Lecture 16: Network Layer Overview, Internet Protocol

COMP 332, Spring 2023

Victoria Manfredi

WESLEYAN UNIVERSITY

Acknowledgements: materials adapted from Computer Networking: A Top Down Approach 7th edition: ©1996-2016, J.F Kurose and K.W. Ross, All Rights Reserved as well as from slides by Abraham Matta at Boston University, and some material from Computer Networks by Tannenbaum and Wetherall.

Today

1. Announcements
   - homework 6 posted
   - midterm: will take Thursday alarm into consideration while grading

2. TCP congestion control

3. Network layer
   - overview
   - what's inside a router
   - Internet protocol (IP)

TCP
CONGESTION CONTROL

3 states in TCP finite state machine

Goal: send segments, adjust cwnd as needed

1. Slow start

– determine available bandwidth starting from no info

2. Congestion avoidance

– deal with fluctuations in bandwidth

3. Fast recovery

– quickly recover from isolated lost packets

We'll first look at different states, then full FSM

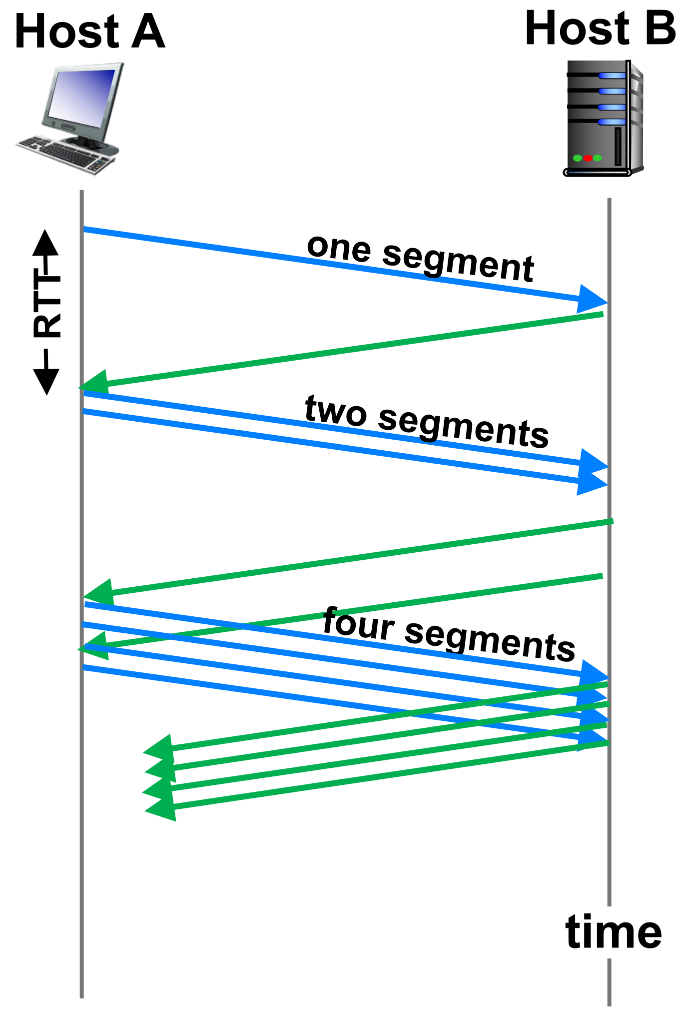
Slow start: initialization

Initial rate is "slow"

- relative to original TCP which had no congestion control
- initially cwnd = 1 MSS

Ramp up exponentially fast

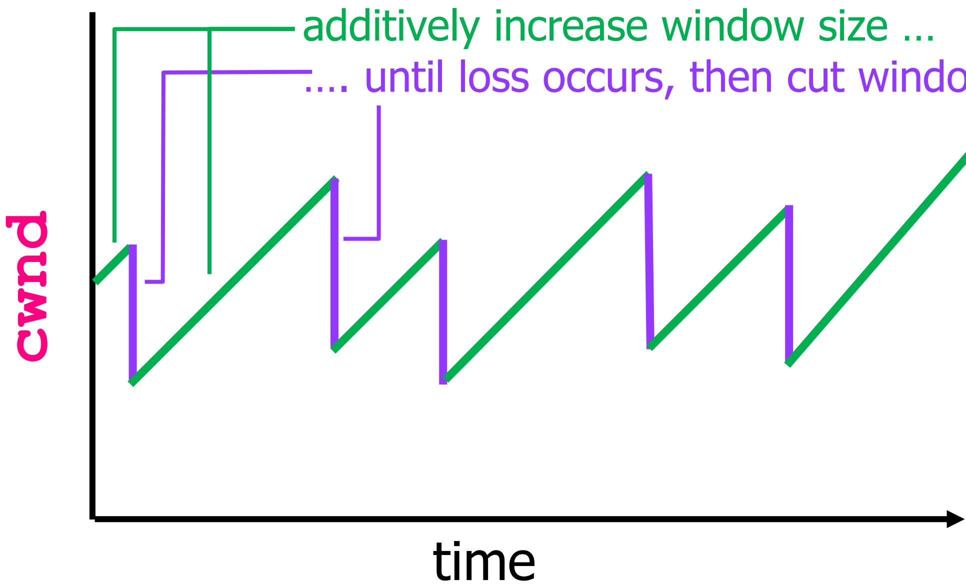
- every time ACK received
  - cwnd = cwnd + MSS
- essentially doubles cwnd every RTT



Congestion avoidance

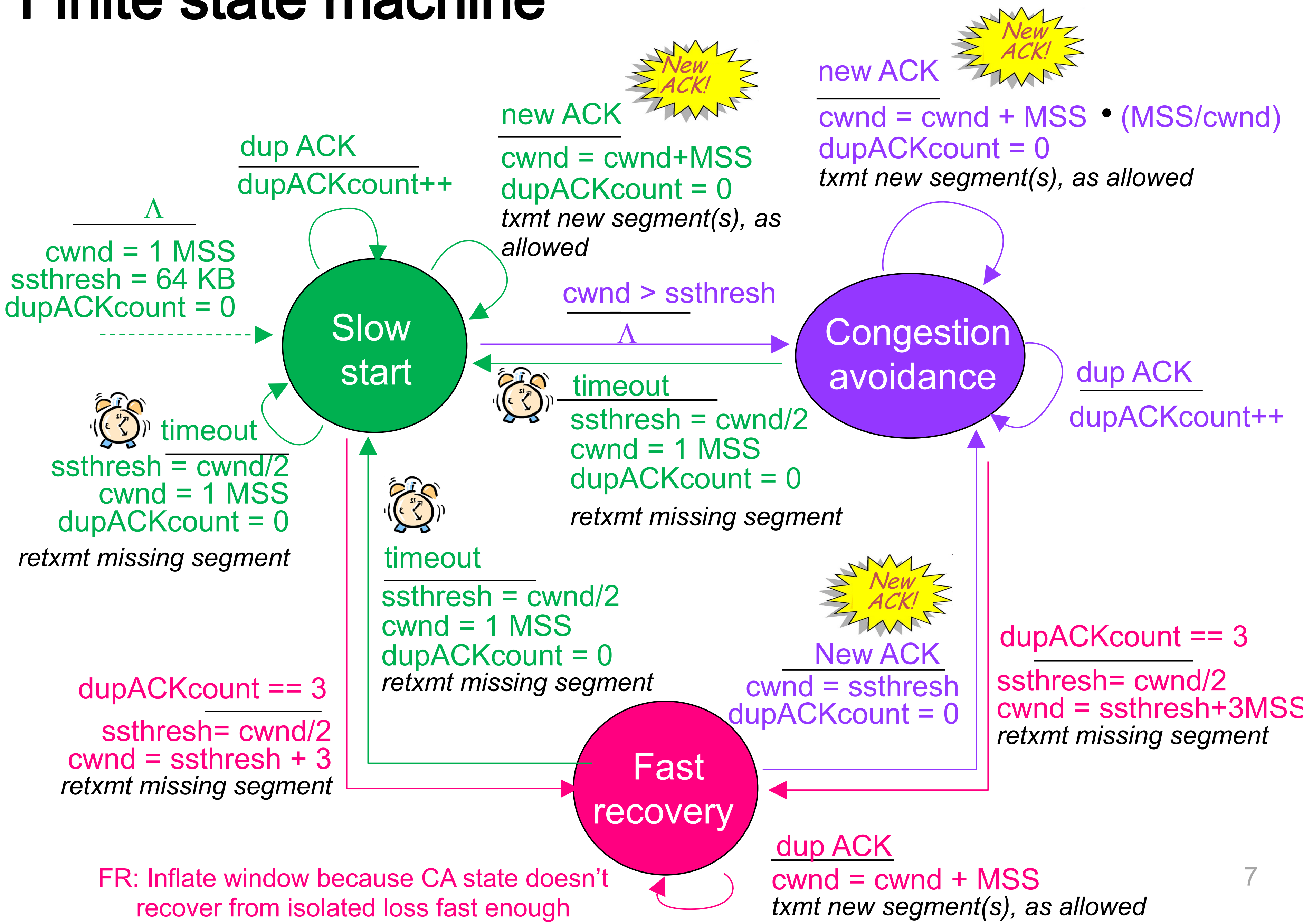
Additive Increase Multiplicative Decrease (AIMD)

