Lecture 15: Network Layer Overview, Internet Protocol

COMP 332, Spring 2024

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

Announcements

- Homework 6 posted, due next Wed.

Network layer

- Overview
- What's inside a router?
- Internet Protocol (IP)

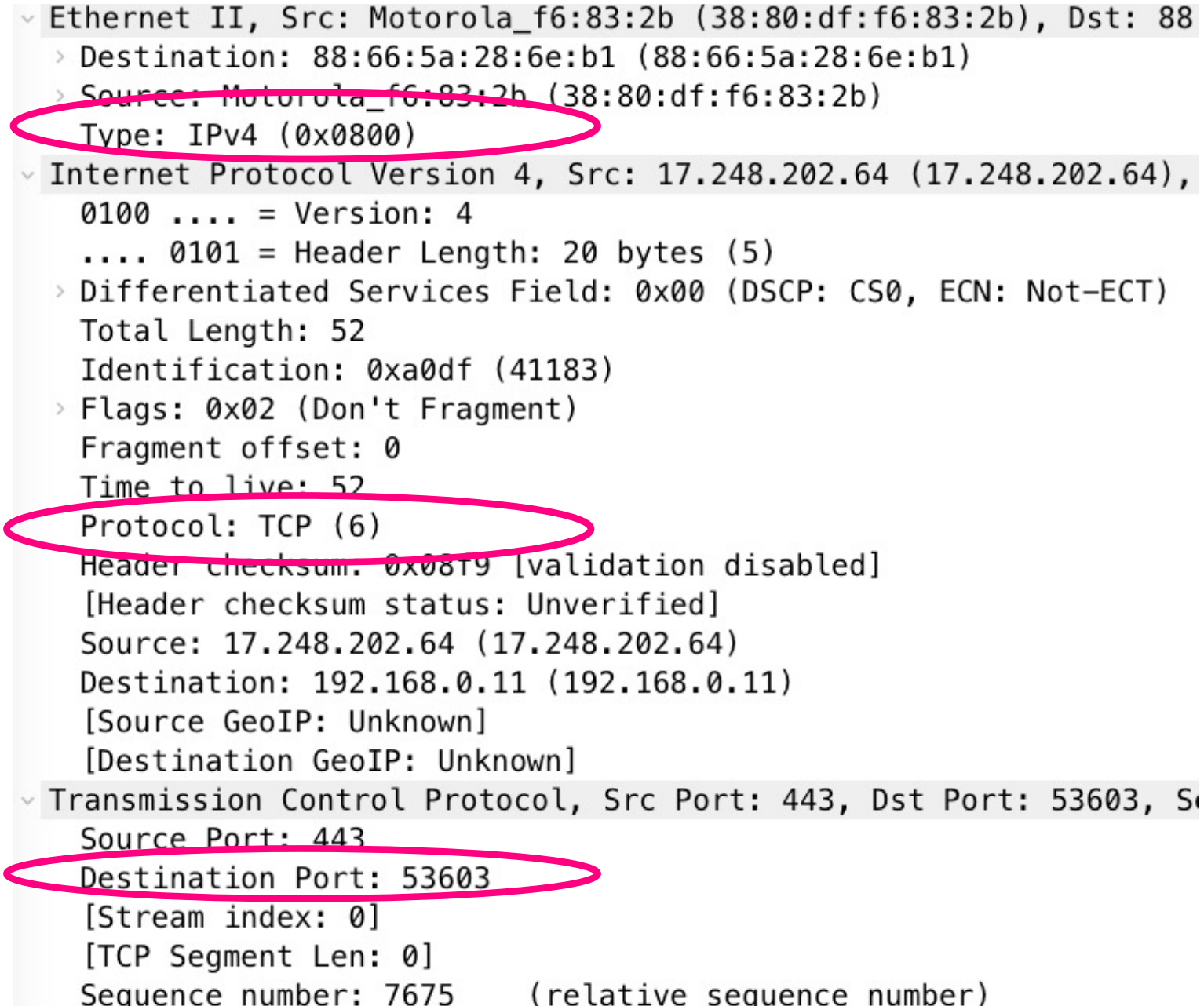
Wireshark

WHAT IS THE UPPER LAYER PROTOCOL?

Directing bits, frames, packets, segments …

```
Frame 30: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
  Interface id: 0 (en0)
  Encapsulation type: Ethernet (1)
  Arrival Time: Mar 22, 2024 09:06:06.881620000 EDT
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1711112766.881620000 seconds
  [Time delta from previous captured frame: 0.016491000 seconds]
  [Time delta from previous displayed frame: 0.016491000 seconds]
  [Time since reference or first frame: 0.273634000 seconds]
  Frame Number: 30
  Frame Length: 66 bytes (528 bits)
  Capture Length: 66 bytes (528 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp]
  [Coloring Rule Name: TCP]
  [Coloring Rule String: tcp]
Ethernet II, Src: Motorola_f6:83:2b (38:80:df:f6:83:2b), Dst: 88:66:5a:28:6e:b1 (88:
  Destination: 88:66:5a:28:6e:b1 (88:66:5a:28:6e:b1)
  Source: Motorola_f6:83:2b (38:80:df:f6:83:2b)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 17.248.202.64 (17.248.202.64), Dst: 192.168.0.11 (
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 52
  Identification: 0xa0df (41183)
  Flags: 0x02 (Don't Fragment)
```

