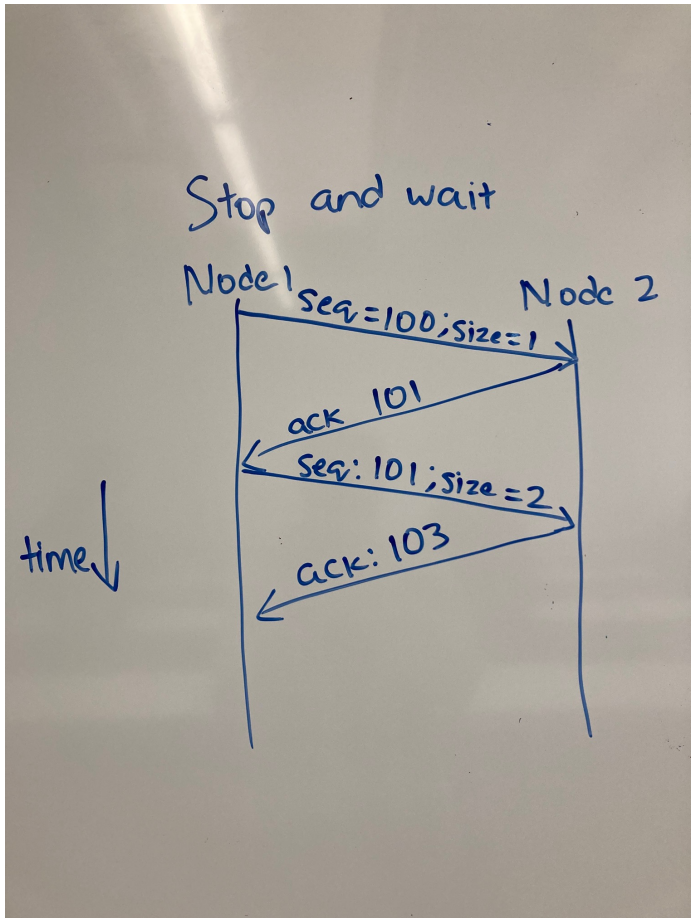
- Application can write any amount of data and TCP will chop it into segments, or packets
- Each segment is assigned a 32-bit sequence number
- Sequence numbers are for bytes, not segments
  - If packet with 5 bytes has sequence number 100, what should next segment's sequence number be?
- Usually sequence numbers start at 0 but they don't have to
- Separate sequence number in each direction

Providing Reliability

- Two types of packets:
  - Data packets
  - Ack packets
- Basic mechanism: data packets are sent and must be acknowledged with an ack
- Ack contains sequence number of the oldest byte it has *not* yet received yet
  - To acknowledge that it received bytes 0 – 7, ack number will be 8

Stop and Wait ARQ

- After sending a packet, wait for ack before sending the next one
- Drawbacks?

