CS-340 Introduction to Computer Networking

Lecture 8: IPv4 Addressing

Steve Tarzia

Many diagrams & slides are adapted from those by J.F Kurose and K.W. Ross

Last Lecture: TCP Congestion Control

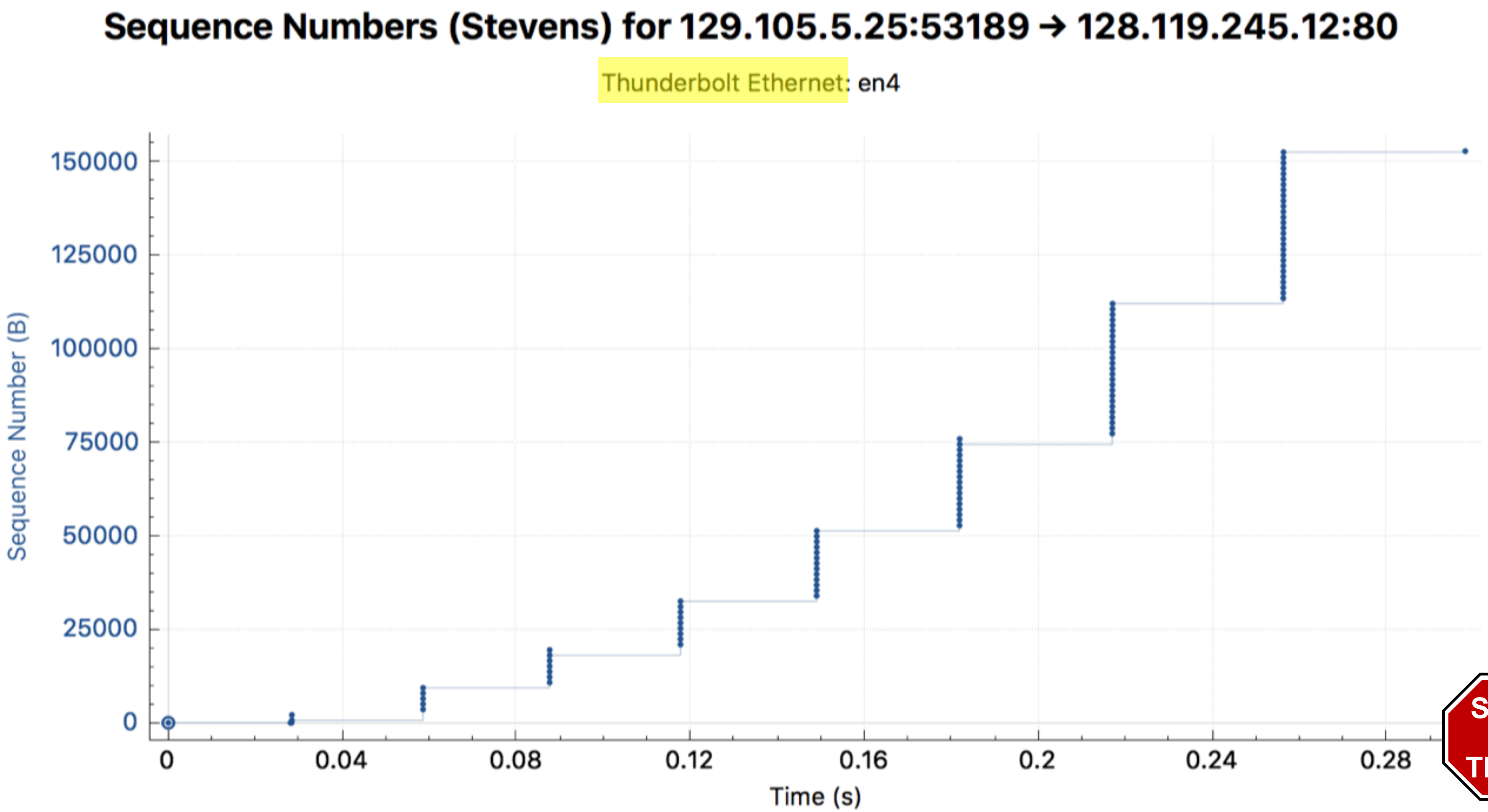
- **Congestion control** is implemented with a dynamic *congestion window*, controlled by *heuristics*. **Reno** congestion control operate in phases:
  - *Slow start* – exponential growth to find approximate network capacity.
  - *Congestion avoidance* – linear growth, slowly trying to increase throughput.
  - *Fast recovery* – If one packet is lost, then cut window in half.
- **Additive Increase, Multiplicative Decrease** ensures fair sharing.
- **CUBIC** and **BBR** are newer alternatives that work better in fast networks, particularly long fat pipes. (Read this for more info.)
- TCP behavior can be controlled with *socket options*:
  - Nagle's algorithm merges small packets to reduce *header overhead*.
  - TCP *keepalive* message can be periodically sent.

Congestion Control

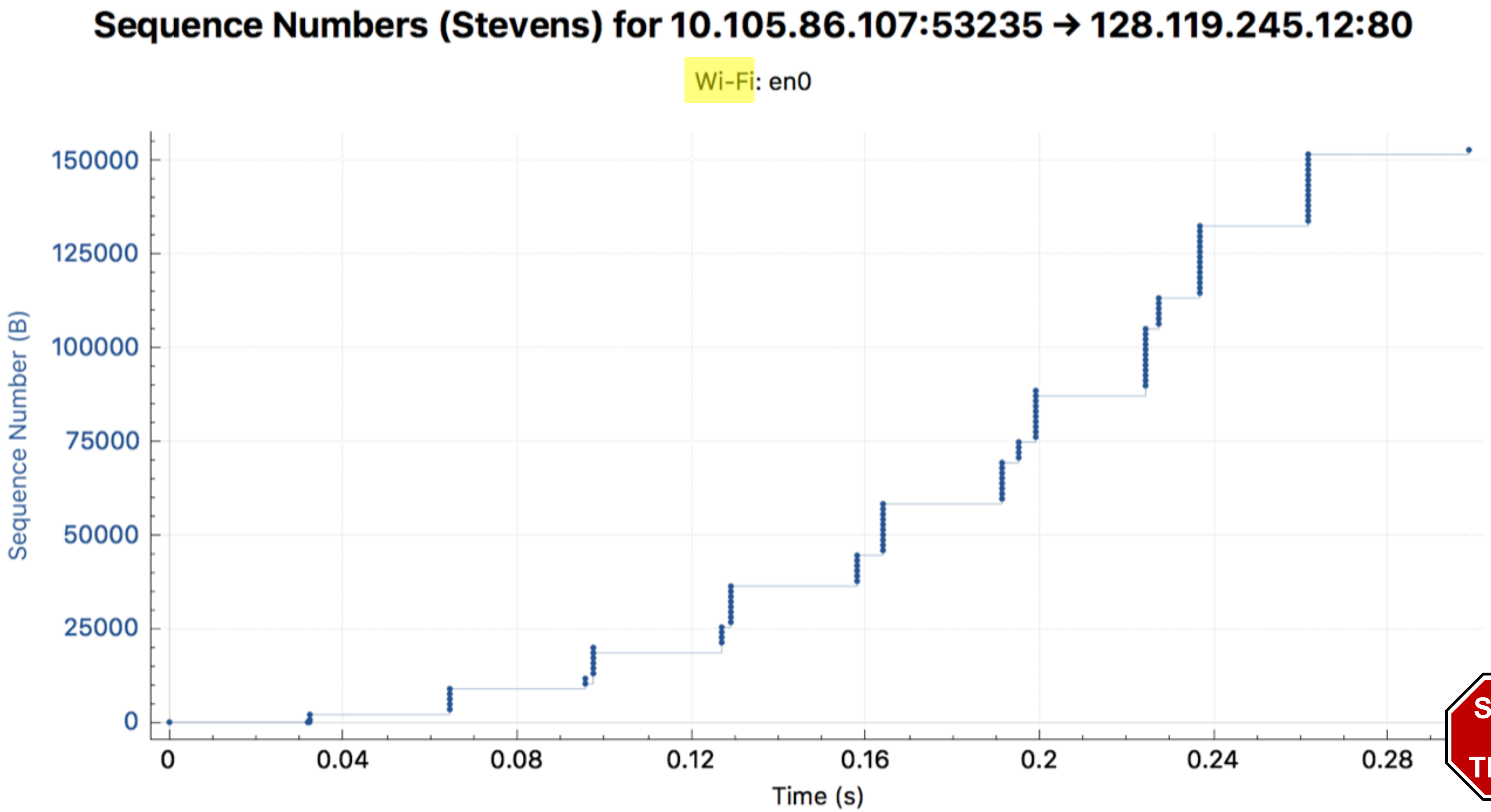
- We covered *TCP Reno* congestion control in detail.
- Many alternative algorithms have been proposed and implemented.
- Most real TCP implementations are similar to Reno, but not identical.

Variant	Feedback	Required changes	Benefits	Fairness
(New)Reno	Loss	-	-	Delay
Vegas	Delay	Sender	Less loss	Proportional
High Speed	Loss	Sender	High bandwidth	
BIC	Loss	Sender	High bandwidth	
CUBIC	Loss	Sender	High bandwidth	
H-TCP	Loss	Sender	High bandwidth	
FAST	Delay	Sender	High bandwidth	Proportional
Compound TCP	Loss/Delay	Sender	High bandwidth	Proportional
Westwood	Loss/Delay	Sender	L	
Jersey	Loss/Delay	Sender	L	
BBR[11]	Delay	Sender	BLVC, Bufferbloat	
CLAMP	Multi-bit signal	Receiver, Routers	V	Max-min
TFRC	Loss	Sender, Receiver	No Retransmission	Minimum delay
XCP	Multi-bit signal	Sender, Receiver, Router	BLFC	Max-min
VCP	2-bit signal	Sender, Receiver, Router	BLF	Proportional
MaxNet	Multi-bit signal	Sender, Receiver, Router	BLFSC	Max-min
JetMax	Multi-bit signal	Sender, Receiver, Router	High bandwidth	Max-min
RED	Loss	Router	Smaller delay	
ECN	Single-bit signal	Sender, Receiver, Router	Less loss	