- probe cautiously for usable bandwidth
- additive increase
  - **cautious:** increase cwnd by 1 MSS every RTT until loss detected
- multiplicative decrease
  - **aggressive:** cut cwnd in half after loss

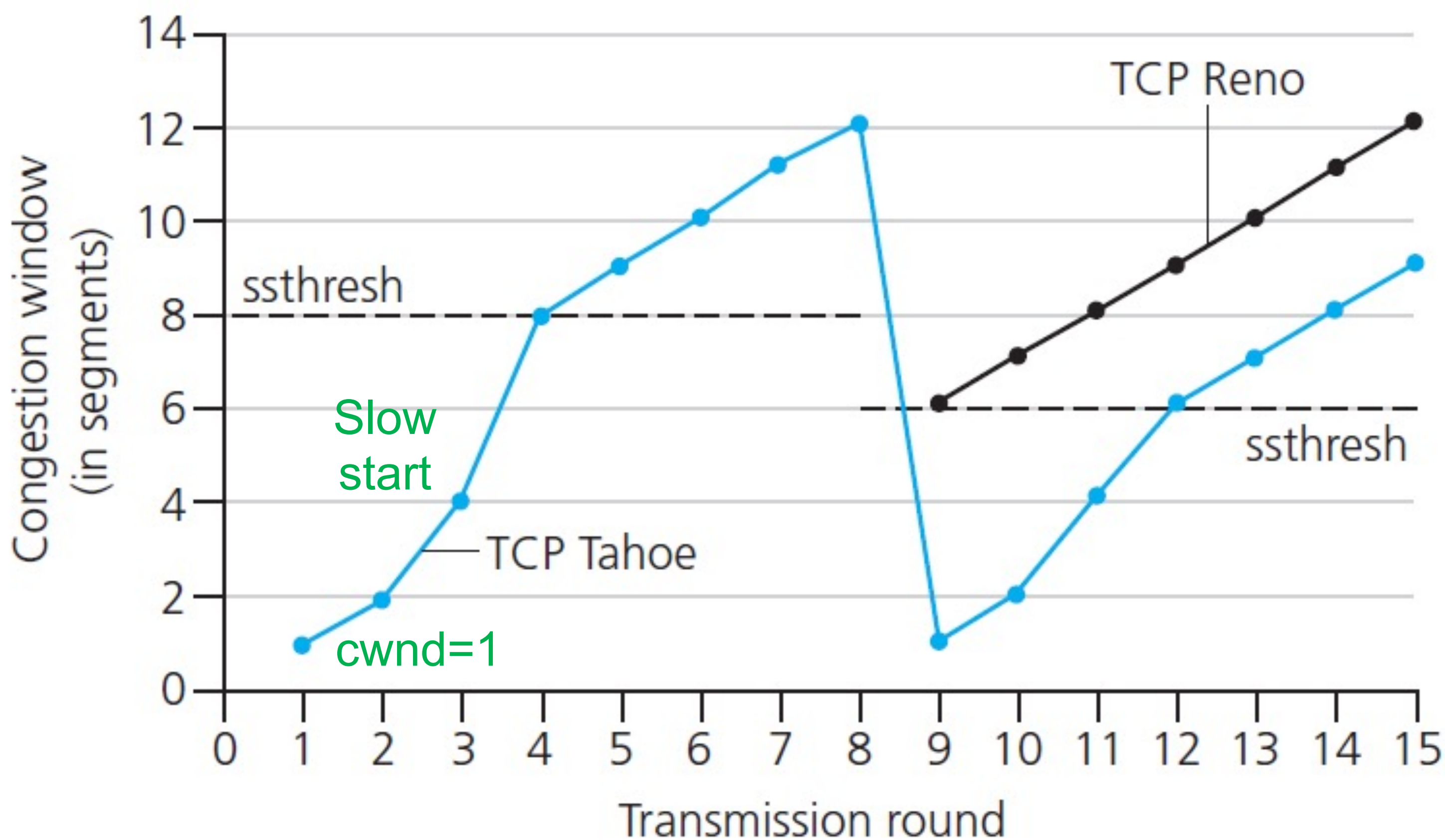


AIMD saw tooth behavior: probing for bandwidth

Finite state machine



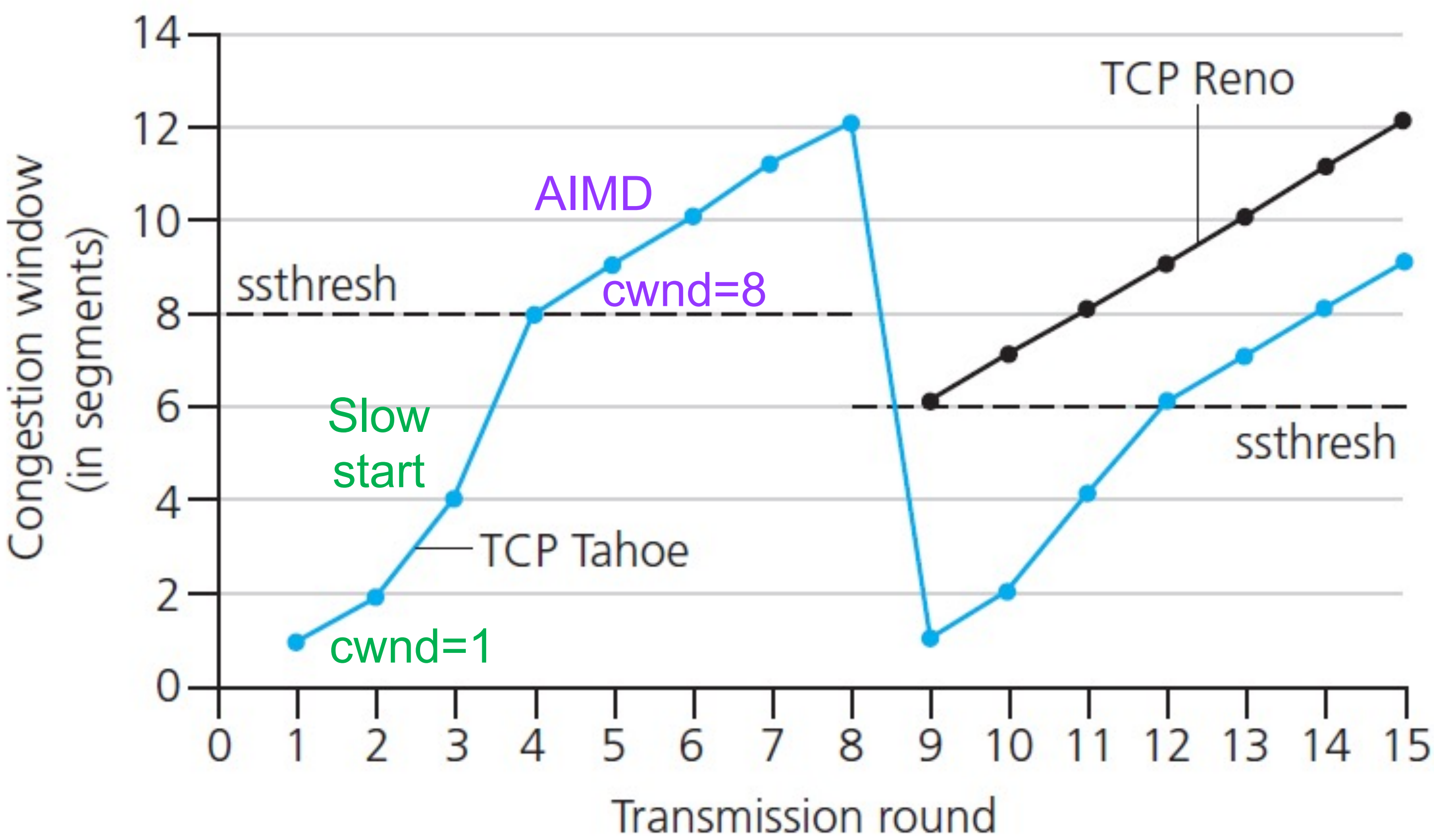
Slow start: when to stop exponential increase?