Directing bits, frames, packets, segments …



Network Layer

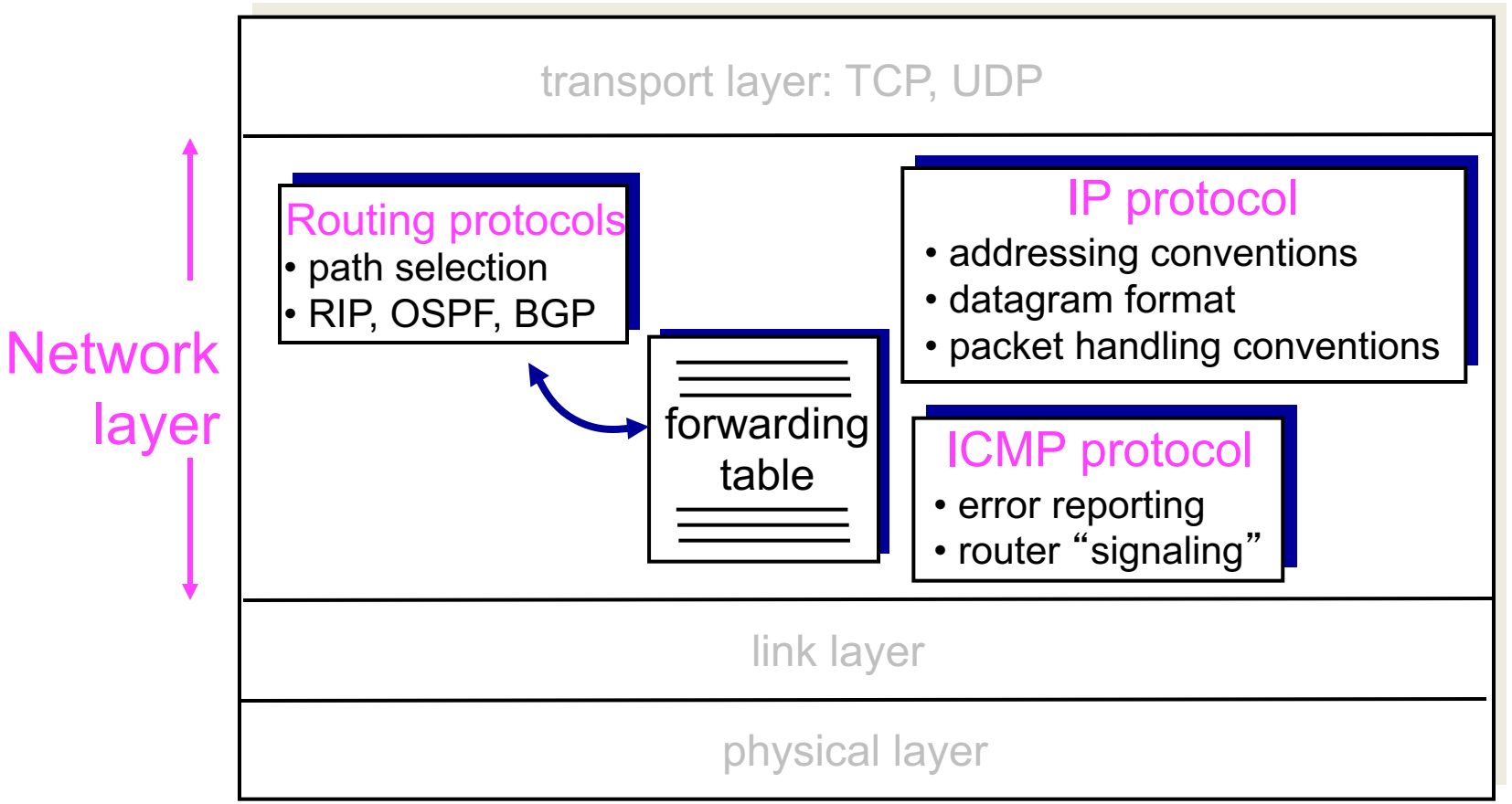
OVERVIEW

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

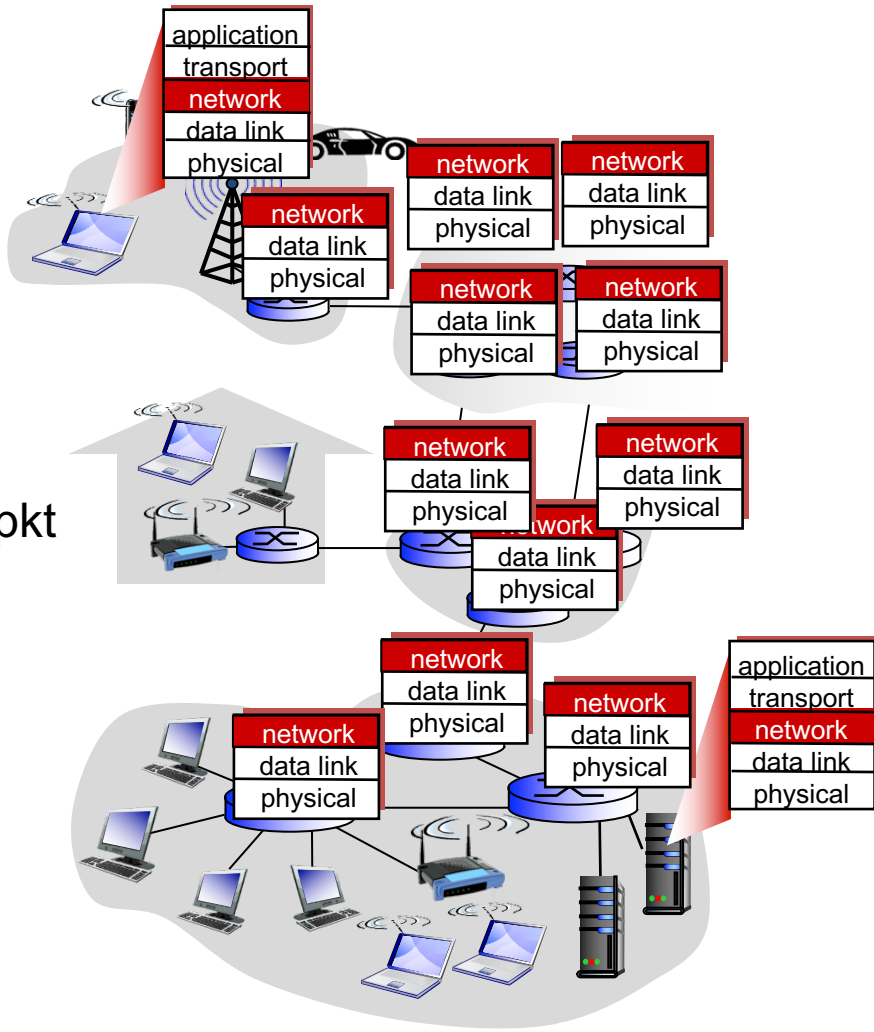
- move pkt from one host to another

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

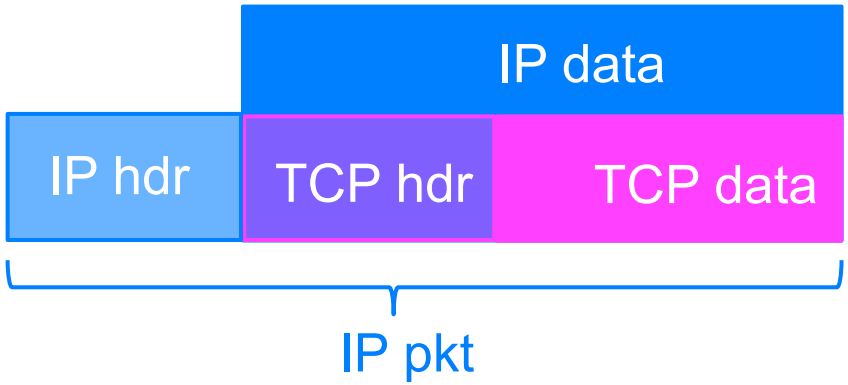