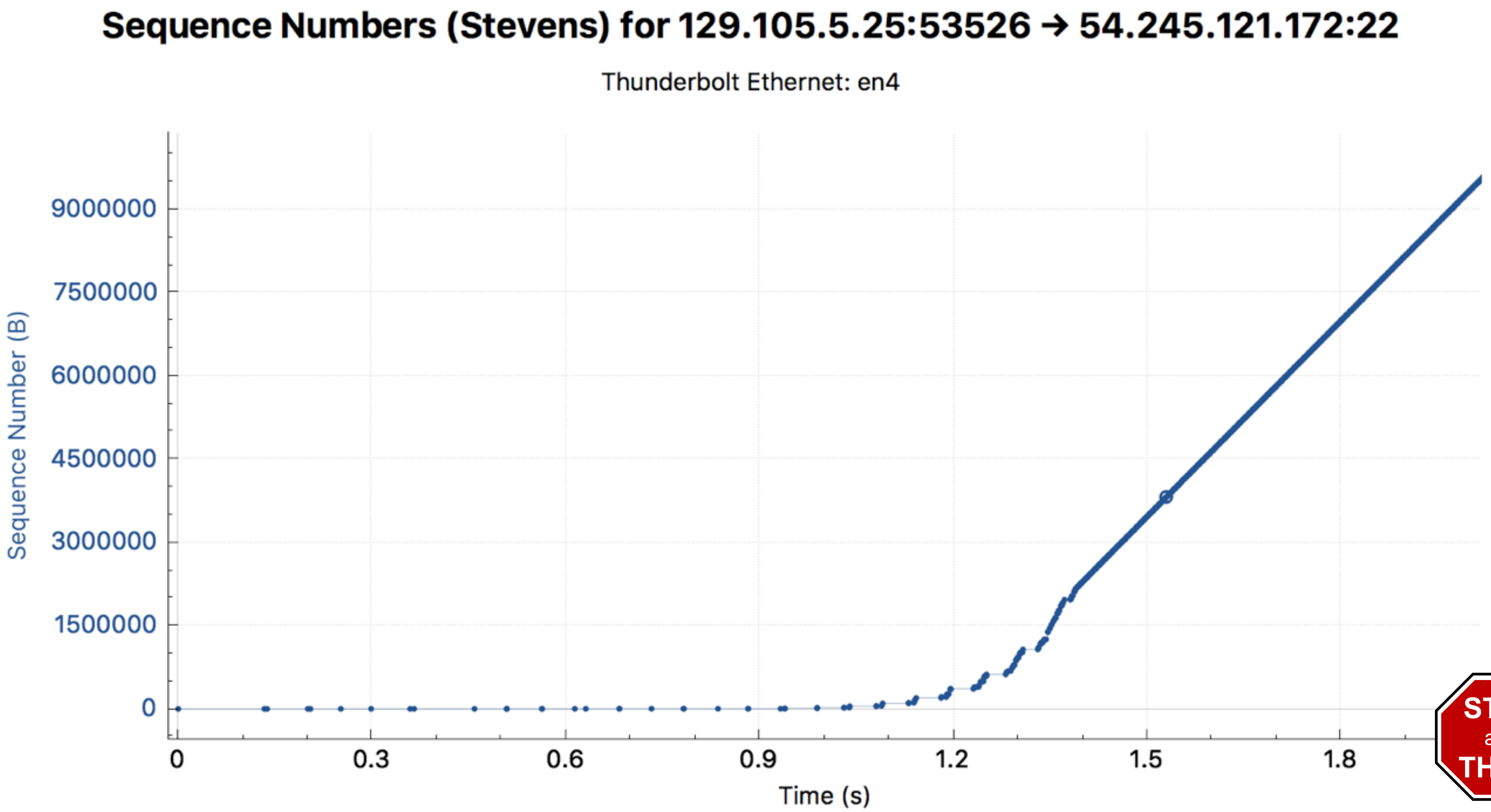
TCP in practice: HTTP POST a small file on Ethernet



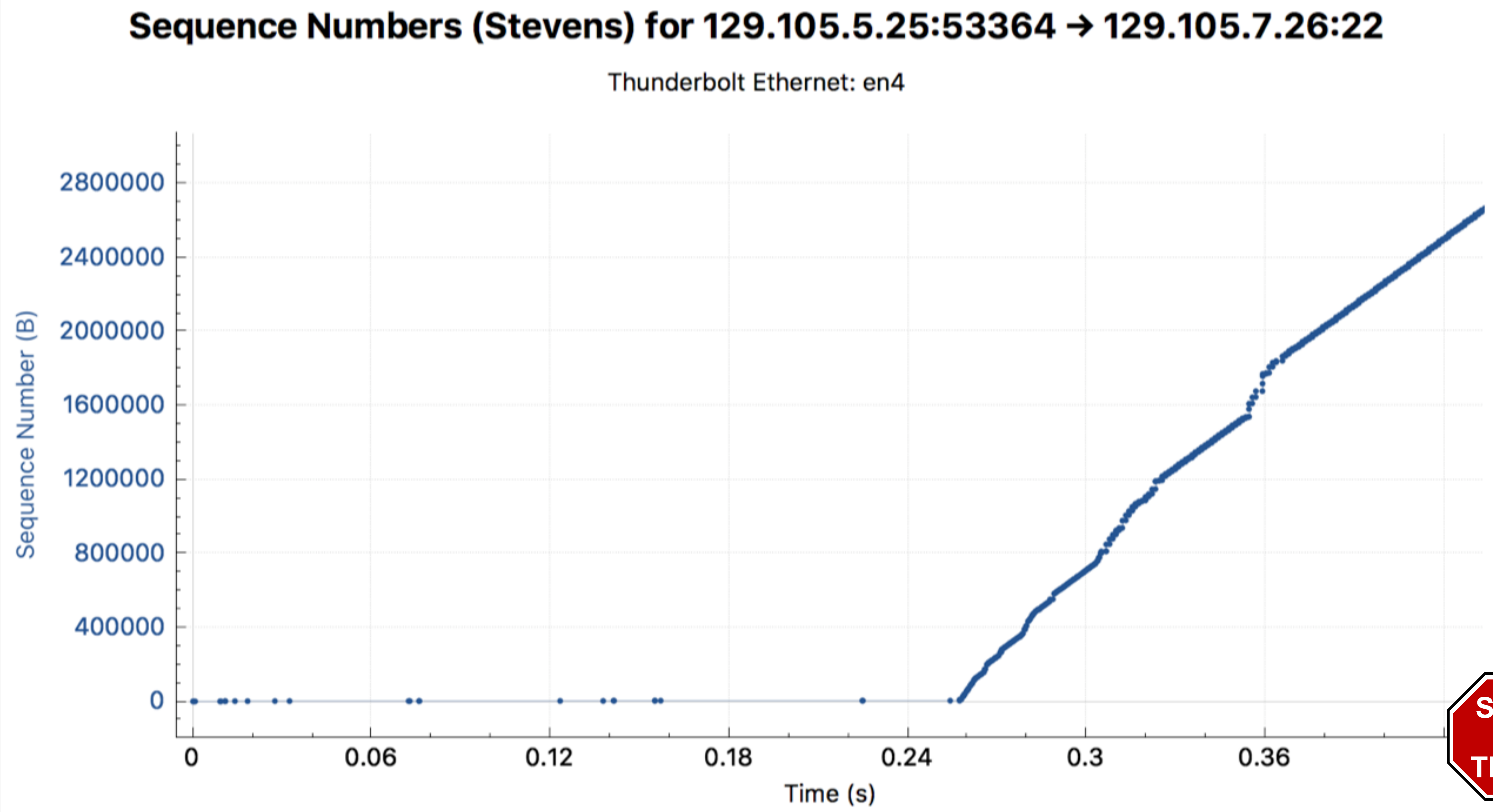
TCP in practice: HTTP POST a small file on WiFi