Slow start

- initially cwnd = 1 MSS
- every time ACK received, double cwnd

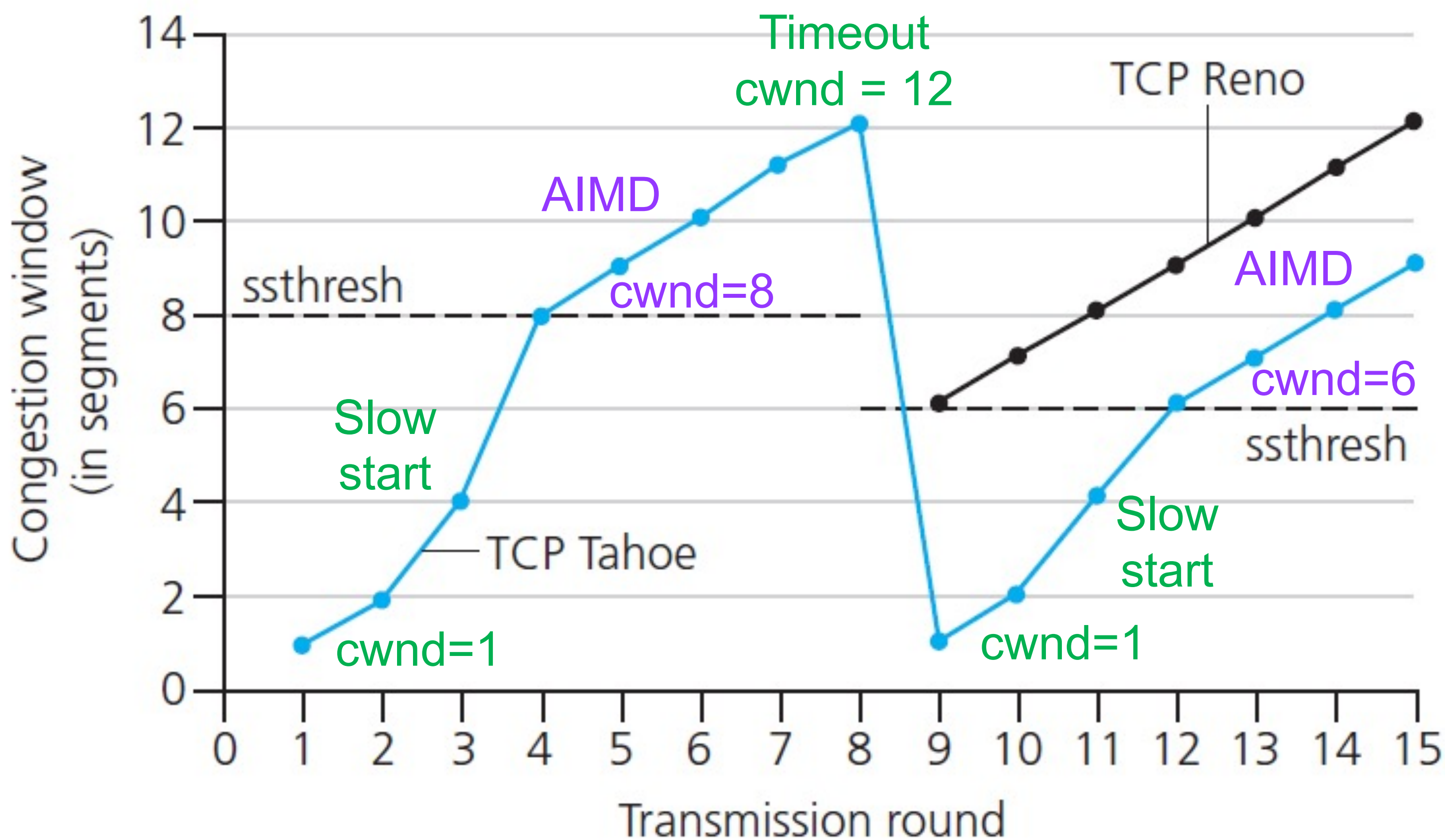
Congestion avoidance



When cwnd = ssthresh

- go to congestion avoidance
- use AIMD

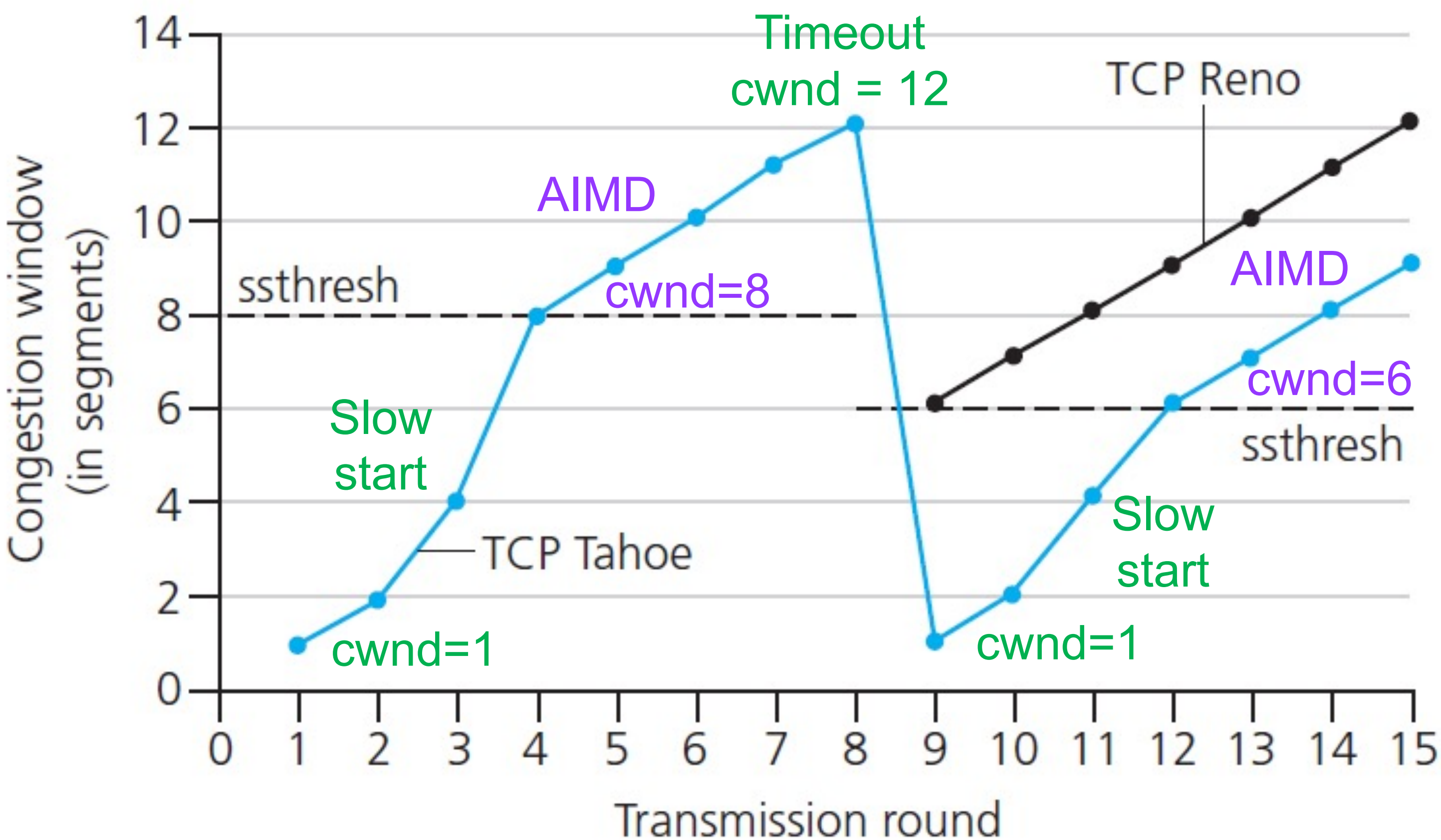
Timeout



Restart slow start when timeout

- ssthresh = cwnd/2
- cwnd = 1 MSS

3 duplicate ACKs



If 3 duplicate ACKs go to fast recovery

- ssthresh = cwnd/2
- cwnd = ssthresh + 3 MSS

Average TCP throughput

Focus on AIMD

- ignore slow start, assume always data to send

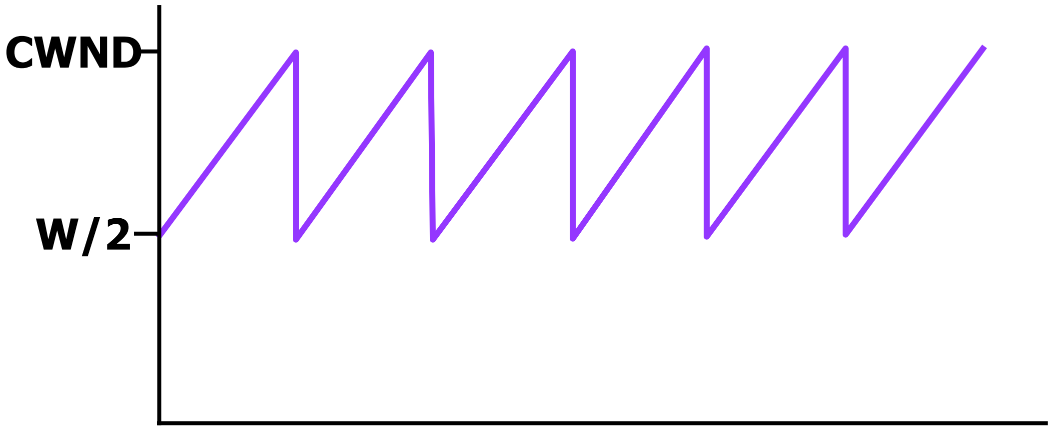
Max rate

- cwnd / RTT

3 dup loss rate

- 0.5 cwnd / RTT

Avg TCP thruput = $\frac{3}{4} \frac{CWND}{RTT}$ bytes/sec



Setting window size

Window is min (rwnd, cwnd)

```
▼ Transmission Control Protocol, Src Port: 443 (443), Dst Port: 52232 (52232), Seq: 0, Ack: 1,
    Source Port: 443
    Destination Port: 52232
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 1    (relative ack number)
    Header Length: 32 bytes