- decapsulates packets into segments, delivers to transport layer

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

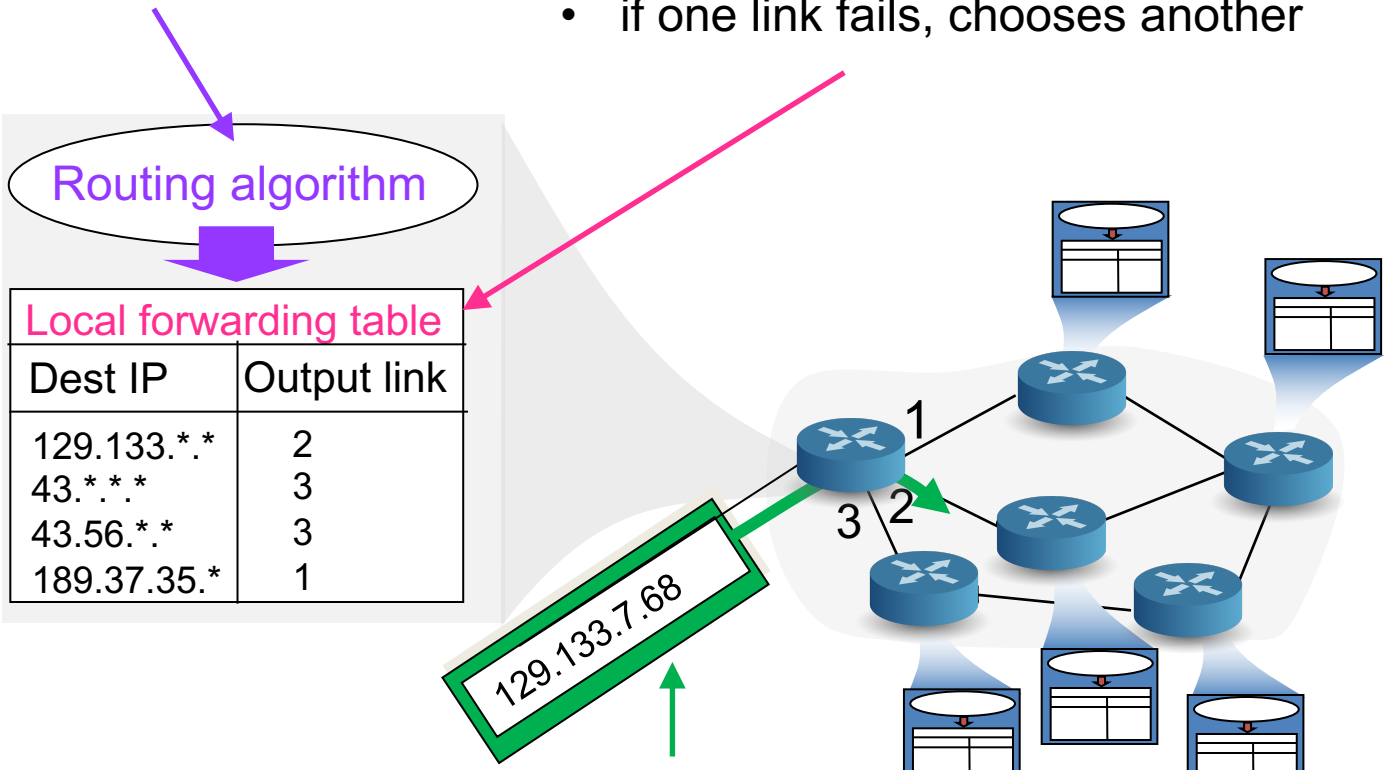
1. Control plane
   - comprises traffic only between routers, to compute routes between src and dst
   - network-wide: routers run routing algorithms