TCP in practice: **SCP** a large file to Seattle

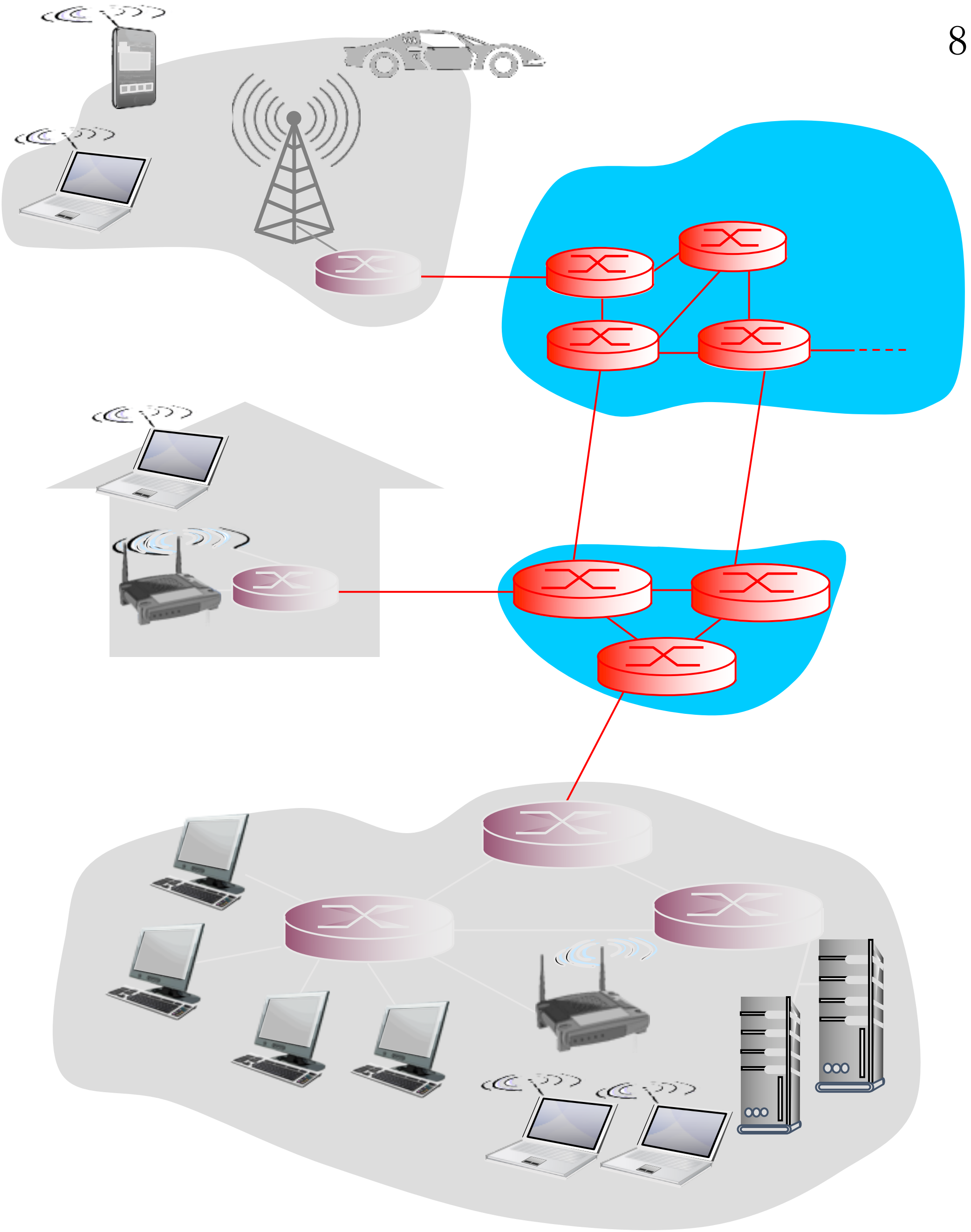


TCP in practice: SCP a large file **on campus**

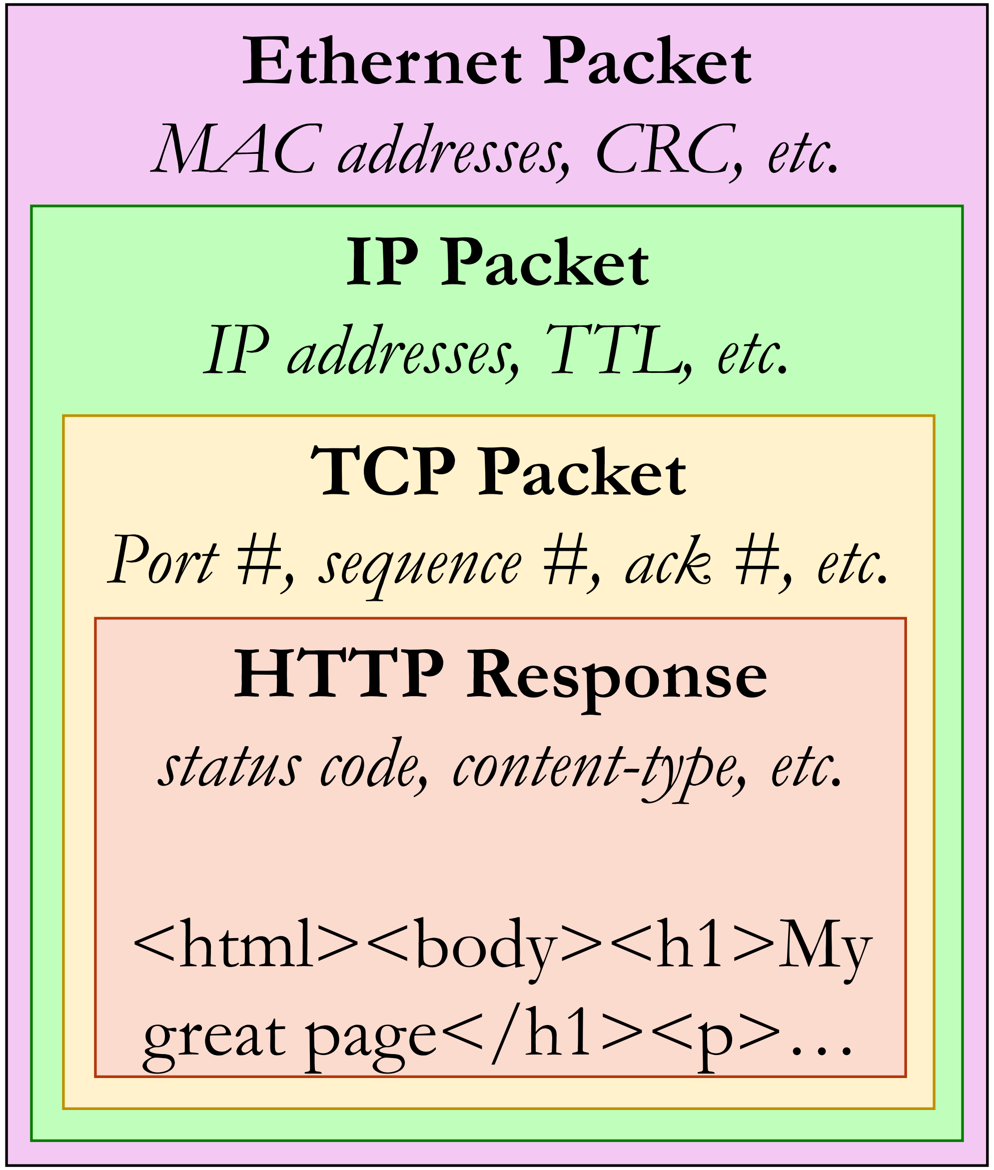


Next topic: IP/Forwarding

a.k.a "The Data Plane" of the Network Layer.
Covered in Chapter 4 of Kurose & Ross.

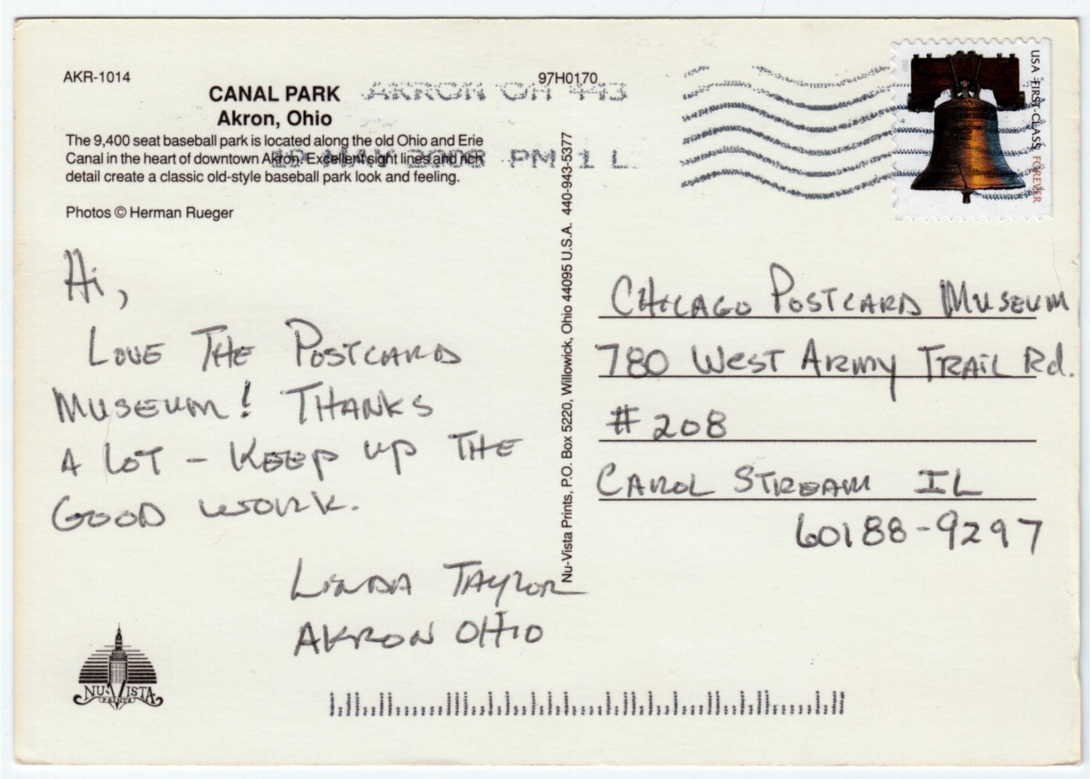


Each layer solves a particular set of problems



- Link layer: shares a physical channel among several transmitters/receivers
- **Network layer**: routes from source to destination, along many hops.
- Transport layer:
  - Multiplexing (>1 connection / machine)
  - Ordering, • Acknowledgement, • Pacing
- HTTP layer:
  - Resource urls, • Response codes,
  - Caching, • Content-types, • Compression

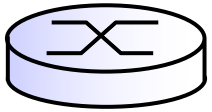
Internet Protocol (IP) delivers packets (datagrams)

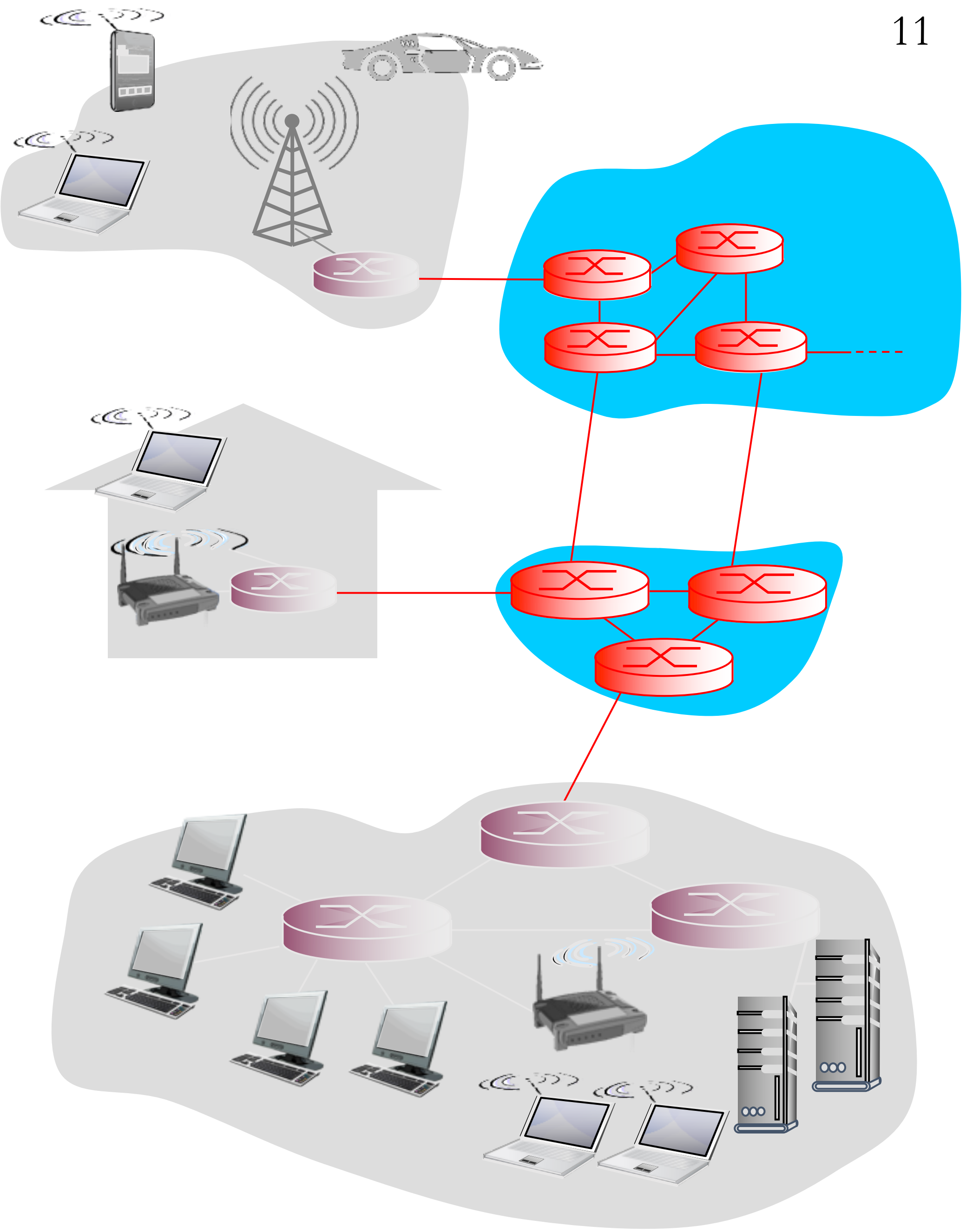


- Each datagram has receiver and sender ***IP addresses***.
- Maximum Transmission Unit (MTU) limits datagram size.
- *Best effort delivery* (may drop packets)

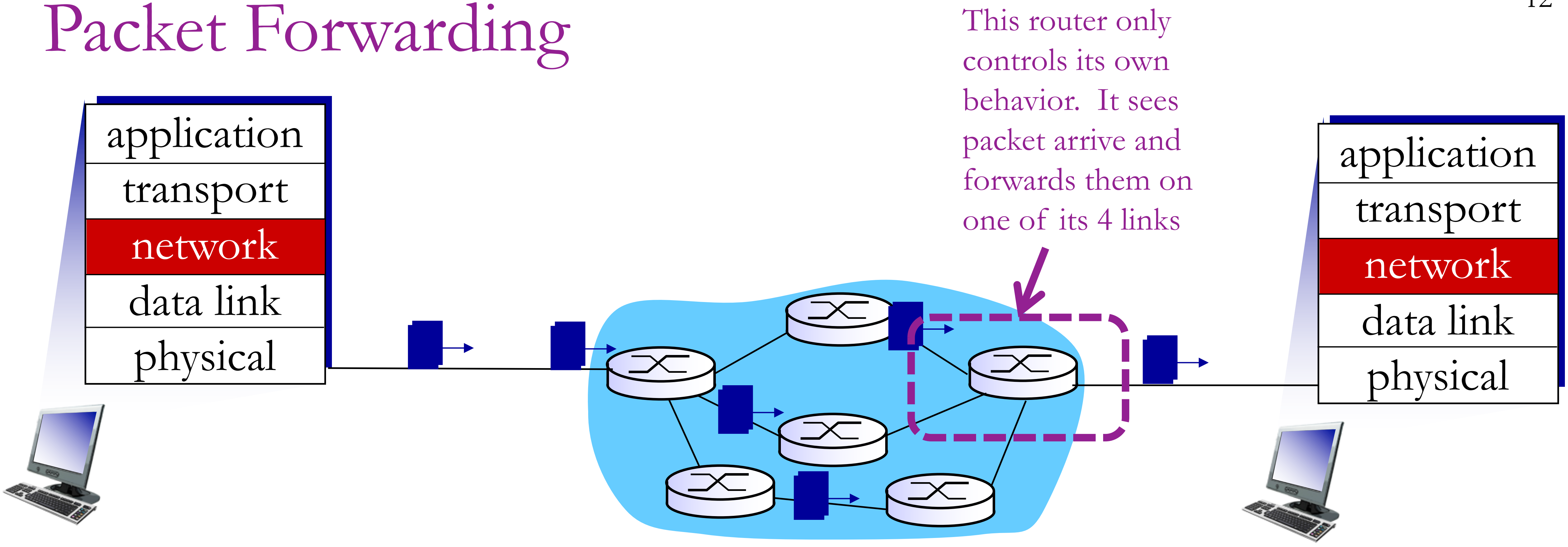
But how do packets reach their intended destination?