  ▼ Flags: 0x012 (SYN, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ▶ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
      [TCP Flags: *******A**S*]
    Window size value: 8190
    [Calculated window size: 8190]
```

rwnd

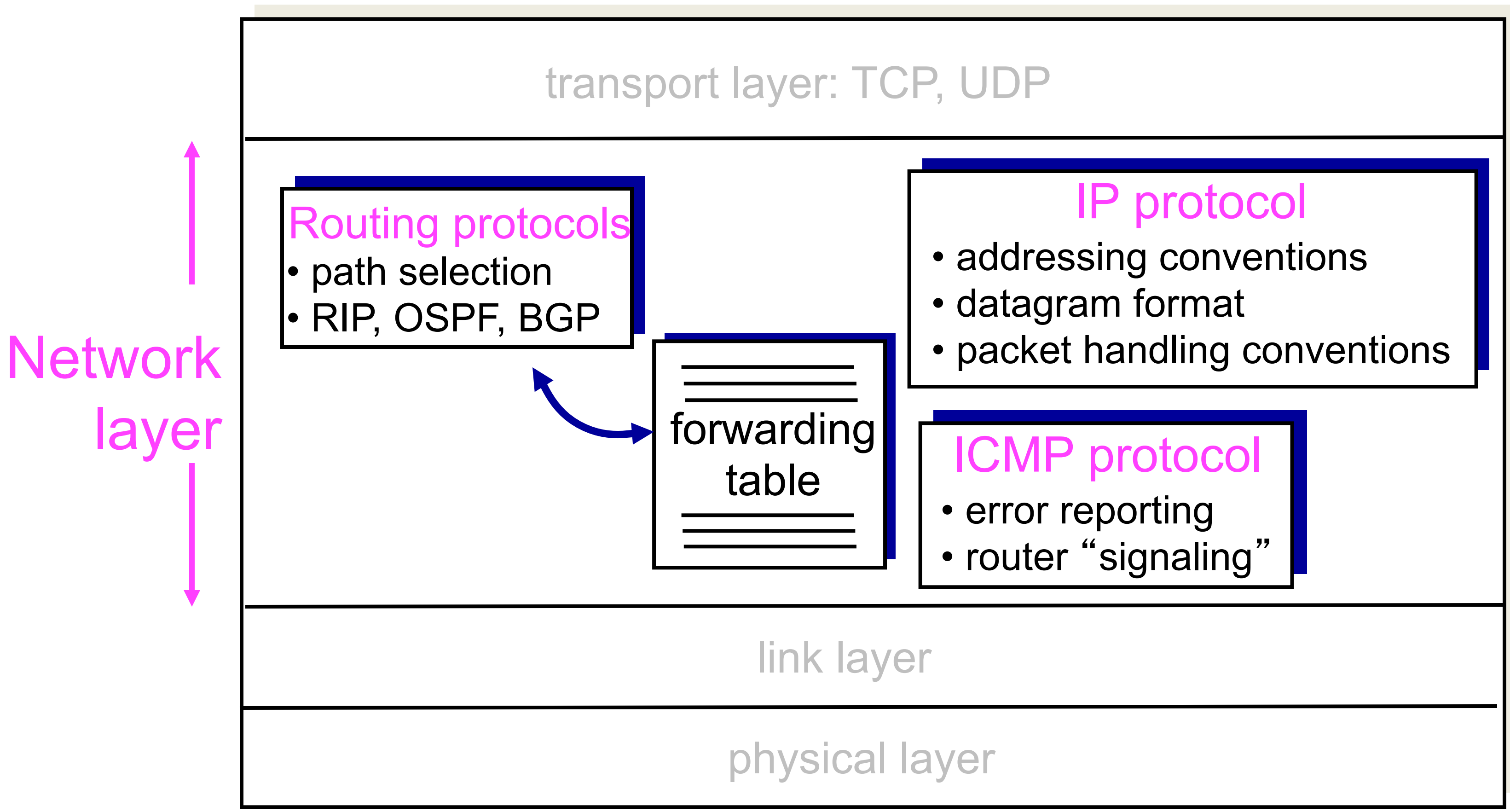
Network Layer
OVERVIEW

5-layer Internet protocol stack

	Layer	Service provided to upper layer	Protocols	Unit of information
5	Application	• Support network applications	FTP, DNS, SMTP, HTTP	**Message** 1 message may be split into multiple segments
4	**Transport**	• Deliver messages to app endpoints • Flow control • Reliability	TCP (reliable) UDP (best-effort)	**Segment** (TCP) **Datagram** (UDP) 1 segment may be split into multiple packets
3	**Network**	• Route segments from source to destination host	IP (best-effort) Routing protocols	**Packet** (TCP) **Datagram** (UDP)
2	**Link**	• Move packet over link from one host to next host	Ethernet, 802.11	**Frame** MTU is 1500 bytes
1	**Physical**	• Move individual bits in frame from one host to next • “bits on wire”	Ethernet phy 802.11 phy Bluetooth phy DSL	**Bit**