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

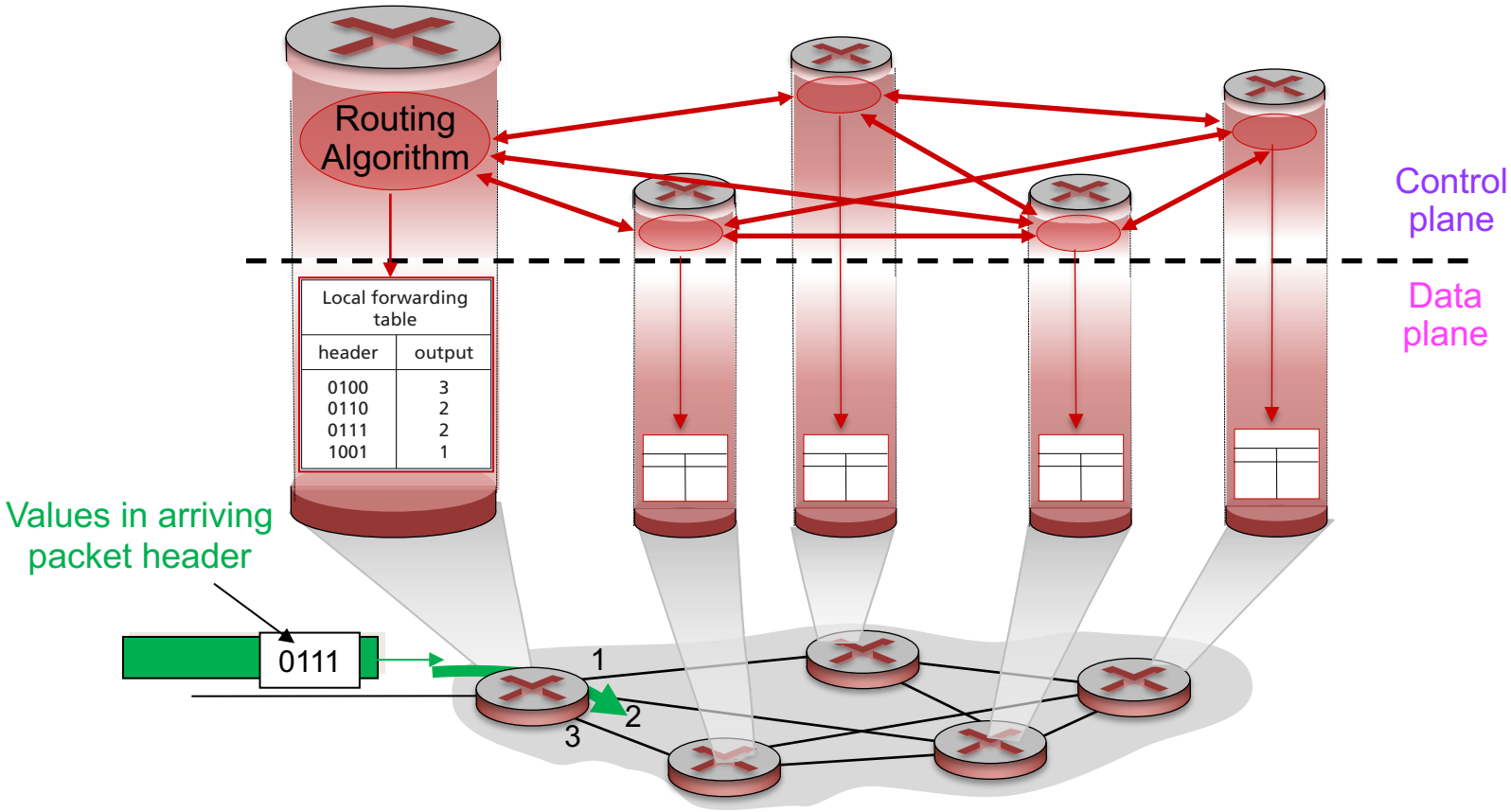
Forwarding (faster time scale)
- routers use paths to choose best output link for packet destination IP address
- if one link fails, chooses another