Routing

- ***Routing*** is the process of guiding packets to their destination.
  - Done by ***routers***  in the network core.
- Operates in two parts:
  - ***Control plane*** monitors traffic over the long-term and coordinates with other routers to **decide** forwarding rules. (Chapter 5)
  - ***Data plane*** **implements** the forwarding rules for each arriving packet. (Chapter 4)

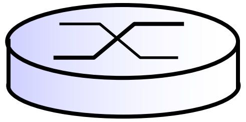


Packet Forwarding



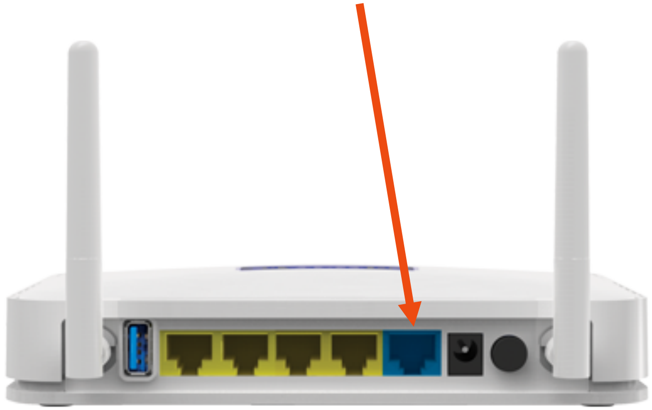
- Each router in network core is ignorant of end-to-end behavior.
  - Does not establish connections or reserve capacity (unlike circuit-switching)
- Just forwards packets in its neighborhood – *an elegant design*

What is a router?

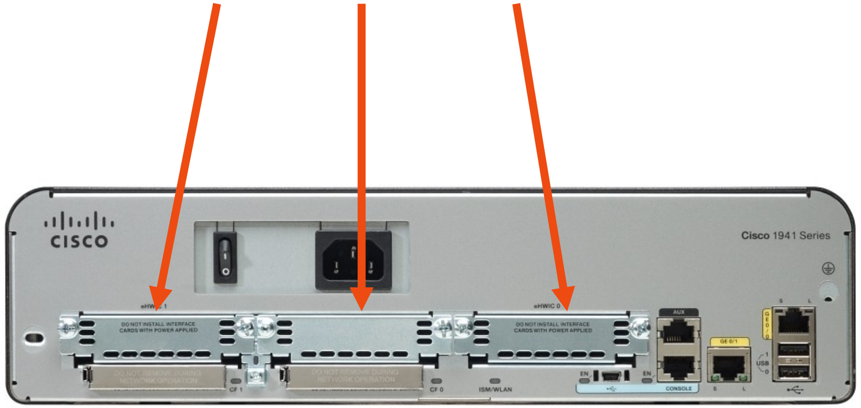


- A device with multiple network connections that is dedicated to forwarding packets (and deciding forwarding rules).

An *edge* router has just one connection to the Internet



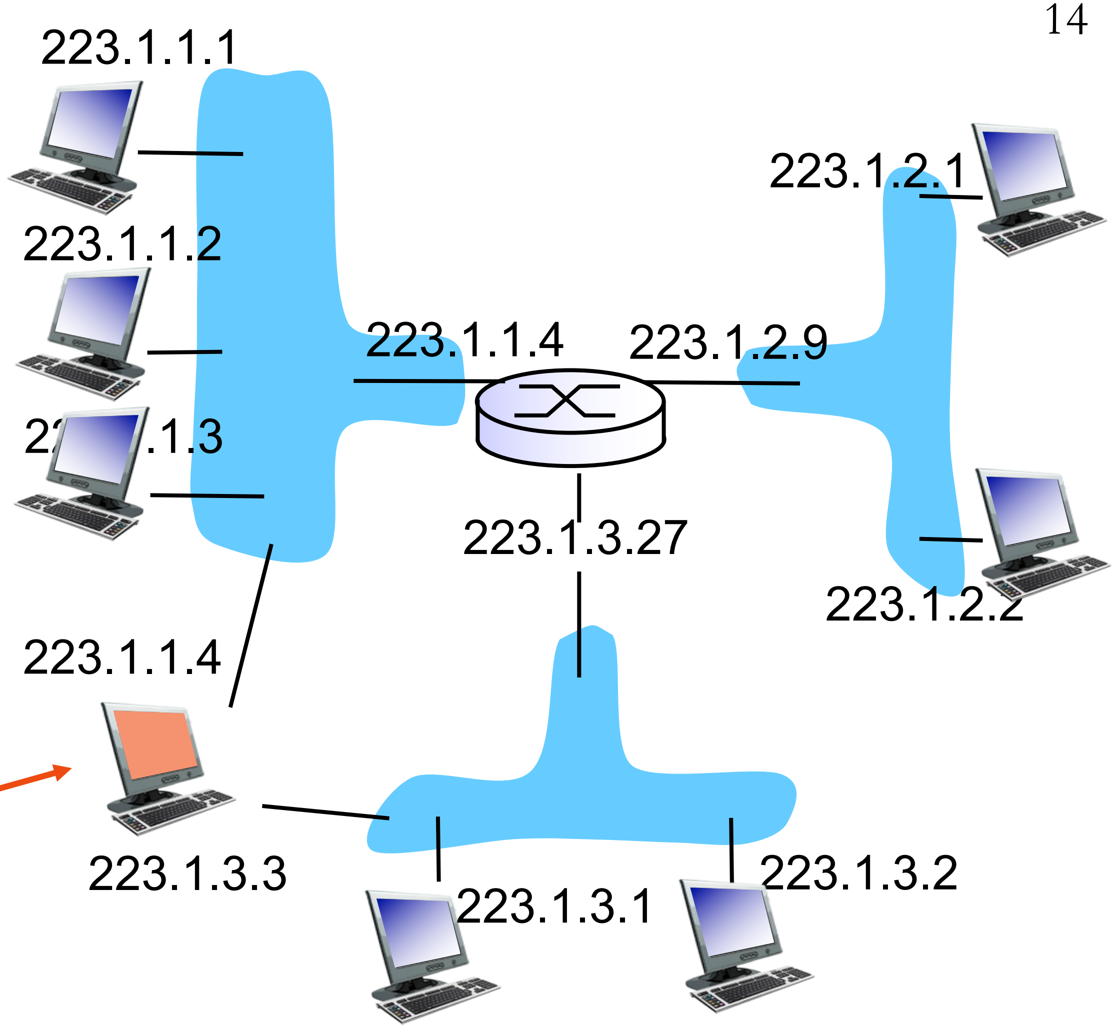
Routers usually are expandable to connect different types of physical media/cables (cat5, fiber, etc.)



Core routers can handle up to 160 TBps

IPv4 addressing

- *Interface* is a connection between host/router and a link.
- Every interface has an *IP address* (32 bits long in IP version 4)
- Routers usually have multiple interfaces (for multiple links)
- Hosts usually have one or two interfaces (WiFi + wired).



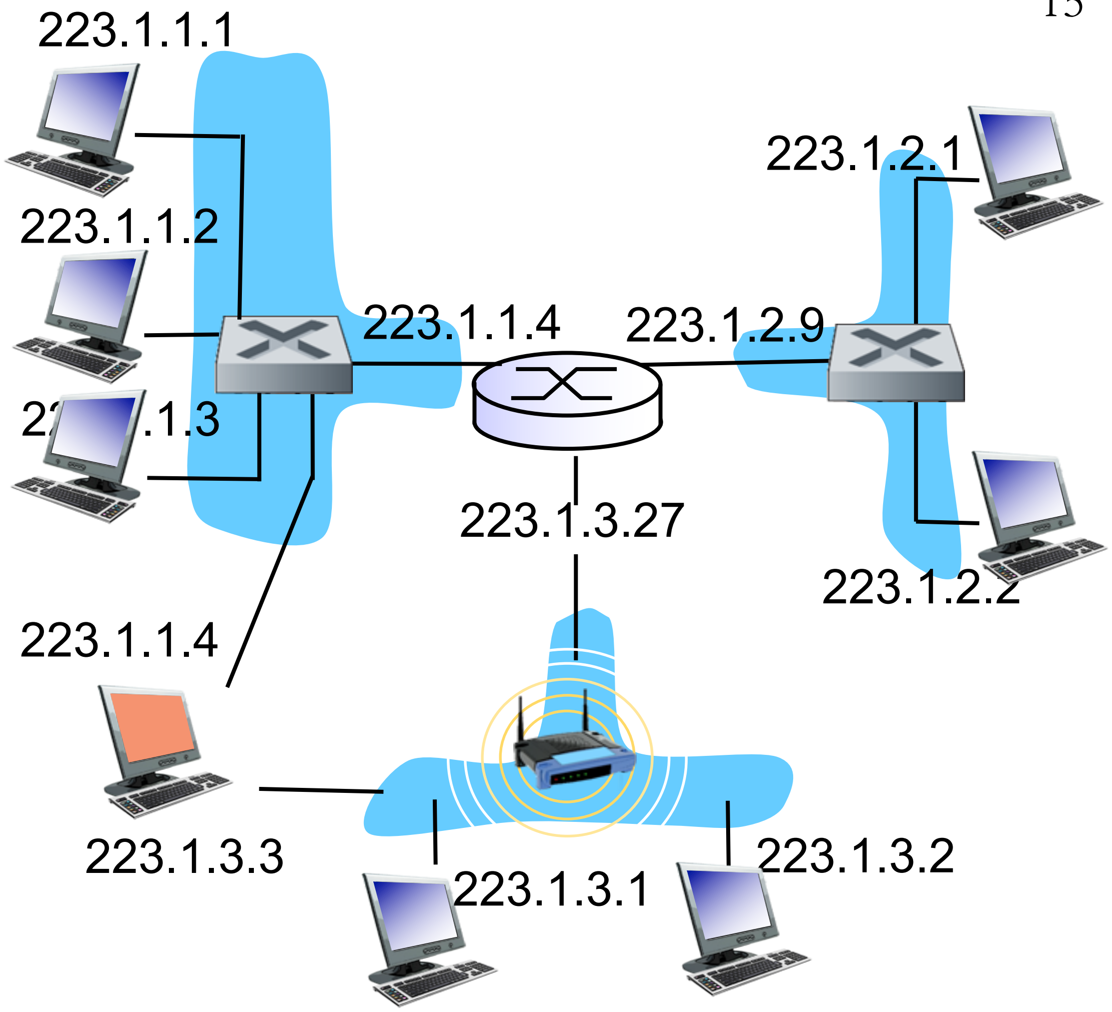
Dotted quad decimal notation is used for IPv4 addresss →

223.1.1.1 = 11011111 00000001 00000001 00000001

11011111	00000001	00000001	00000001
223	1	1	1

Interface connections

- We'll cover this later when discussing Link and Physical layers.
- For now, assume that multiple interfaces can somehow be connected.