Internet's network layer

Network layer functions on hosts and routers



Network layer

Goal

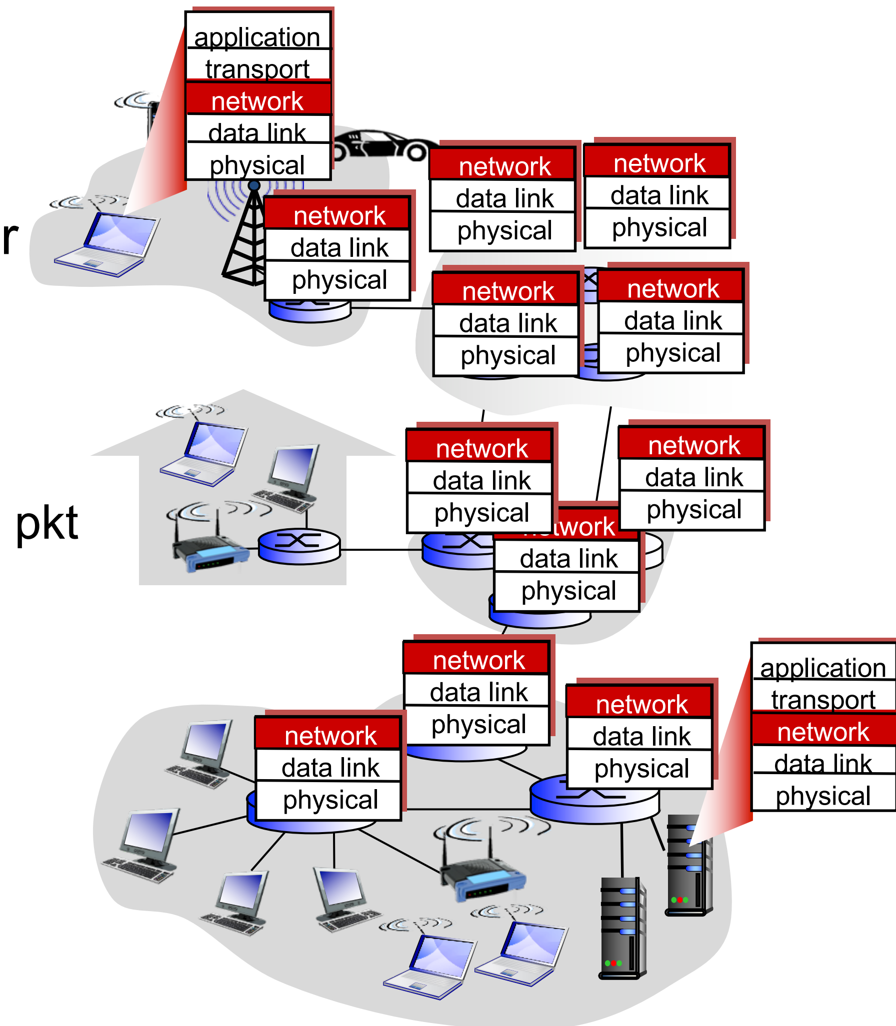
- move pkt from one host to another

How done on Internet?

- routers
  - examine header fields in every IP pkt
  - determines outgoing link

Internet e2e argument

- some functionality only properly implemented in end systems
- smart hosts vs. dumb routers



Network layer is in every host and router on Internet

Encapsulation and decapsulation

Sender

- encapsulates segments into packets, puts src, dest IP in IP pkt hdr

Receiver

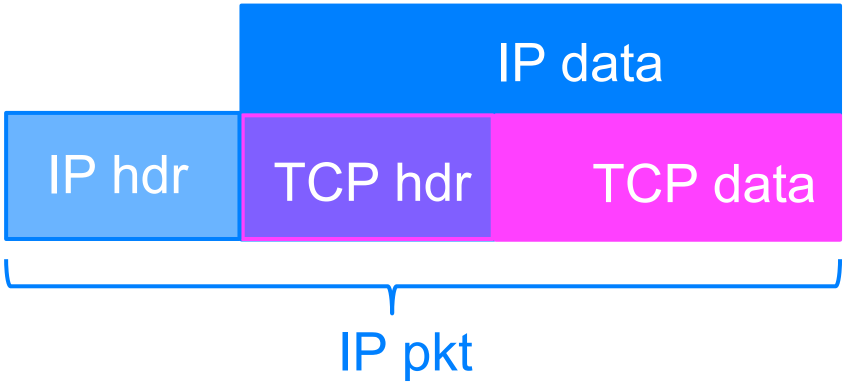
- decapsulates packets into segments, delivers to transport layer

Max length of IP packet in bytes

- MTU: Maximum Transmission Unit
- 1500 bytes if Ethernet used as link layer protocol

Max length of TCP data in bytes

- MSS: Maximum Segment Size
- MSS = MTU – IP hdr – TCP hdr
  - TCP header >= 20bytes



Division of network layer functionality

1. Control plane
   - comprises traffic only between routers, to compute routes between src and dst
   - network-wide: routers run routing algorithms

2. Data plane
   - comprises traffic between end hosts, forwarded by routers
   - forwarding table set based on routes computed in control plane
   - local: each router stores, forwards packets

Interplay between routing and forwarding

Routing (slower time scale)

- routers view Internet as graph
- run shortest path algorithms

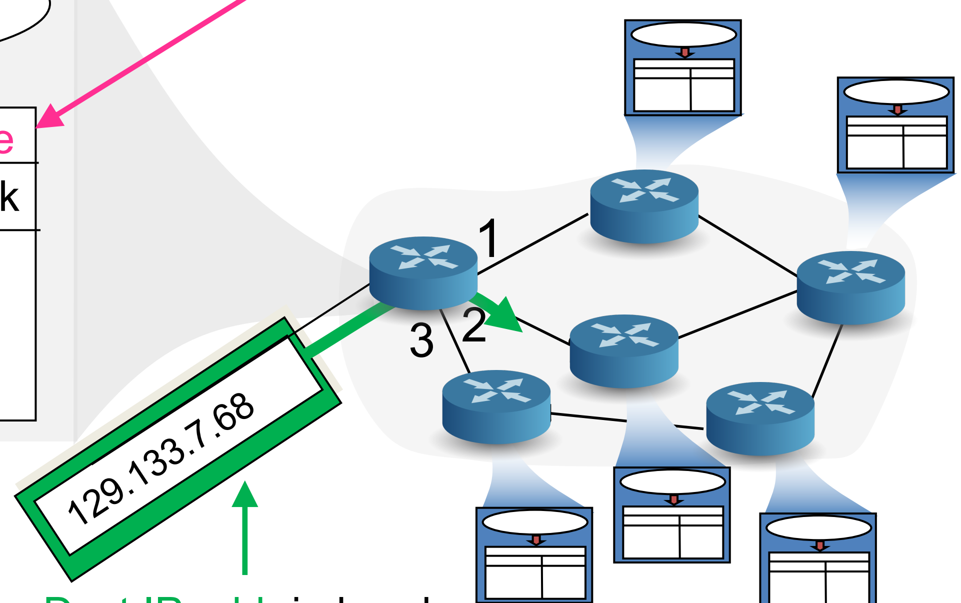
Forwarding (faster time scale)

- routers use paths to choose best output link for packet destination IP address
- if one link fails, chooses another

Routing algorithm

Local forwarding table

Dest IP	Output link
129.133.*.*	2
43.*.*.*	3
43.56.*.*	3
189.37.35.*	1



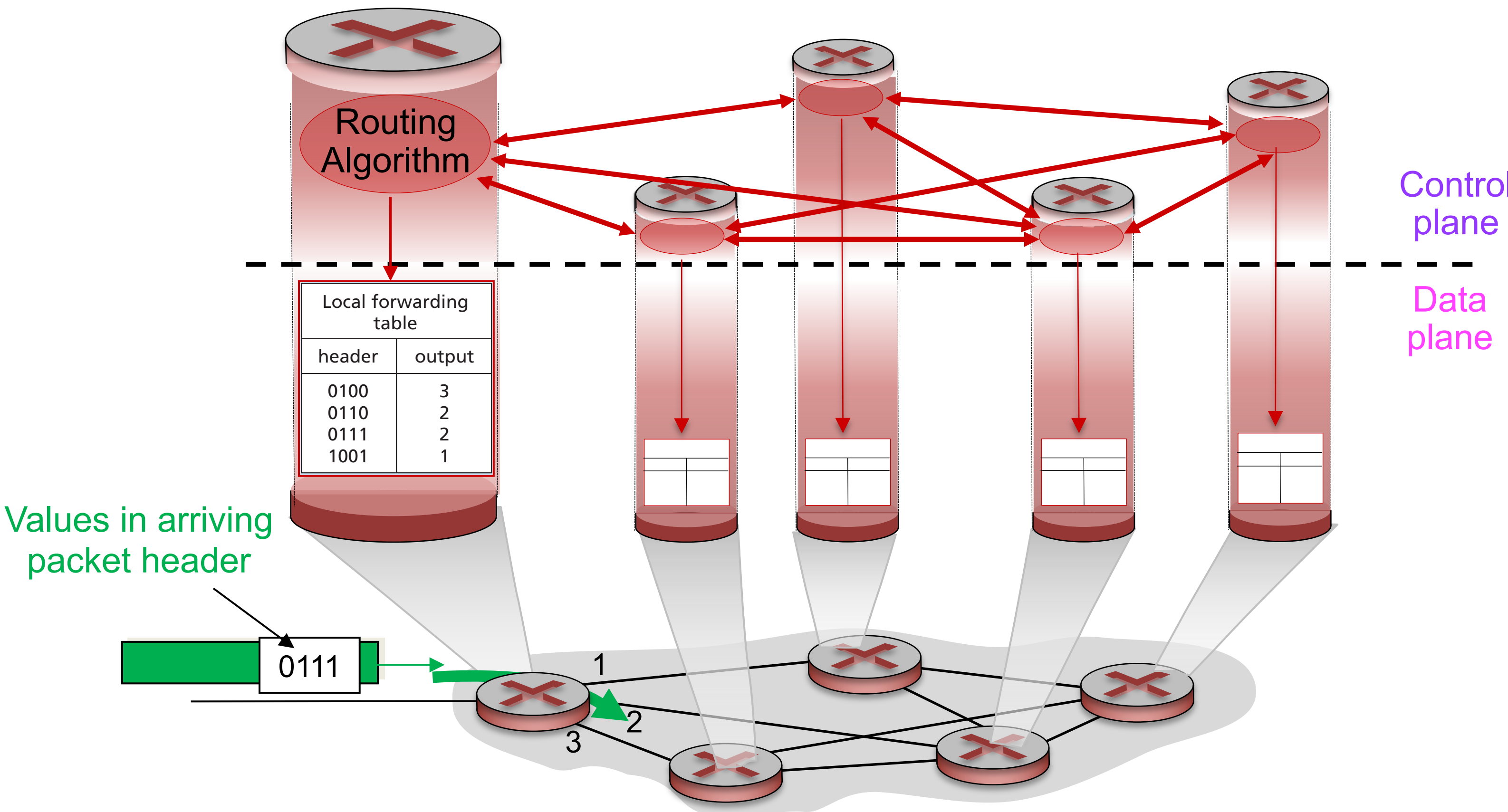
Dest IP addr in header of arriving packet