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

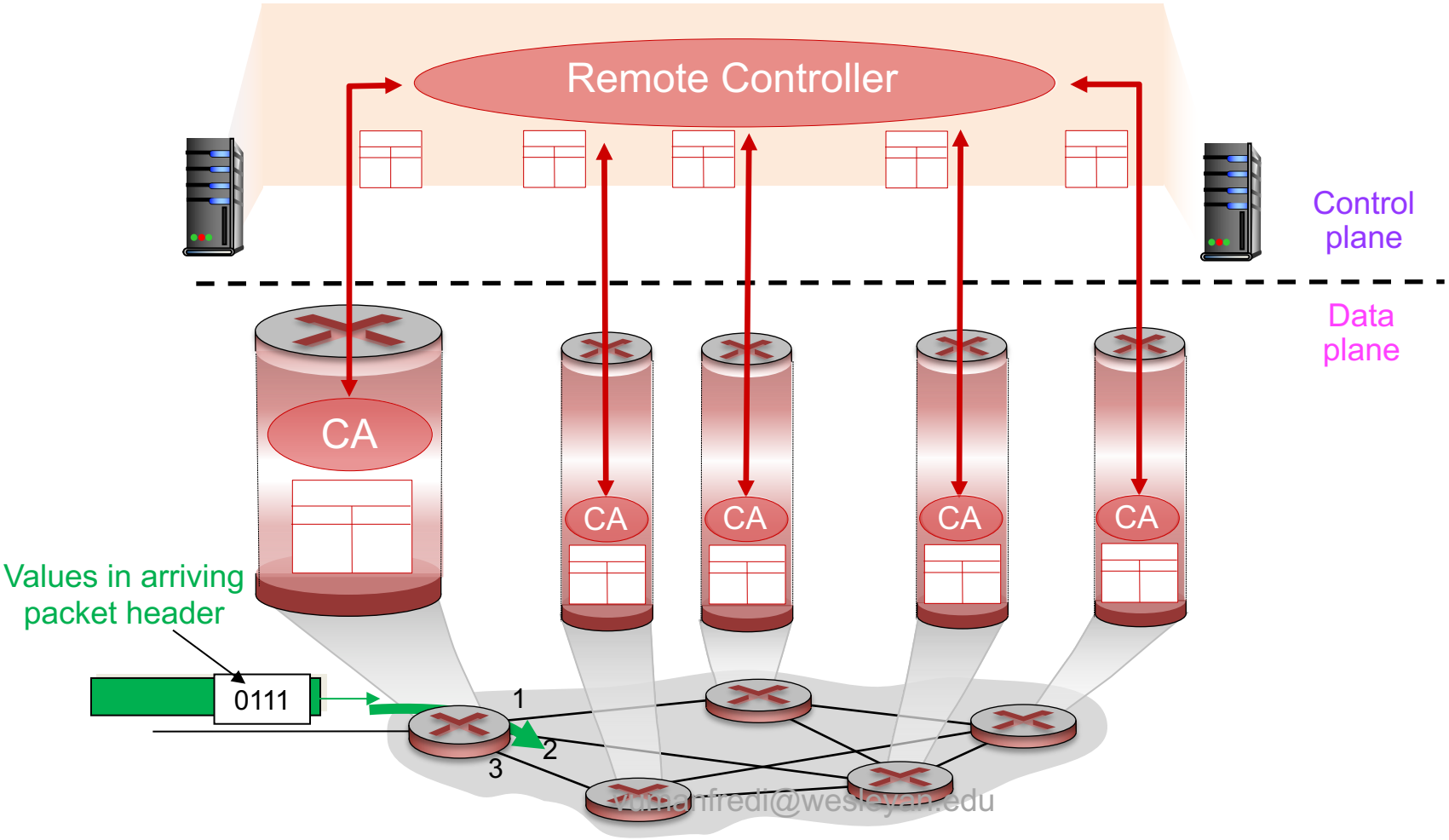
Approach 1: per-router control plane

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 ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
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

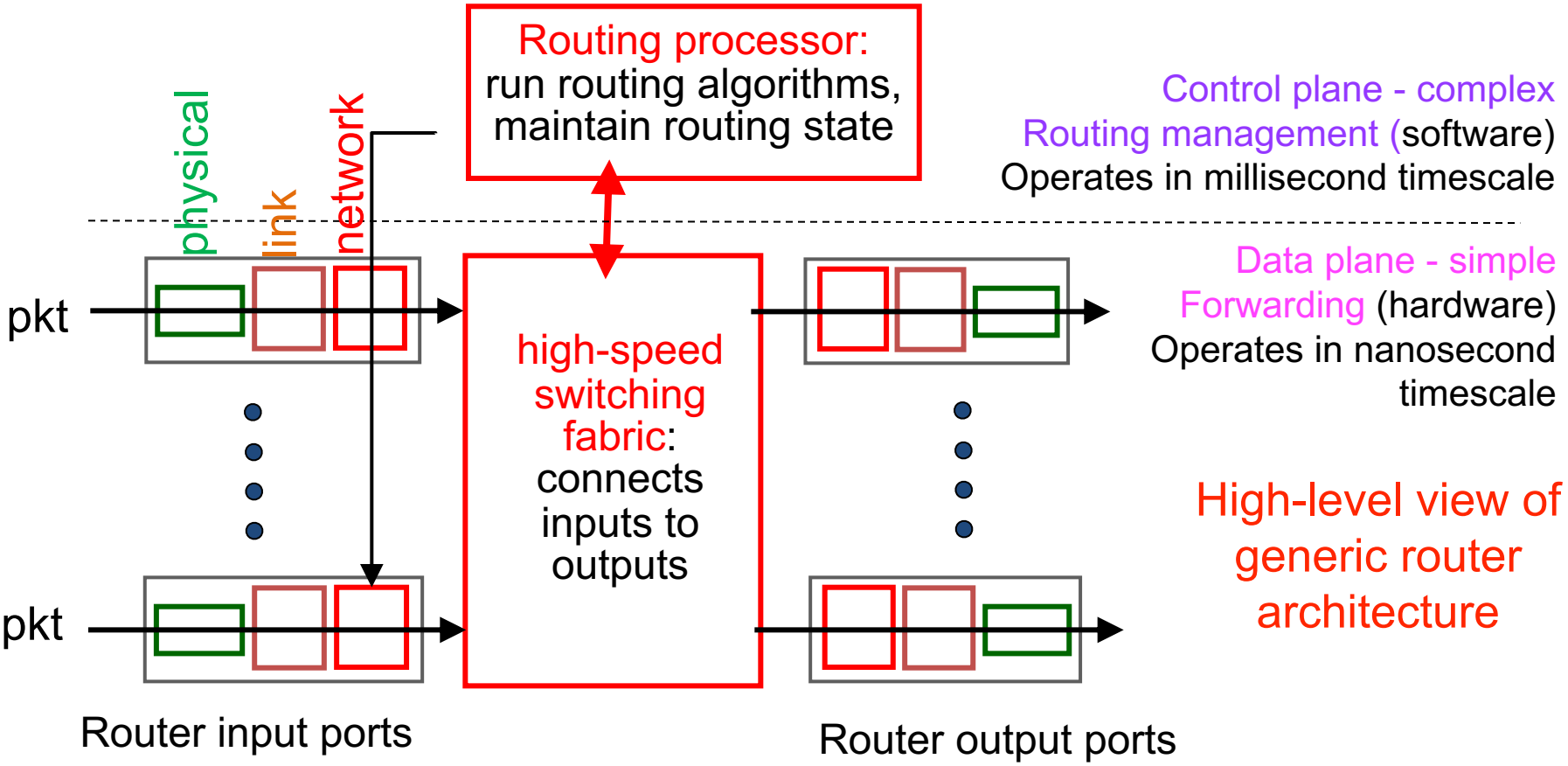
ATM: Asynchronous Transfer Mode
e.g., used in public switched telephone network

Network Layer

WHAT'S INSIDE A ROUTER?