cables connect two interfaces

switches connect multiple wired ethernet interfaces

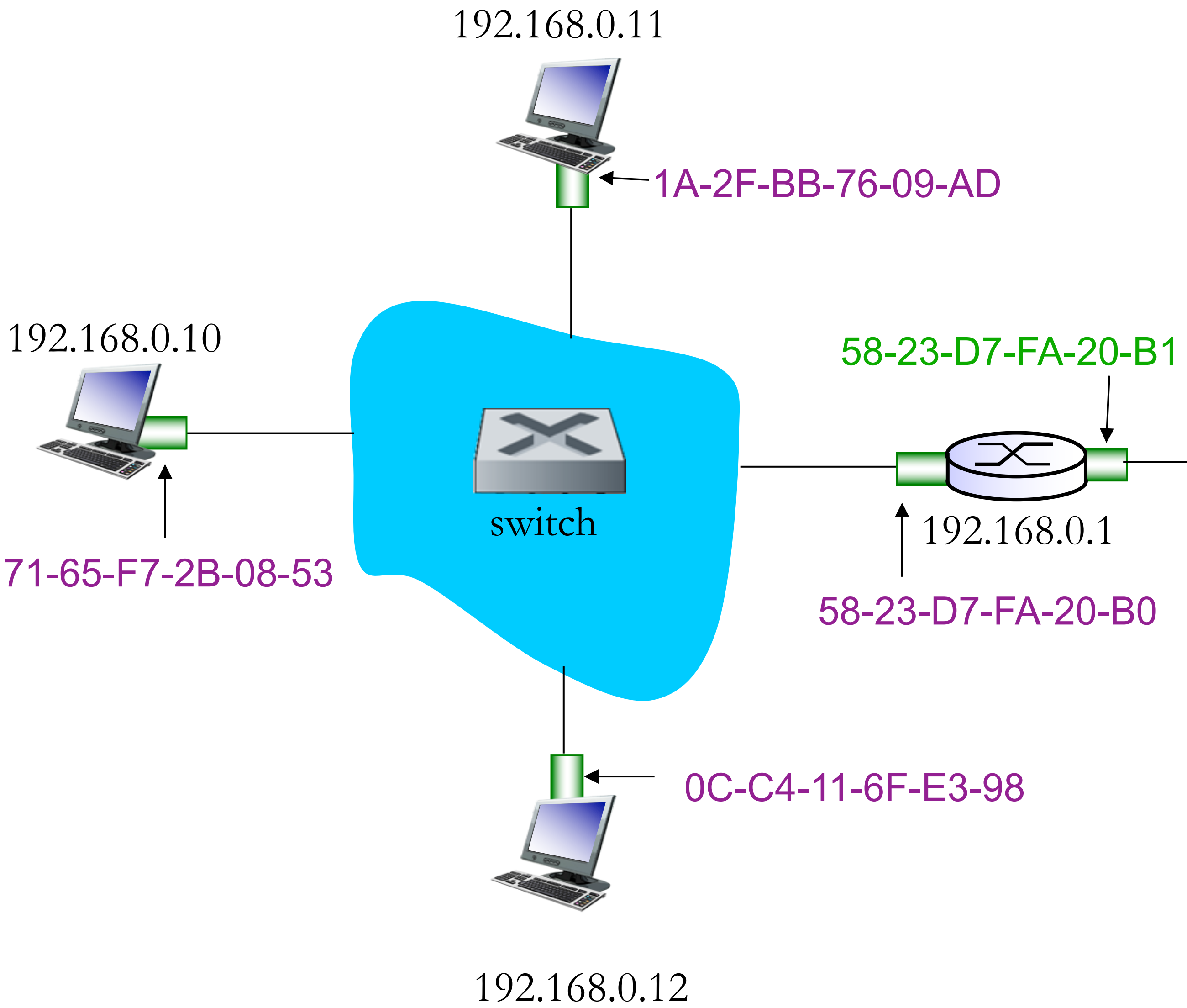
base stations connect multiple wireless interfaces.

Operate *below* IP layer

Preview of MAC addresses

- Previously, we assumed that links were *point-to-point*, but…
- Links are often shared channels.
  - Radio is an obvious example: When you broadcast, anyone nearby can hear.
  - Ethernet also uses shared channels called "subnets":
- MAC addresses are factory-coded addresses that identify the different interfaces on a shared communication channel.
- More details are coming later in Chapter 6 on the Link Layer.

An ethernet subnet with four interconnected hosts/interfaces:



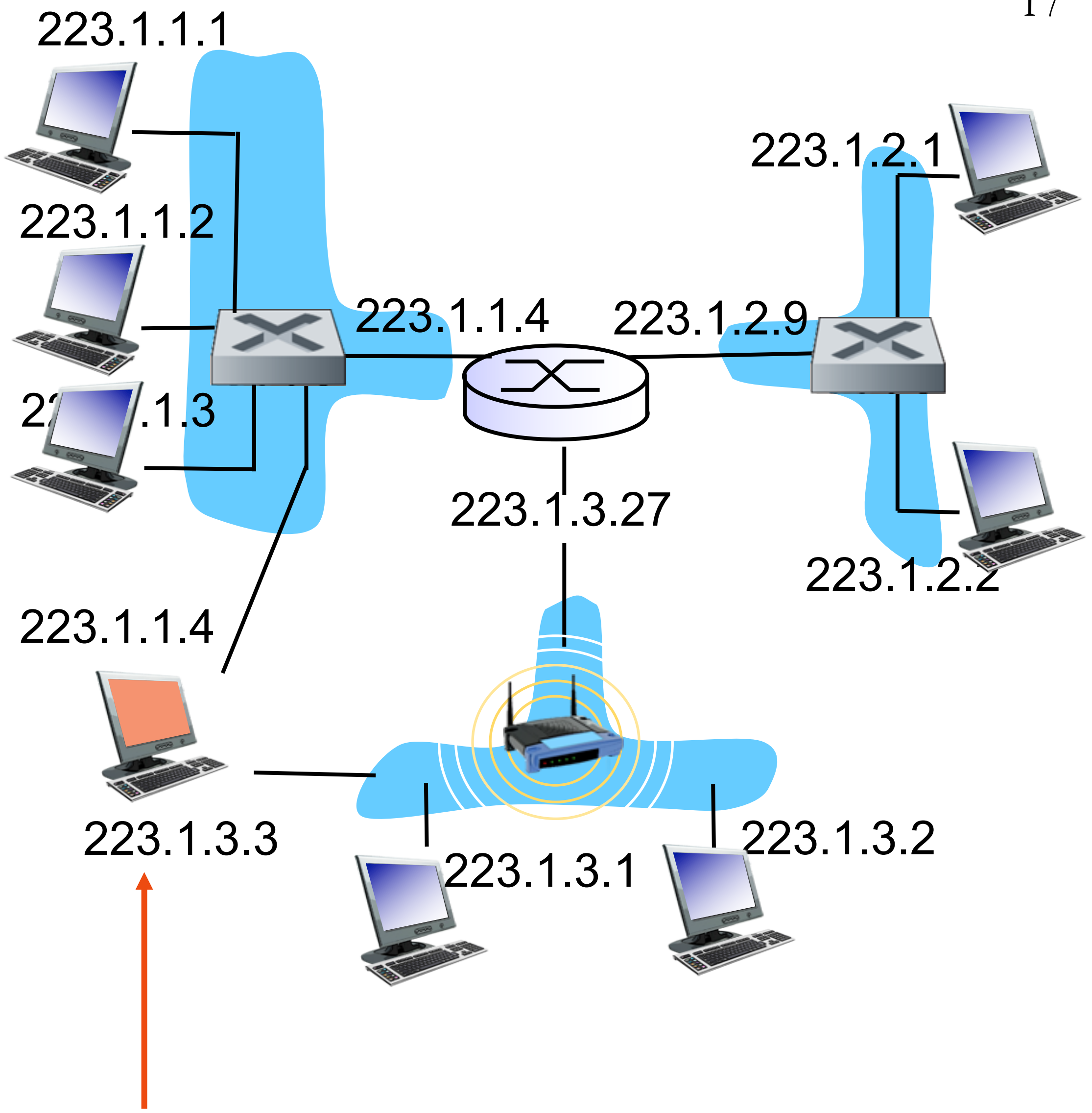
IP subnets

- Subnets are chunks of IP addresses in that can communicate *directly*, without traveling through a router.
  - Three subnets are shown →
- Routers send packets between subnets.
- ***Subnet mask*** indicates bit position where subnet is split:

`223.1.1.3/24`

/24 means that the high 24 bits identify the subnet, and remaining 8 low bits identify the host.

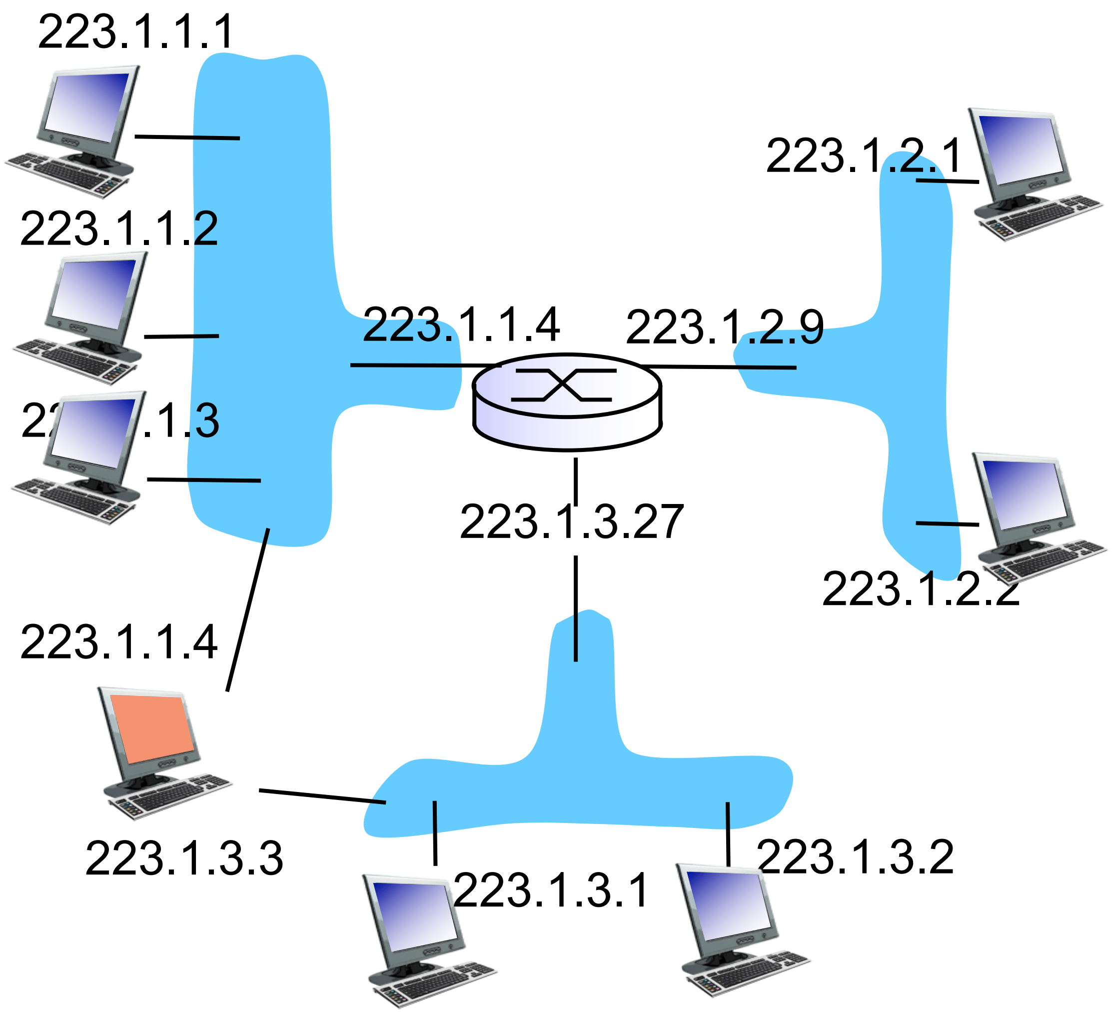
  - Subnet can also be expressed as a bitmask: 255.255.255.0 (equivalent to /24)



One machine on two different subnets

What are subnets used for?