2 control-plane approaches

1. traditional routing algorithms implemented in routers
2. software-defined networking (SDN) implemented in (remote) servers

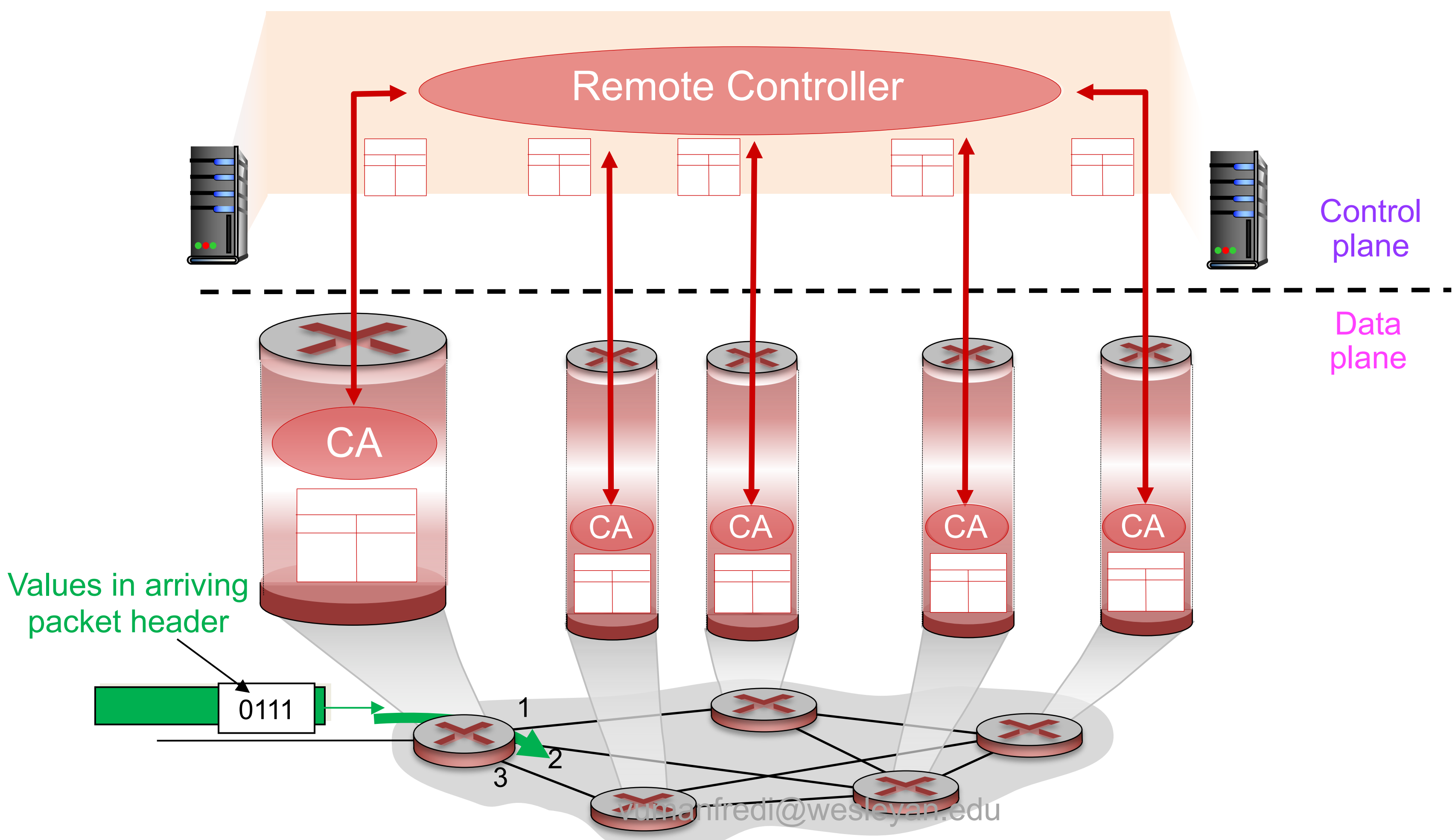
Approach 1: per-router control plane

Individual routing algorithm components in each and every router interact in the control plane



Approach 2: logically centralized control plane

A distinct (typically remote) controller interacts with local control agents (CAs)



Network layer service model

Q: What service model does network layer provide to transport layer for moving packets from sender to receiver?

Example services

- individual packets
  - guaranteed delivery
  - guaranteed delivery with less than 40 ms delay
- flow of packets
  - in-order packet delivery
  - guaranteed minimum bandwidth to flow
  - restrictions on changes in inter-packet spacing

Network layer service models

Network Architecture	Service Model	Guarantees ? Bandwidth	Loss	Order	Timing	Congestion feedback
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

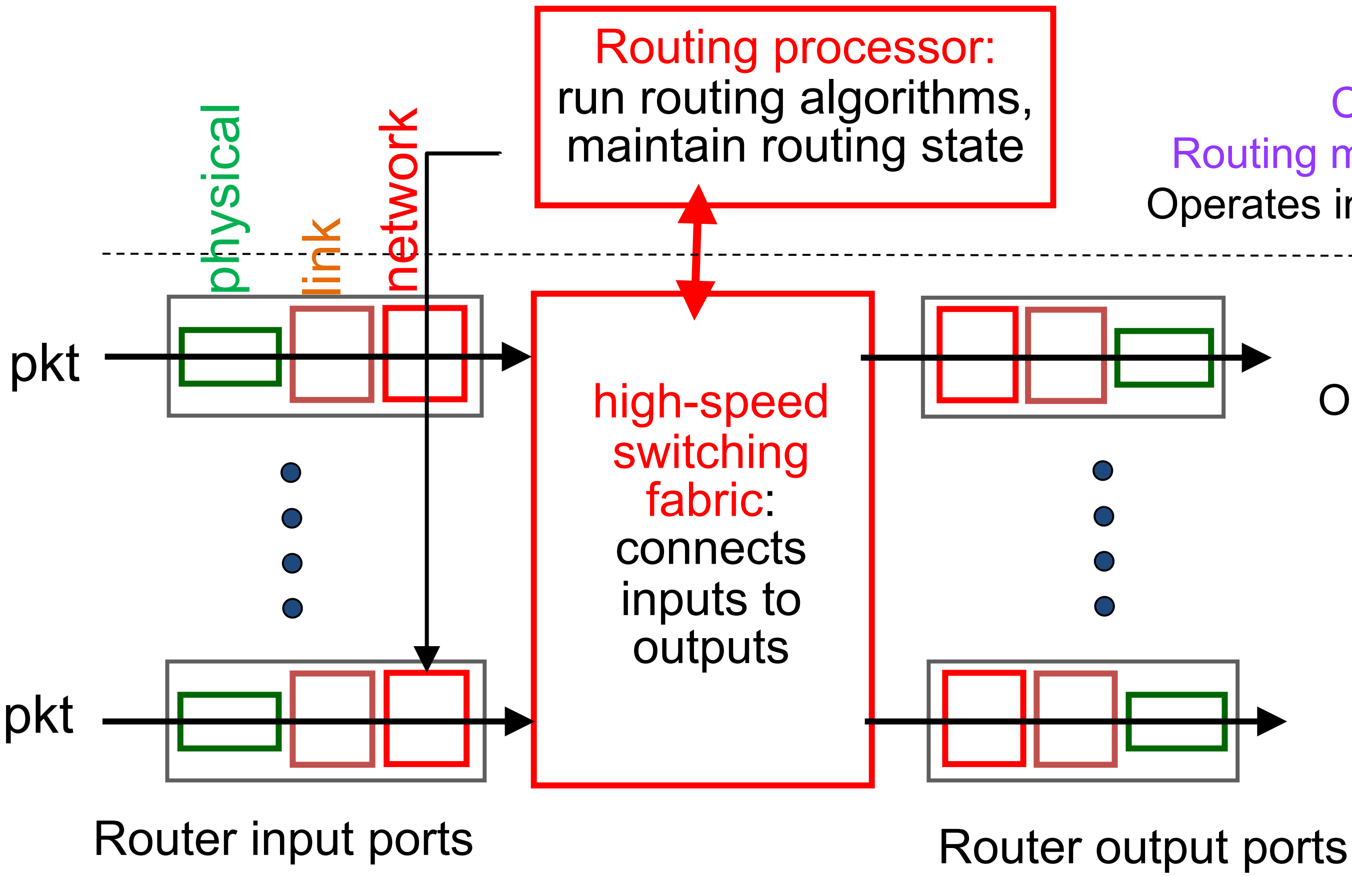
ATM: Asynchronous Transfer Mode
e.g., used in public switched telephone network