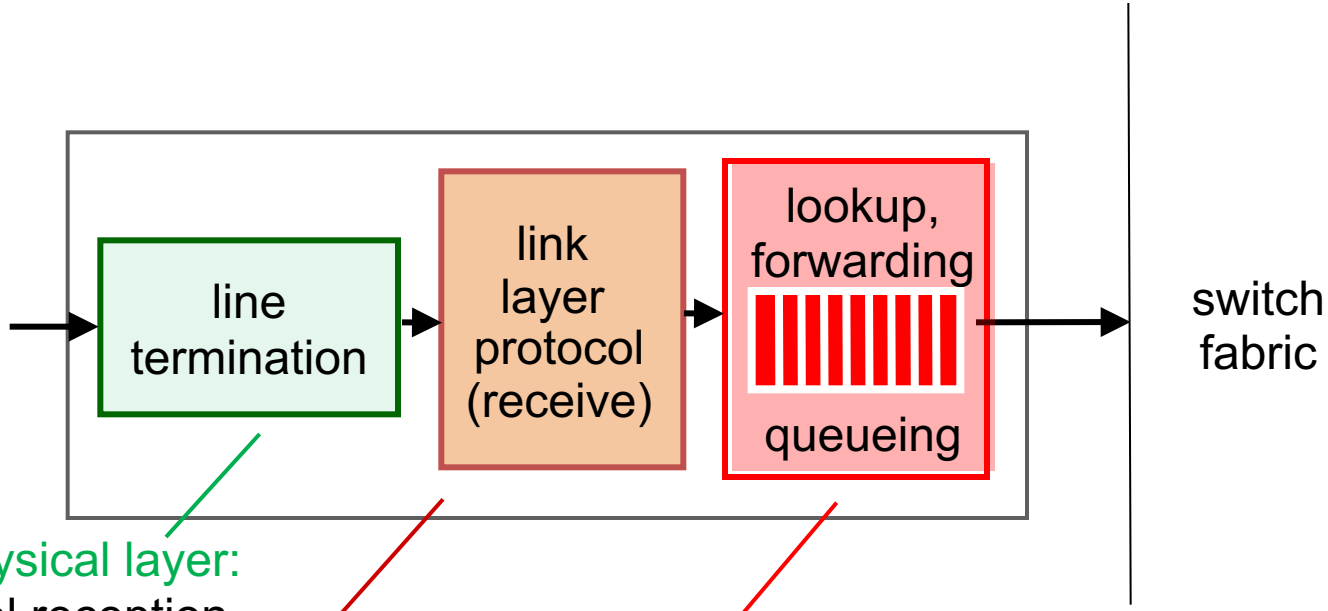
What does a router need to do?

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



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

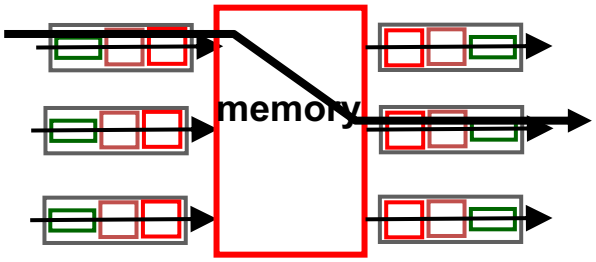
Transfer packet

- from input buffer to appropriate output buffer

Switching rate

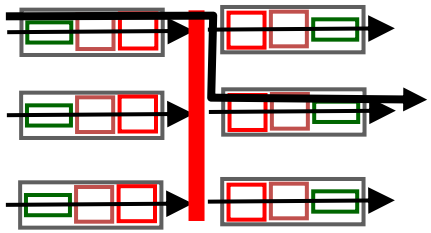
- rate at which packets can be transferred from inputs to outputs
- N inputs: switching rate = N x line rate desirable

3 types of switching fabrics



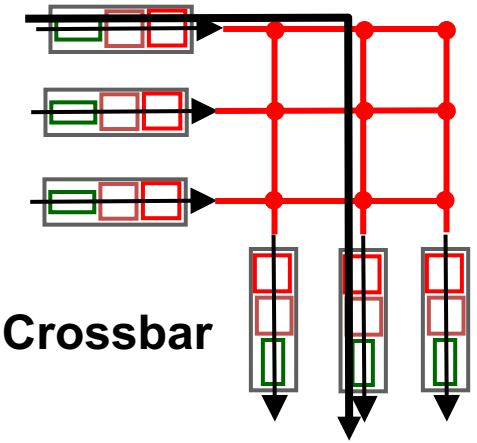
Memory

Speed limited by memory bandwidth



Bus

Speed limited by bus contention



Crossbar

Forward multiple pkts in parallel

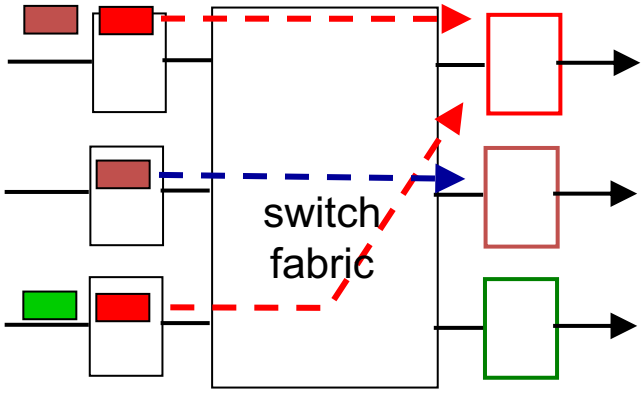
Contention at input ports

If switching fabric slower than input ports combined

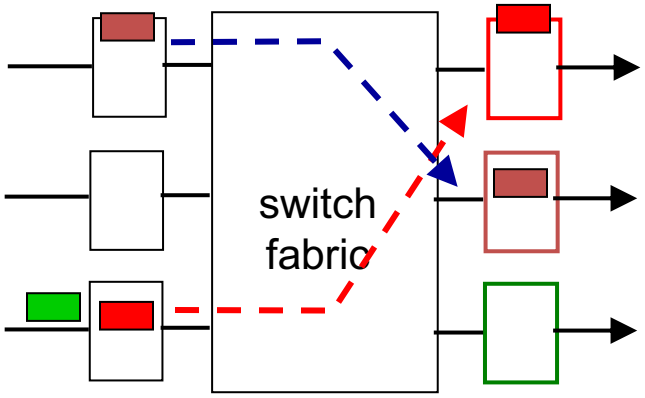
- queueing may occur at input queues
- queueing delay and loss due to input buffer overflow!