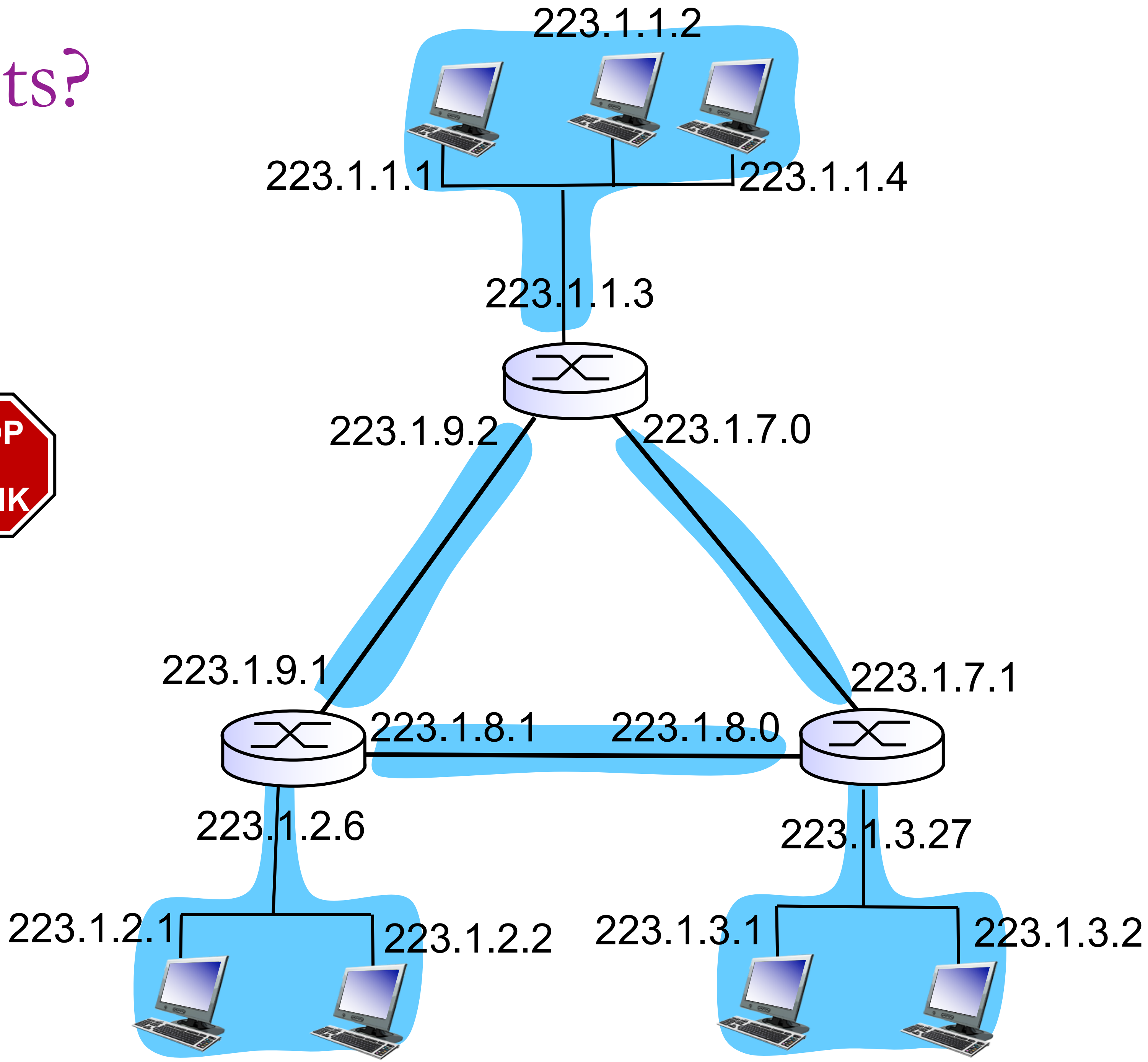
- Informally, host must decide where to "send" the packet:
  1. if destination IP is in the same subnet, packet can be sent directly to the receiver.
  2. Otherwise, must give the packet to the the gateway router for relay.
- A solid answer requires knowledge of Link Layer, Chapter 6.
  - Specifically, we are choosing to address the packet with either the MAC address of the gateway or the final destination.



How many subnets?

- Count the groups of interfaces that can communicate without passing through a router.

- There are six subnets



IPv4 addressing

- ICANN distributes chunks of IP addresses
- *Early scheme:* class A, B, C subnets (/8, /16, /24-bit subnets)
  - However, this only allowed block of 256, 65.5k, 17M addresses. *Very wasteful!*
- *Nowadays:* Classless InterDomain Routing (**CIDR**) allows blocks to be distributed with any power of 2 subnet size (2^x)

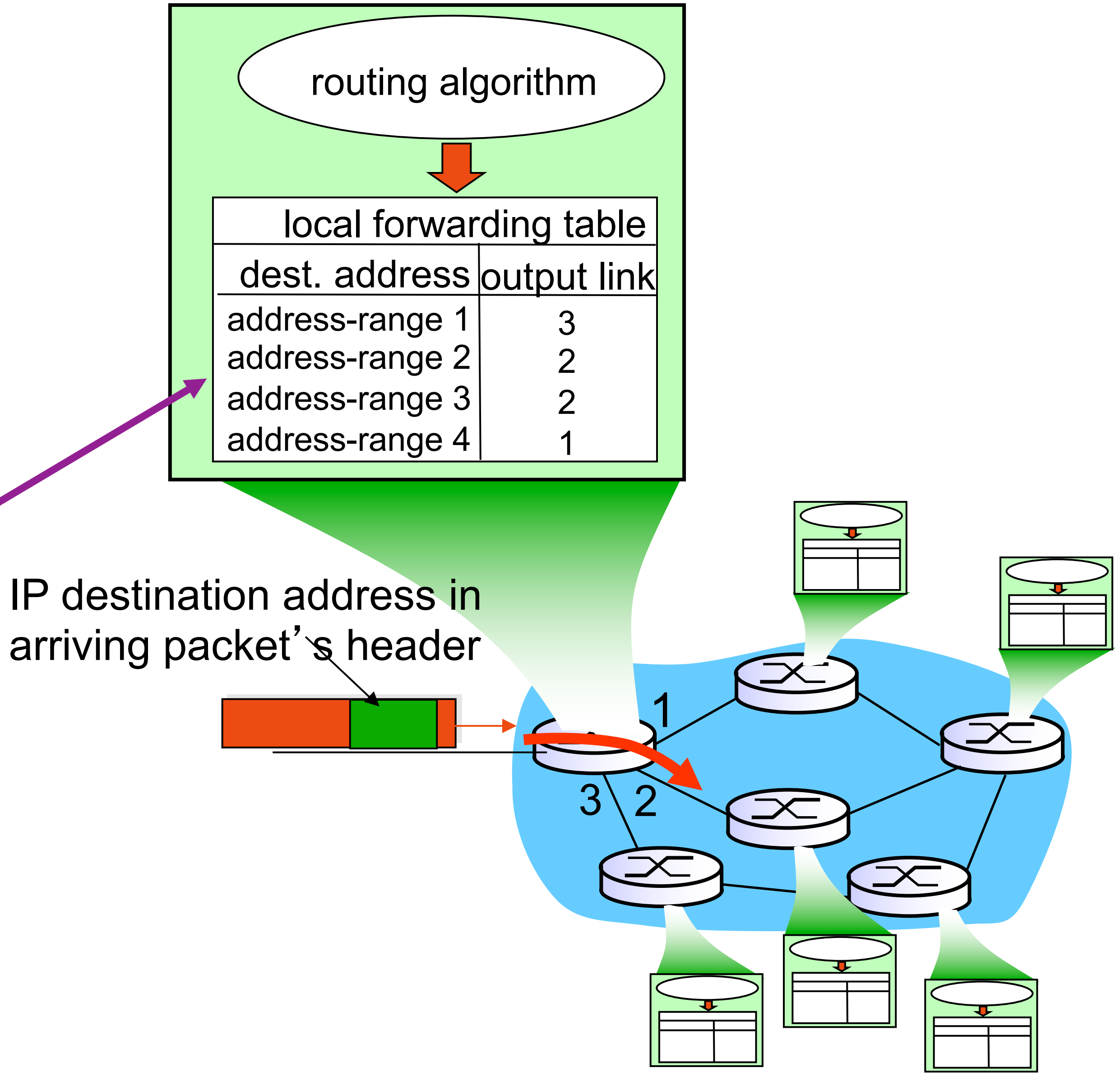
← subnet part → ← host part →

11001000 00010111 00010000 00000000

200.23.16.0/23

Forwarding rules

- Stored in each router in its local **forwarding table**
- Rules decide which link a packet should be sent out on, depending on the packet's destination IP address.
- Billions of IP addresses are possible, so rules apply to ***ranges*** of addresses.



Forwarding tables in IP use **longest prefix matching**

- **IP address ranges** are expressed in an unusual way – by prefix.
  - This can only express ranges of size 2^n and whose starting address is a multiple of 2^n (it must have n zeros in the least-significant bits).
- Router looks for a rule that matches the high bits of the address.
  - IPv4 addresses are 32 bits (up to 4 billion possible addresses)
- If more than one rule matches, choose the most specific one. In other words, choose the rule with *longest prefix*.

STOP and THINK

How is the last address in the range computed?