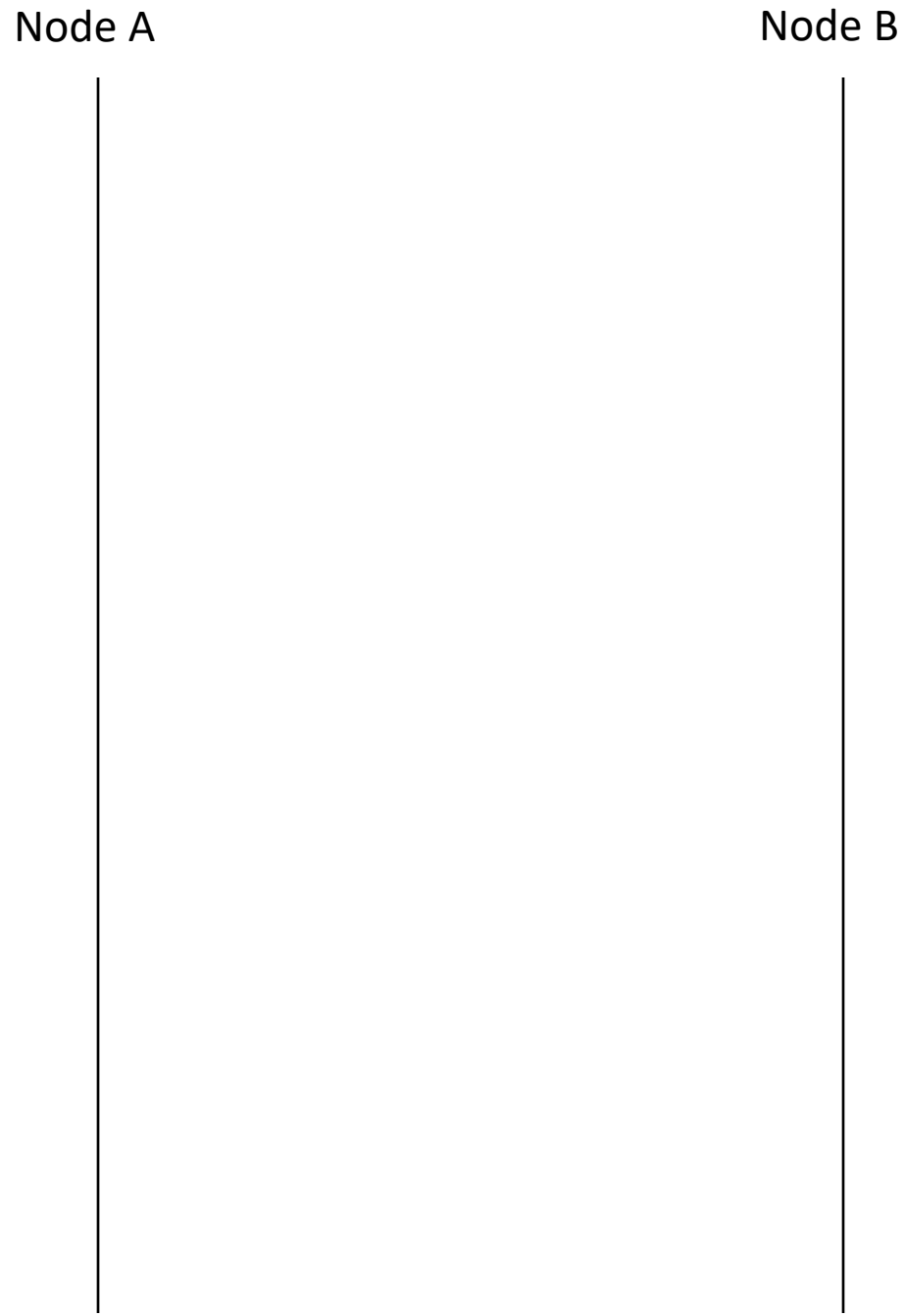


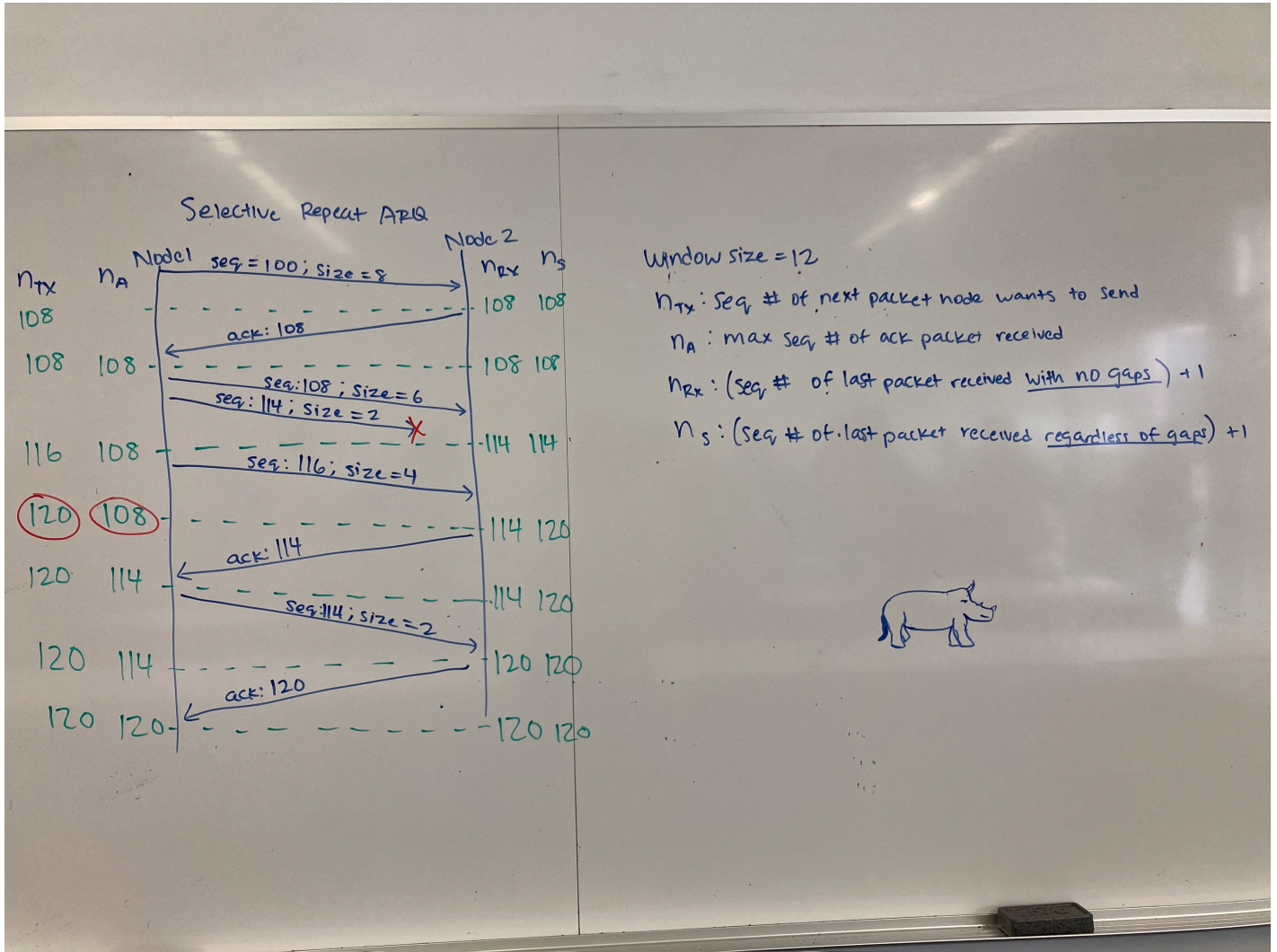
Selective Repeat ARQ

- Sender and receiver agree on a window size (# bytes). This is the number of bytes that can be sent without getting an ack
- Acks are often piggybacked on return data
- Receiver will buffer out-of-order segments

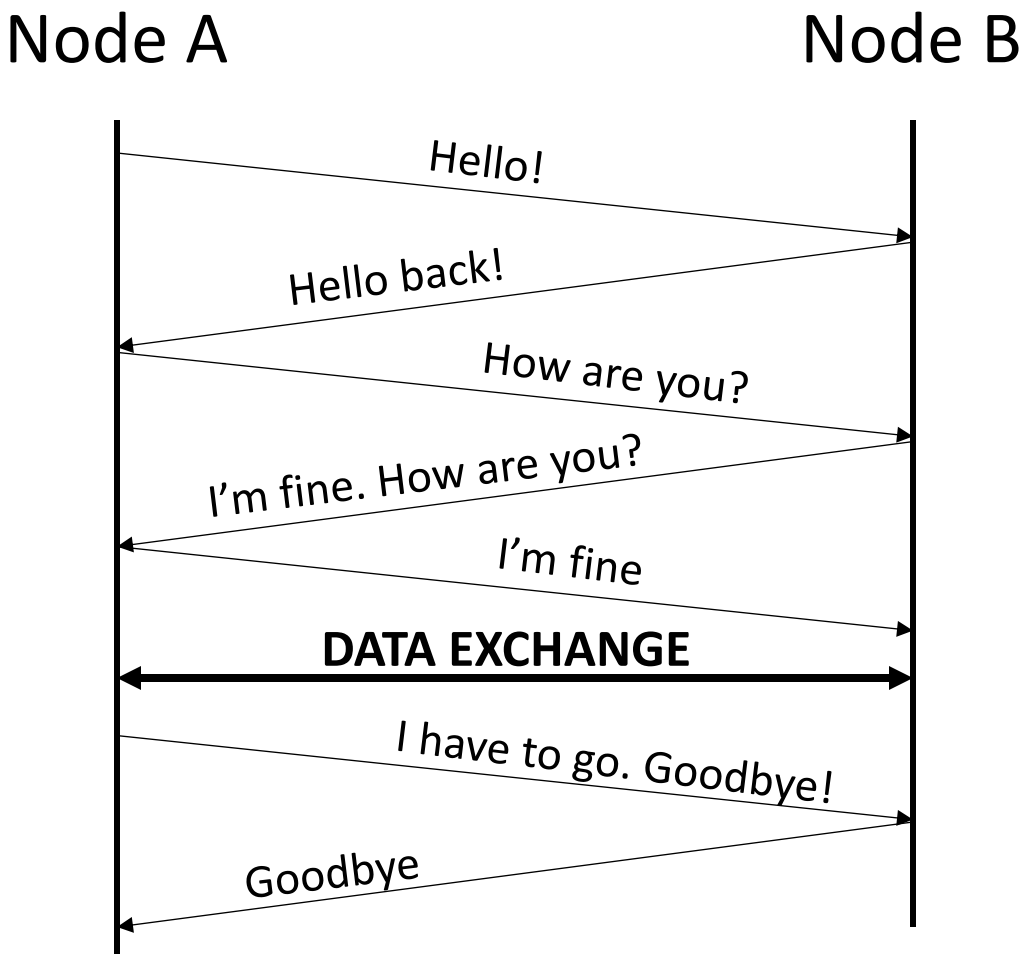
Example

- Window size = 12




Selective Repeat ARQ
Node 1
Node 2
n_{TX}
n_A
n_{Rx}
n_s
seq = 100; size = 8
ack: 108
seq: 108; size = 6
seq: 114; size = 2
seq: 116; size = 4
ack: 114
seq: 114; size = 2
ack: 120
Window size = 12
n_{TX}: Seq # of next packet node wants to send
n_A: max seq # of ack packet received
n_{Rx}: (seq # of last packet received with no gaps) + 1
n_s: (seq # of last packet received regardless of gaps) + 1

Starting a TCP Session



Next Time

- Congestion control
- Starting a session in more detail