Head-of-the-Line (HOL) blocking

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

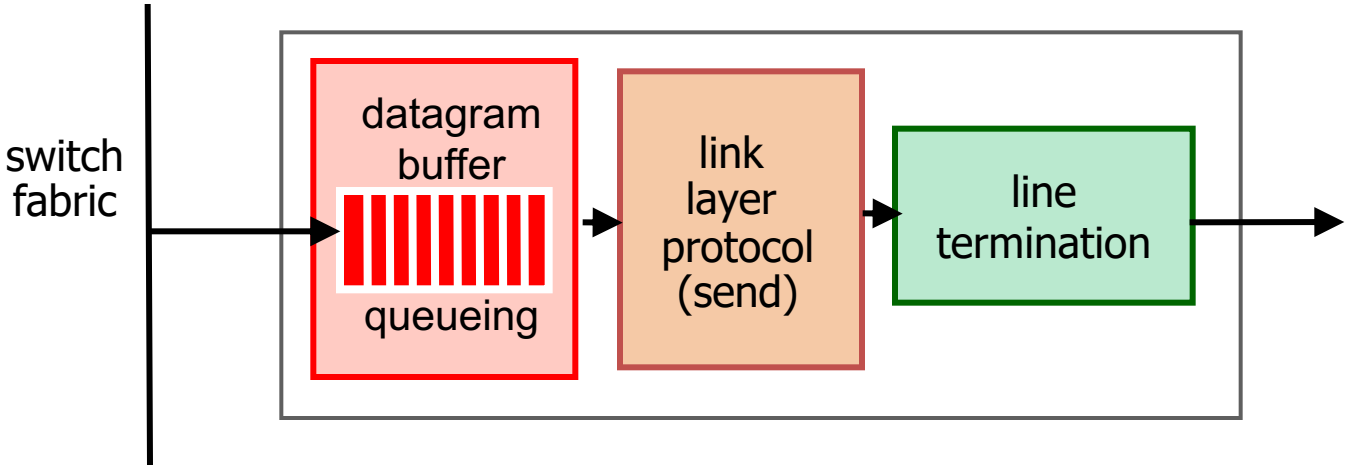


Output port contention: only one red packet can be transferred.
Lower red packet is blocked



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

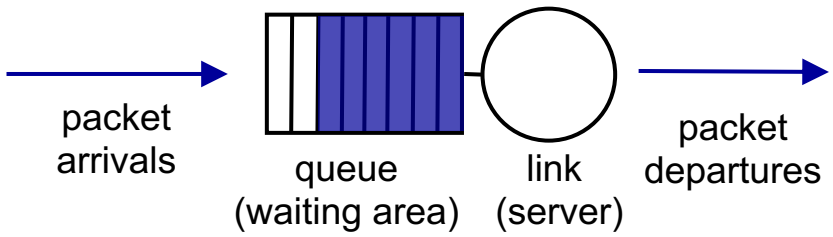
Scheduling

- 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

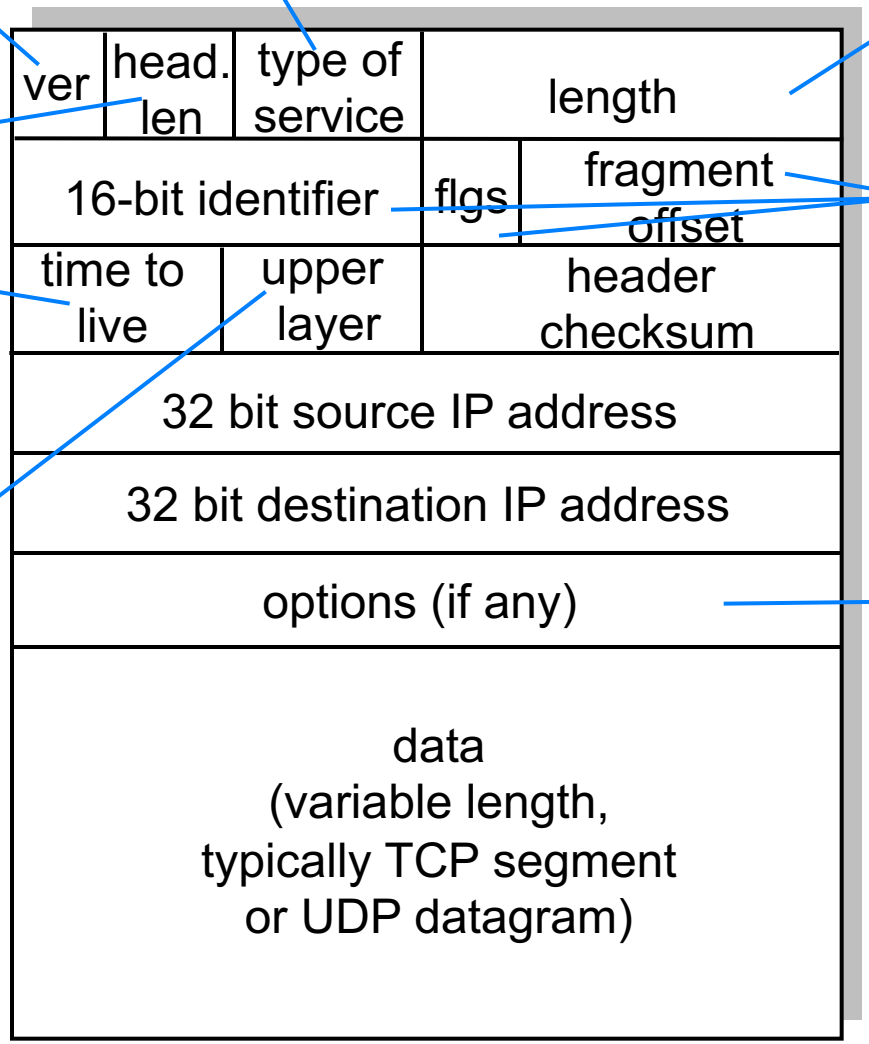
- best effort service
  - to get pkts from one end host to another across many interconnected networks using dst IP address in IP hdr
- addressing
  - format and usage of addresses
- fragmentation
  - e.g., if pkt size exceeds Ethernet MTU of 1500 bytes
- some error detection