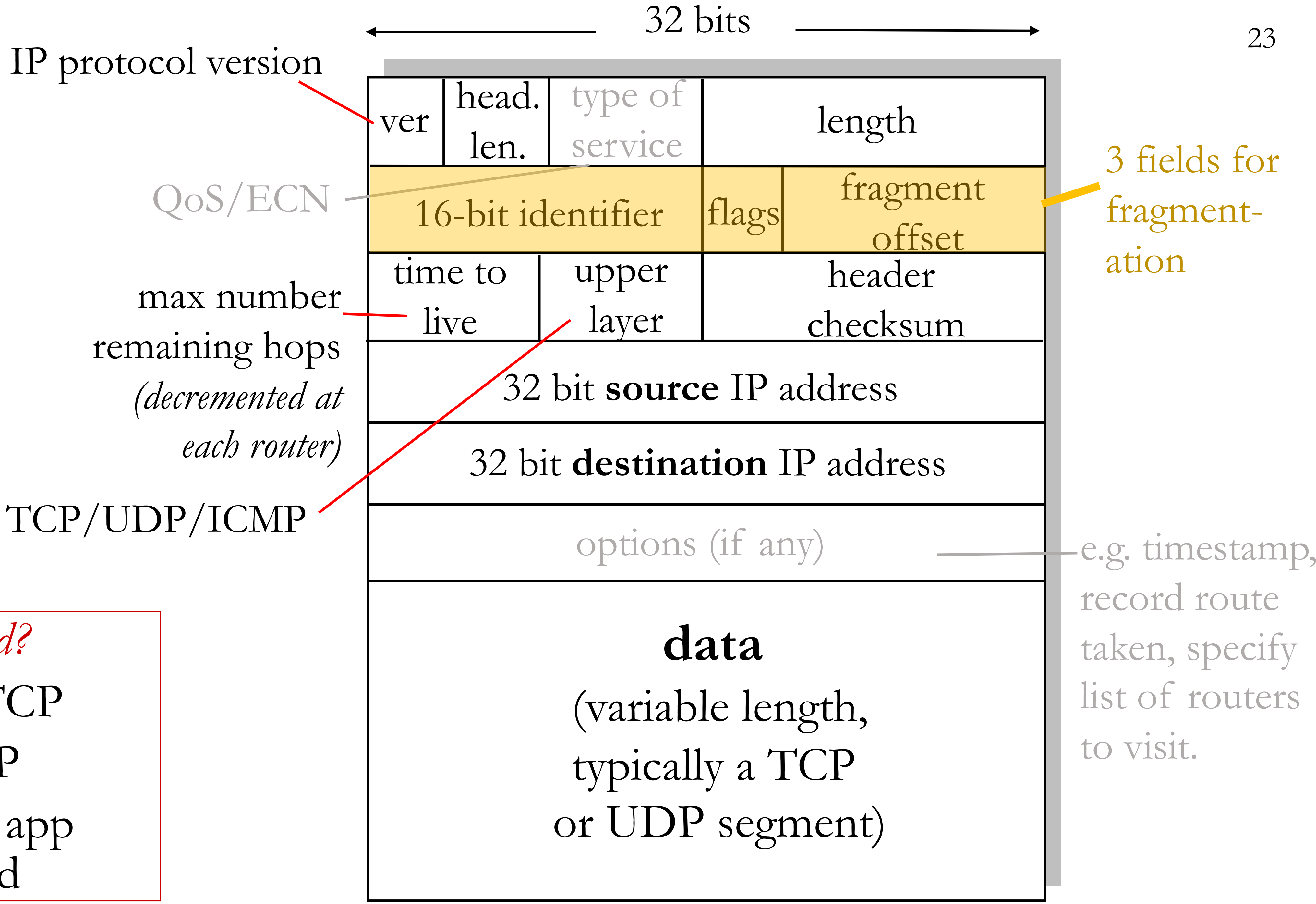
Forwarding Table

Destination Address Range	Link
11001000 00010111 00010*** ********	1
11001000 00010111 00011000 ********	3
11001000 00010111 00011*** ********	2
******** ******** ******** ********	3

Range Meaning

First address	Last address	CIDR notation
200.23.16.0	200.23.23.255	200.23.16.0/**21**
200.23.24.0	200.23.24.255	200.23.24.0/**24**
200.23.24.0	200.23.31.255	200.23.24.0/**21**
0.0.0.0	255.255.255.255	0.0.0.0/**0**

IPv4 packet (datagram)

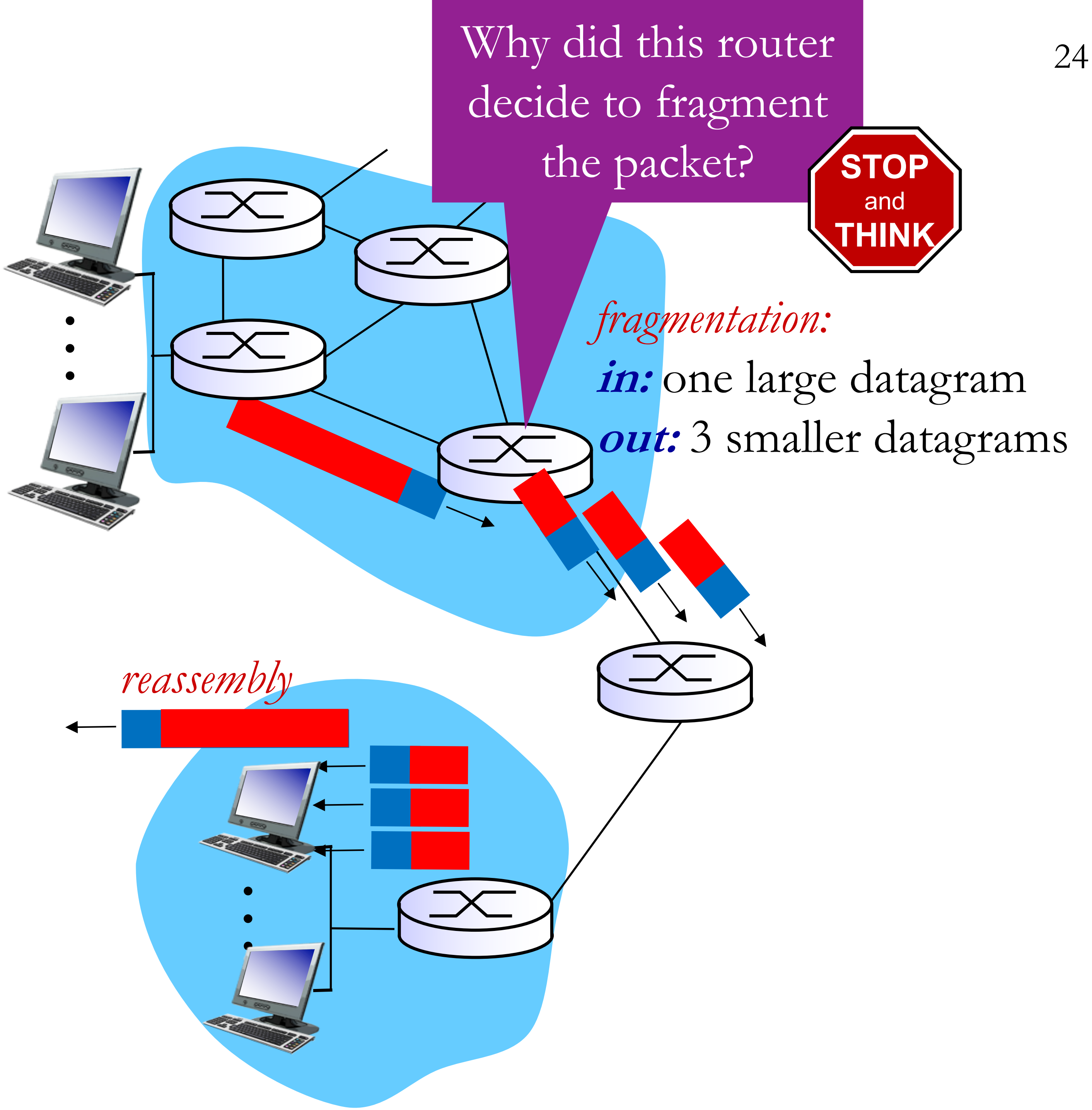


how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

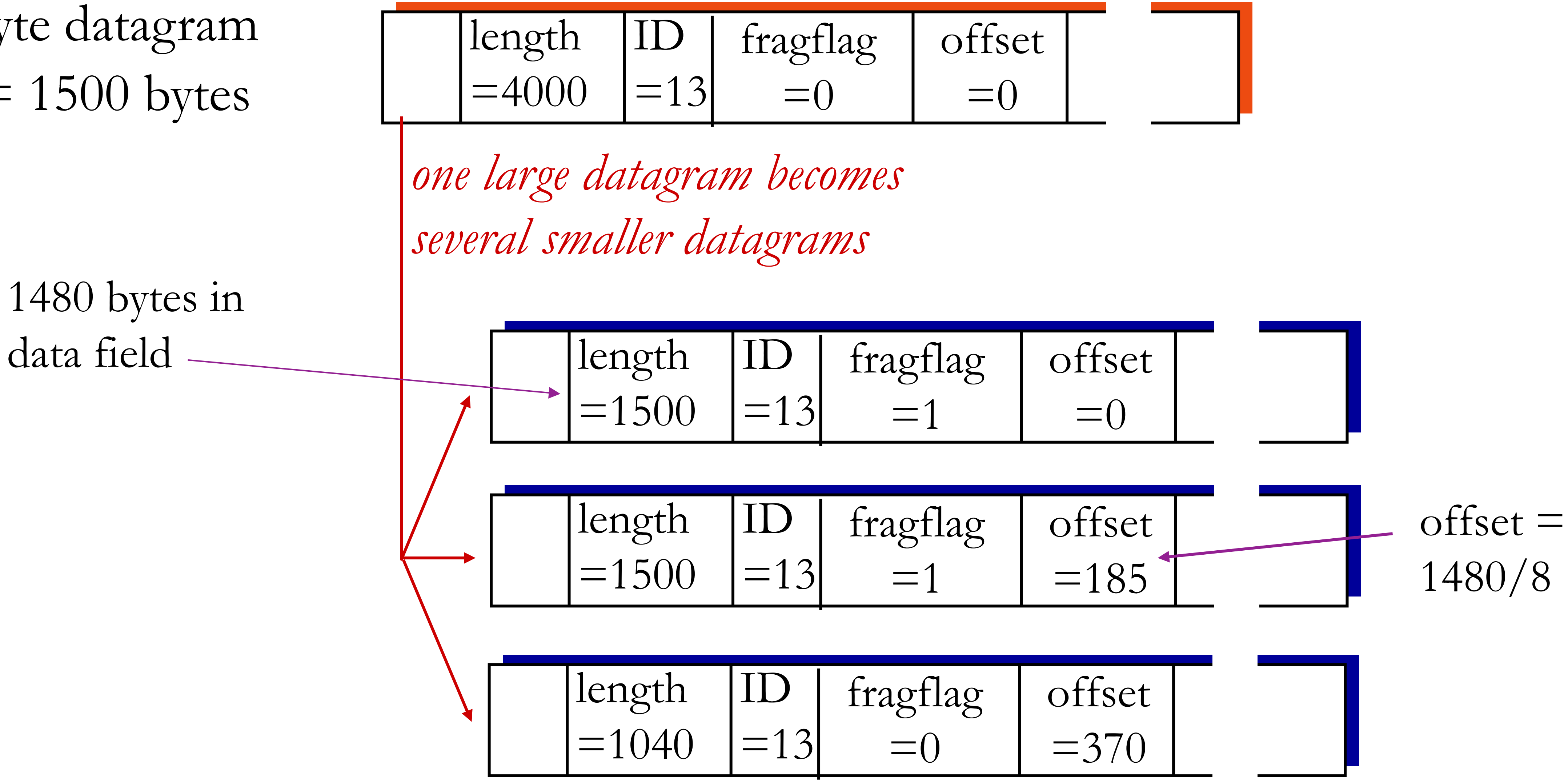
IPv4 fragmentation

- Links in network may have different maximum transmission unit (MTU)
  - Usually MTU=1500 bytes
- Large IP datagrams may be fragmented anywhere, and *reassembled* at final destination.
- Much simpler than TCP stream reassembly:
  - No ACKs or retransmissions