Network Layer

WHAT'S INSIDE A ROUTER?

What does a router need to do?

Run routing protocols (control) and store and forward pkts (data)



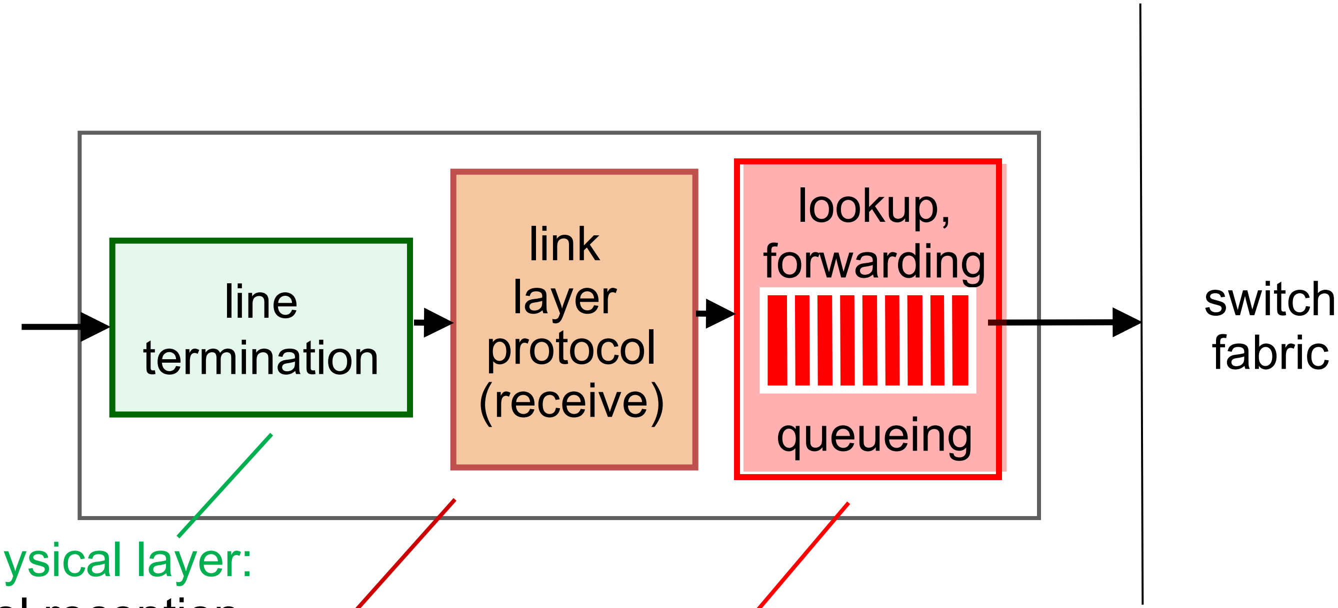
Control plane - complex
Routing management (software)
Operates in millisecond timescale

Data plane - simple
Forwarding (hardware)
Operates in nanosecond timescale

High-level view of generic router architecture

Port is an interface not a socket

Input port functions



Physical layer:
bit-level reception,
terminate phys. conn.

Data link layer:
e.g., Ethernet processing,
error-checking, de-capsulation,

Network layer

- validate/update checksum, decrement TTL
- switching: use header field values, lookup output port
- queue: if packets arrive faster than forwarding rate into switch fabric

Switching fabrics

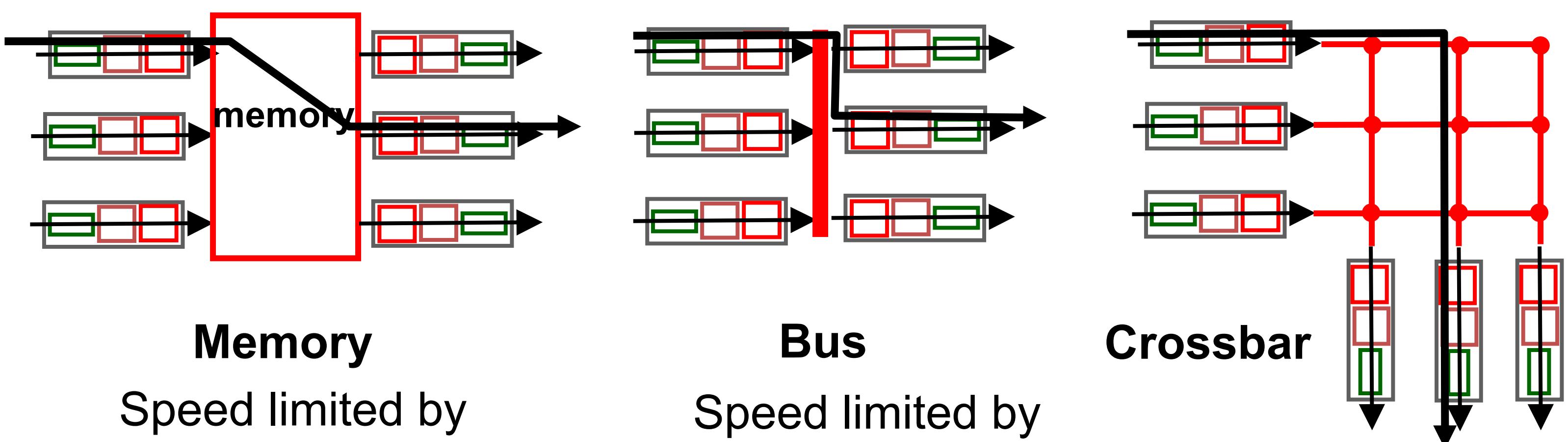
Transfer packet

- from input buffer to appropriate output buffer

Switching rate

- rate at which packets can be transferred from inputs to outputs
- N inputs: switching rate = N x line rate desirable

3 types of switching fabrics



Memory

Speed limited by memory bandwidth

Bus

Speed limited by bus contention

Crossbar

Forward multiple pkts in parallel

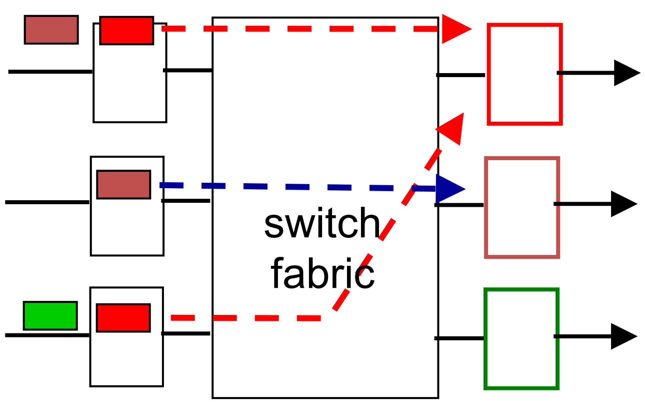
Contention at input ports

If switching fabric slower than input ports combined

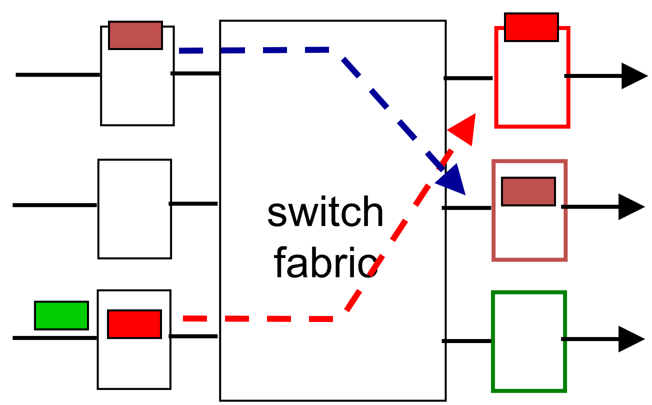
- queueing may occur at input queues
- queueing delay and loss due to input buffer overflow!