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



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: IPv4

120	4.462069	192.168.0.14	TCP	17.248.202.1	52107 → 443 [ACK]
121	4.462512	17.248.202.1	TLSv1.2	192.168.0.14	Application Data

```
> Frame 120: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
> Ethernet II, Src: 88:66:5a:28:6e:b1 (88:66:5a:28:6e:b1), Dst: Motorola_f6:83:2b (38:80:df:f6:83:2b)
v Internet Protocol Version 4, Src: 192.168.0.14 (192.168.0.14), Dst: 17.248.202.1 (17.248.202.1)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0x0000 (0)
  > Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x9e14 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.0.14 (192.168.0.14)
    Destination: 17.248.202.1 (17.248.202.1)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
> Transmission Control Protocol, Src Port: 52107, Dst Port: 443, Seq: 1316034368, Ack: 813129735, Len: 0
```

Wireshark: IPv6

No.	Time	Source	Protocol	Destination	Info
149	6.686651	2001:558:feed:443::55	TCP	2601:181:4700:bc00:c…	443 → 581
150	6.687209	2001:558:feed:443::55	TCP	2601:181:4700:bc00:c…	443 → 581
151	6.687854	2001:558:feed:443::55	TLSv1.2	2601:181:4700:bc00:c…	Applicat

```
> Frame 150: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
> Ethernet II, Src: Motorola_f6:83:2b (38:80:df:f6:83:2b), Dst: 88:66:5a:28:6e:b1 (88:66:5a:28:6e:b1)
v Internet Protocol Version 6, Src: 2001:558:feed:443::55 (2001:558:feed:443::55), Dst: 2601:181:4700:bc00:cc5e:2f71:a04a:b698 (26
    0110 .... = Version: 6
  > .... 0000 0001 .... .... .... .... .... = Traffic Class: 0x01 (DSCP: CS0, ECN: ECT(1))
    .... .... .... 0000 0000 0000 0000 0000 = Flow Label: 0x00000
    Payload Length: 32
    Next Header: TCP (6)
    Hop Limit: 51
    Source: 2001:558:feed:443::55 (2001:558:feed:443::55)
    Destination: 2601:181:4700:bc00:cc5e:2f71:a04a:b698 (2601:181:4700:bc00:cc5e:2f71:a04a:b698)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
> Transmission Control Protocol, Src Port: 443, Dst Port: 58110, Seq: 2343448060, Ack: 2003653776, Len: 0
```

Wireshark

Look at IP headers and ping/traceroute

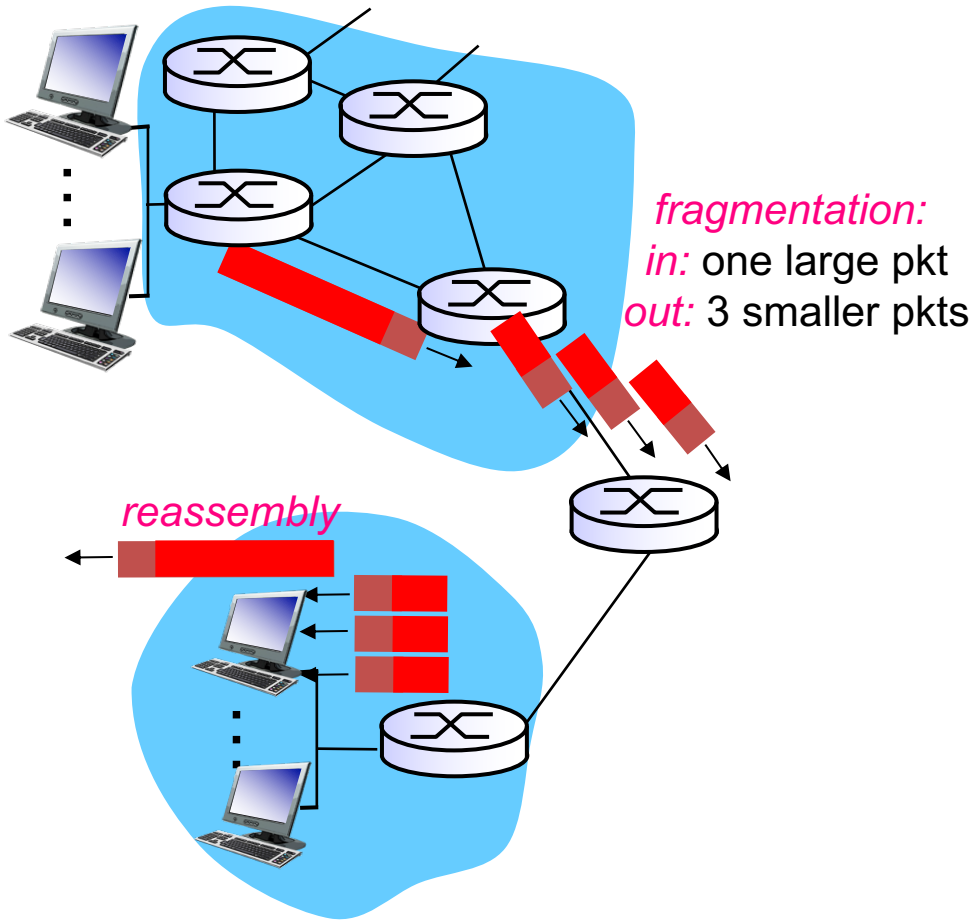
IP fragmentation and reassembly

Network links have MTU

- largest possible link-level frame
- different link types have different MTUs

Fragment when pkt > MTU

- 1 pkt becomes several pkts
  - IP header bits used to identify and order related fragments
- reassembled only at final dst
- re-fragmentation possible
- don't recover from lost fragments
- (IPv6 does not support)



DoS attack: send fragmented pkts but leave one out

IP fragmentation and reassembly

4000 byte packet
- 3980 bytes payload
- IP hdr >=20 bytes

MTU = 1500 bytes