IPv4 fragmentation & reassembly example

- 4000 byte datagram
- MTU = 1500 bytes



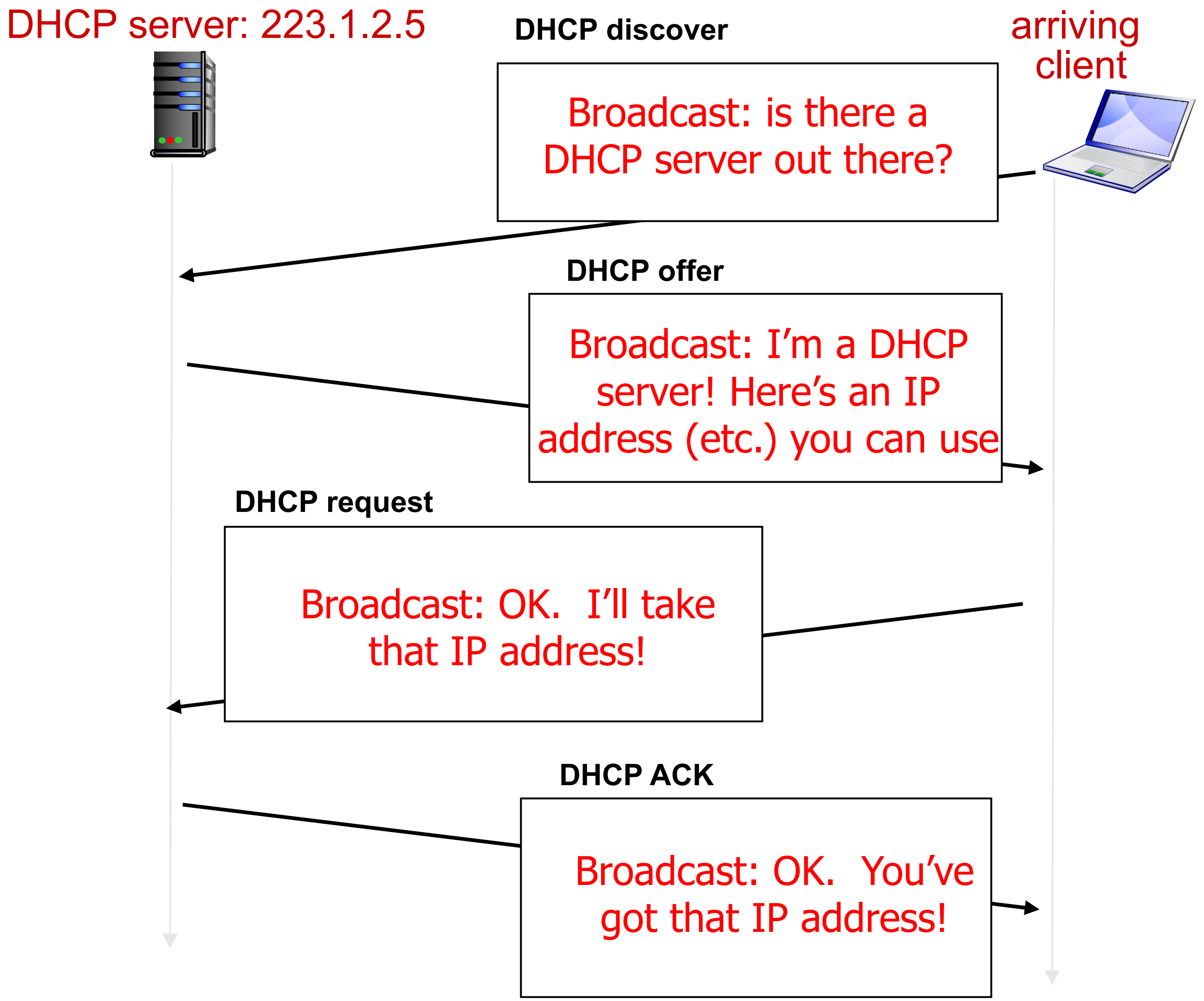
Intermission

IP network configuration

- For a host to operate on the Internet it must know:
  - Its **own IP address** – to set the source address in outbound messages.
  - The **subnet mask** – to identify which IP address can be reached locally, using link-layer MAC address. (Who are my neighbors?)
  - The **gateway IP address** – the router to which it will send messages destined for IP addresses outside of the subnet.
  - The **DNS server IP address** – to get IP addresses for hostnames.
- In total, four 32-bit numbers are needed.
  - Can be statically (manually) configured.
  - Or, use *dynamic host configuration protocol* **(DHCP)** to request these parameters.
- DHCP allows machines to be *mobile*: connect to new network and get new network configuration automatically.

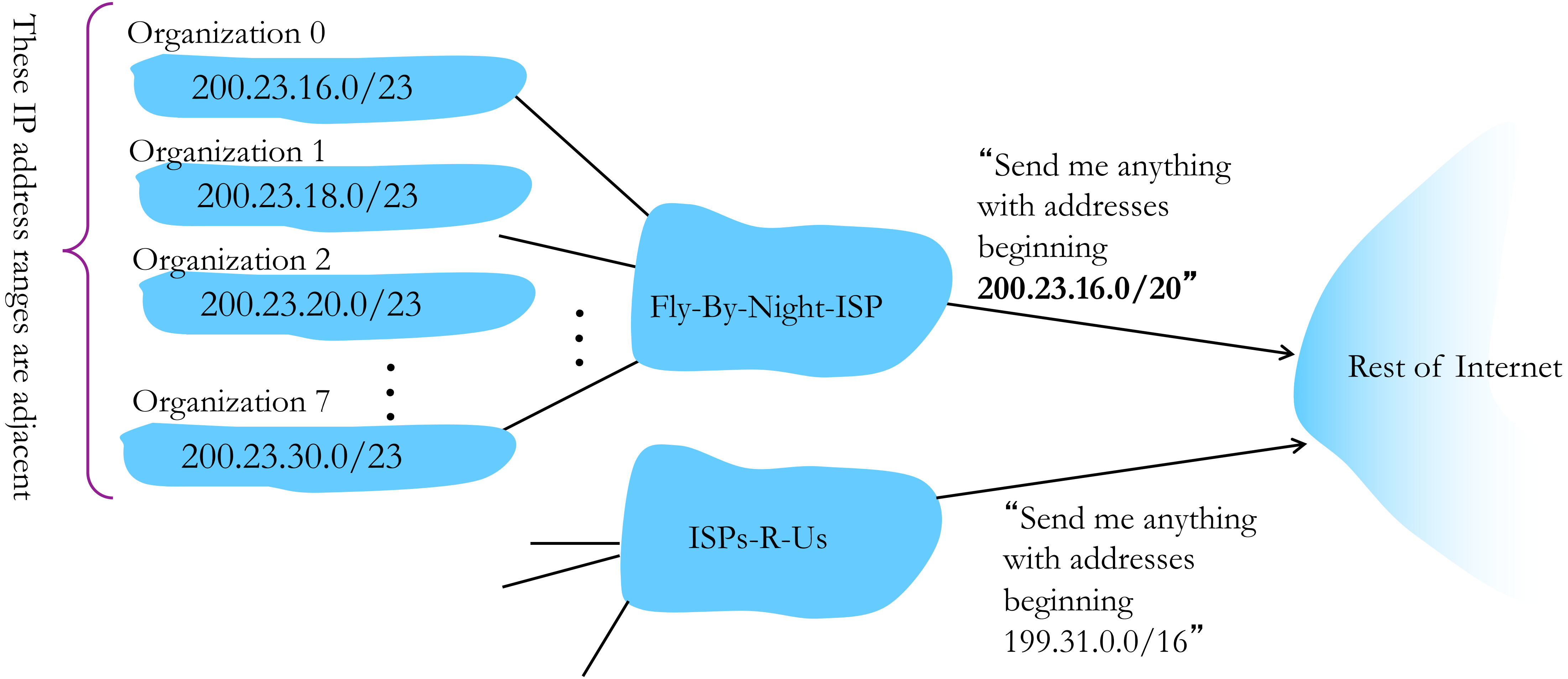
DHCP, greatly simplified

- DHCP server usually runs on a nearby router (*the* router in home networks):
- But how do we communicate at all if we don't have an IP address?
  - Send **local broadcasts** – messages that are received by everyone in the subnet.
  - We'll come back to DHCP when we discuss the Ethernet layer.



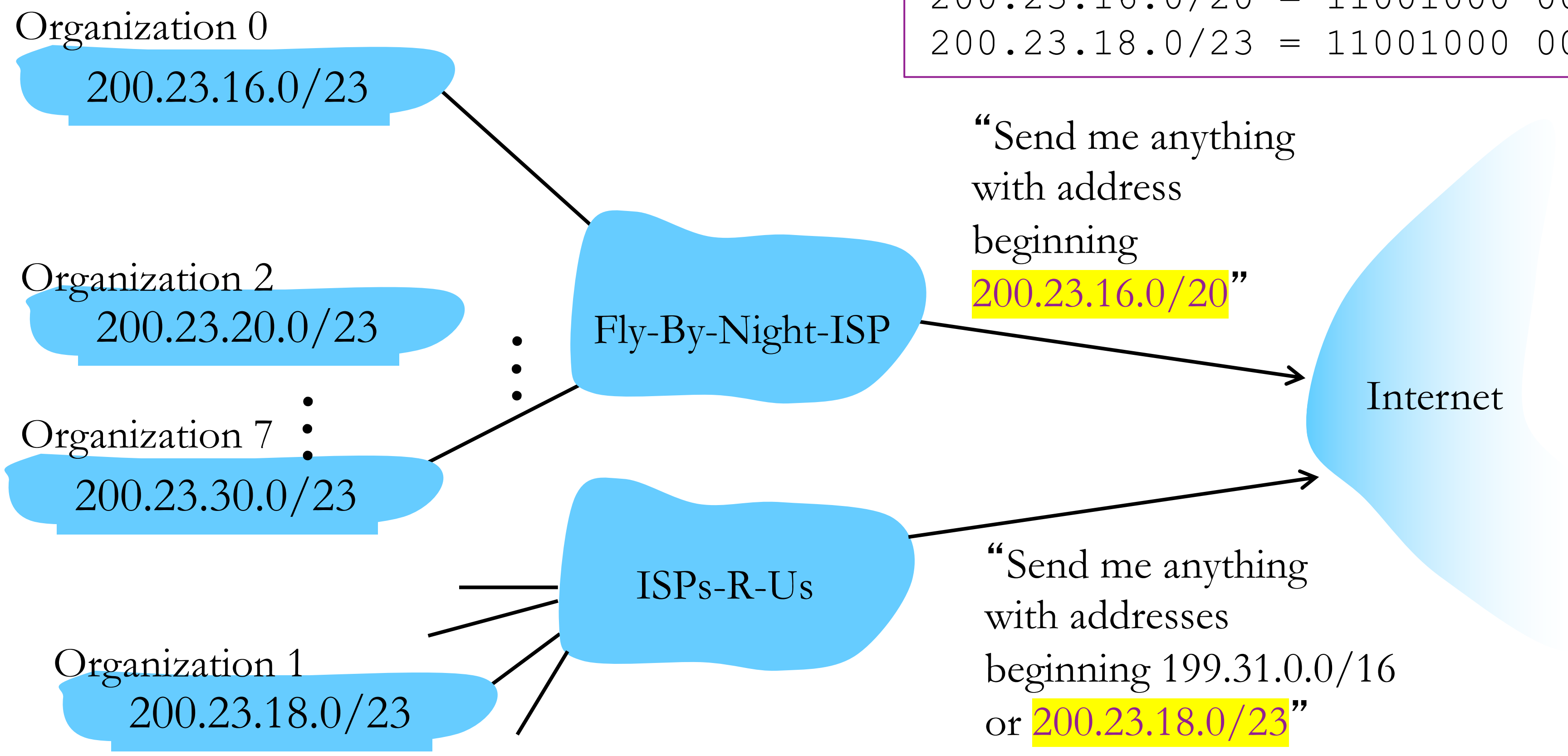
Hierarchical addressing

- Routing tables are much simplified (due to *route aggregation*) if addresses are distributed *hierarchically*, corresponding to physical connections:



Longest-prefix matching, again

```
200.23.16.0/20 = 11001000 00010111 0001**** ****
200.23.18.0/23 = 11001000 00010111 0001001* ****
```



- 200.23.16.0/20 and 200.23.18.0/23 IP ranges overlap!
- But IPSs-R-Us is advertising a more specific route (longer prefix), so it will receive the traffic for 200.23.18.0/23

Recap

- **IP routing** gets packets to their destination on the net.
- Each router has a **forwarding table** mapping addresses → outbound links.
  - The forwarding action of routers is called the *data plane* in IP networking.
  - Later we will learn how forwarding tables are determined (the *control plane*).
- IPv4 *fragments* packets larger than *MTU*. Are reassembled at the destination.
- IP **subnets** define ranges of address that can communicate directly
  - **CIDR notation** (123.100.16.0/28) specifies a range of addresses
    - Used both for specifying subnets and for routing rules.
    - /28 or 255.255.255.240 is called a **subnet mask**.
- Host's IP configuration is: *address*, *subnet mask*, *gateway*, and *DNS server*
  - **Gateway** is IP address of the router who will route packets outside the subnet
  - **DHCP** allows newly-arriving machines to request an IP configuration.