Head-of-the-Line (HOL) blocking

- queued pkt at front of queue prevents others from moving forward

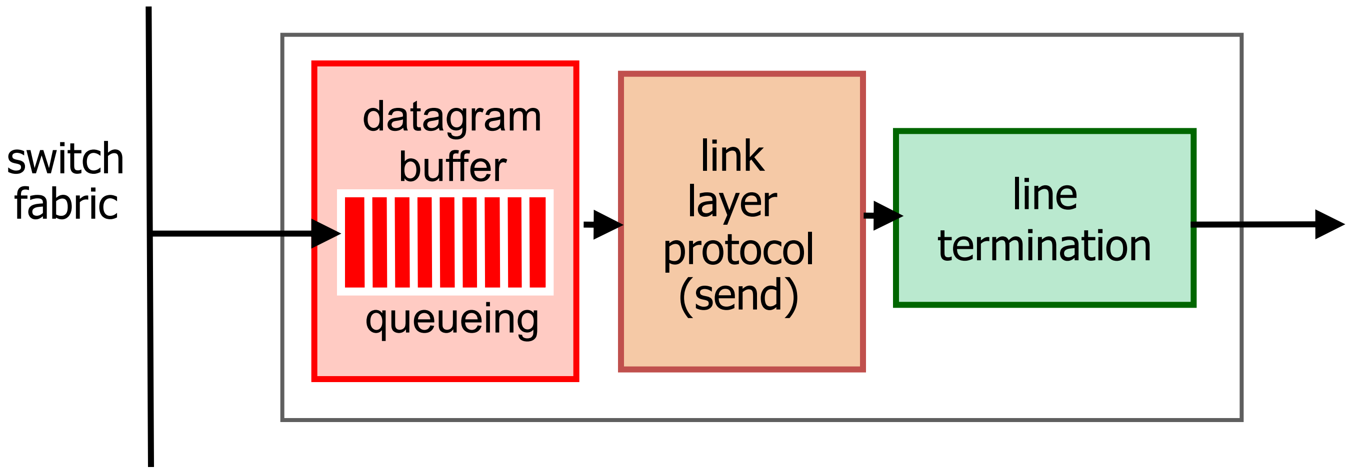


Output port contention: only one red packet can be transferred.
Lower red packet is blocked



One packet time later: green packet experiences HOL blocking

Contention at output ports



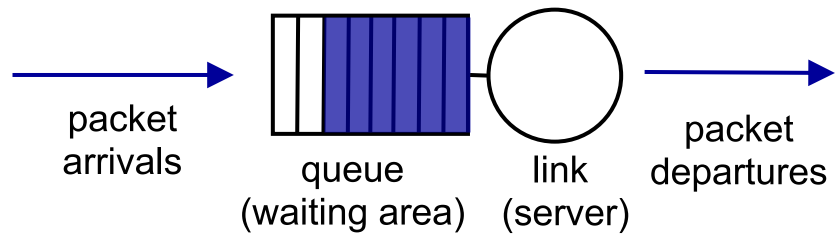
Buffering

- when packets arrive from fabric faster than transmission rate
- packet loss: due to congestion, lack of buffers

Scheduling

- chooses next among queued packets to transmit on link
- net neutrality: who gets best performance

Scheduling mechanisms



FIFO (first in first out)

- send in order of arrival to queue

Priority

- multiple classes, with different priorities (e.g., based on hdr info)
  - send highest priority queued packet

Round robin scheduling

- multiple classes, cyclically scan class queues
  - send one packet from each class (if available)

Weighted fair queueing

- generalized round robin
  - each class gets weighted amount of service in each cycle

In practice: hardware queues use FIFO, need software to do priority

Network Layer

INTERNET PROTOCOL

Internet Protocol (IP)

THE network layer protocol of the Internet

- protocol your device **must** implement to run on Internet
- RFC published ~1980

Provides

- best effort service
  - to get pkts from one end host to another across many interconnected networks using dst IP address in IP hdr
- addressing
  - format and usage of addresses
- fragmentation
  - e.g., if pkt size exceeds Ethernet MTU of 1500 bytes
- some error detection

Q: what does IP not provide?

- QoS, reliability, ordering, persistent state for e2e flows, connections

IP packet format

Q: Why is version 1st?
So you know how to parse pkt

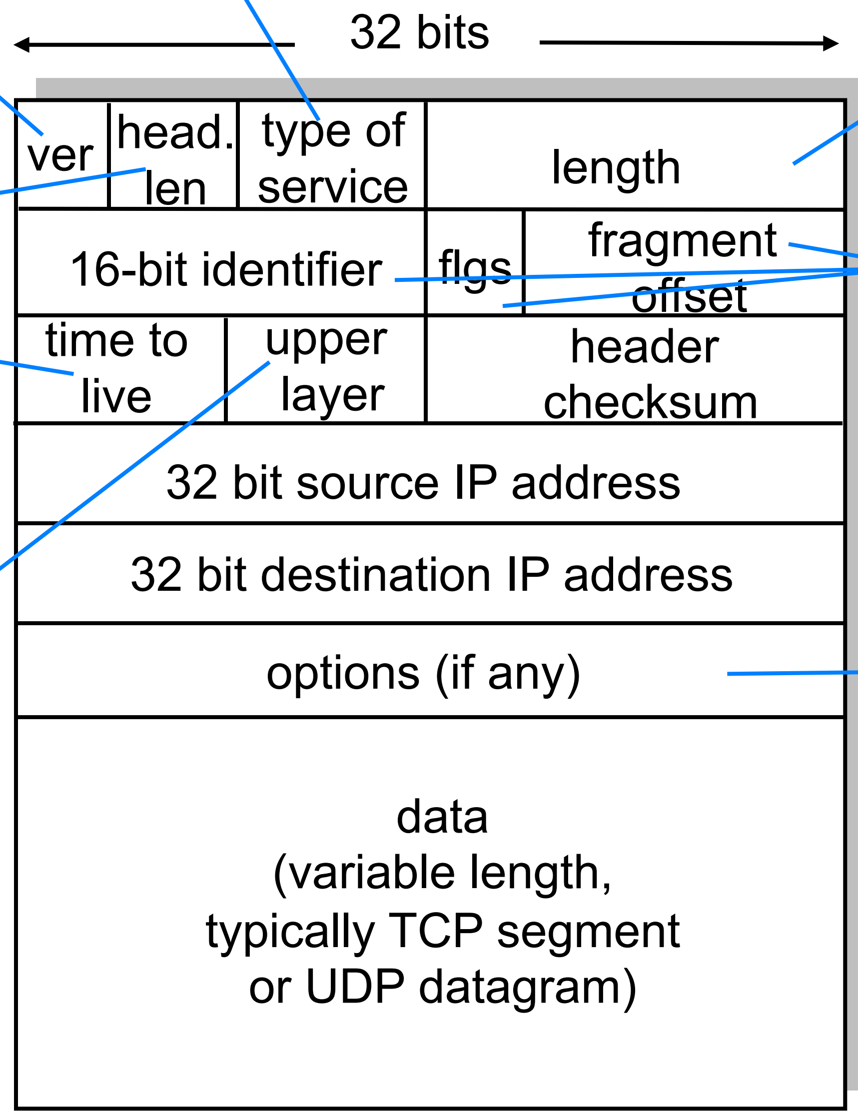
Not widely used

IP protocol version number

header length (# of 32bit words)

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to
ICMP=1, UDP=17, TCP=6



total packet length (bytes)

for fragmentation/ reassembly

e.g. timestamp, record route taken, specify routers to visit. When used usually indicate suspicious activity

How much overhead?
20 bytes of TCP
20 bytes of IP
= 40 bytes + app layer overhead

Bits transmitted left to right, top to bottom

Wireshark

Look at IP headers and ping/traceroute