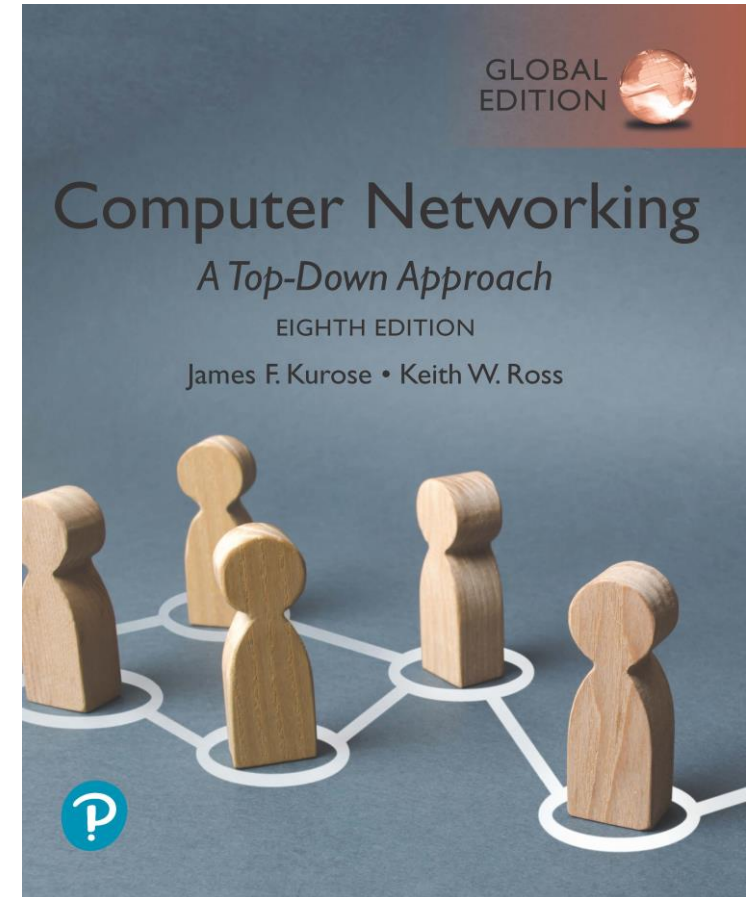


# Chapter 4

## Network Layer: Data Plane



### *Computer Networking: A Top-Down Approach*

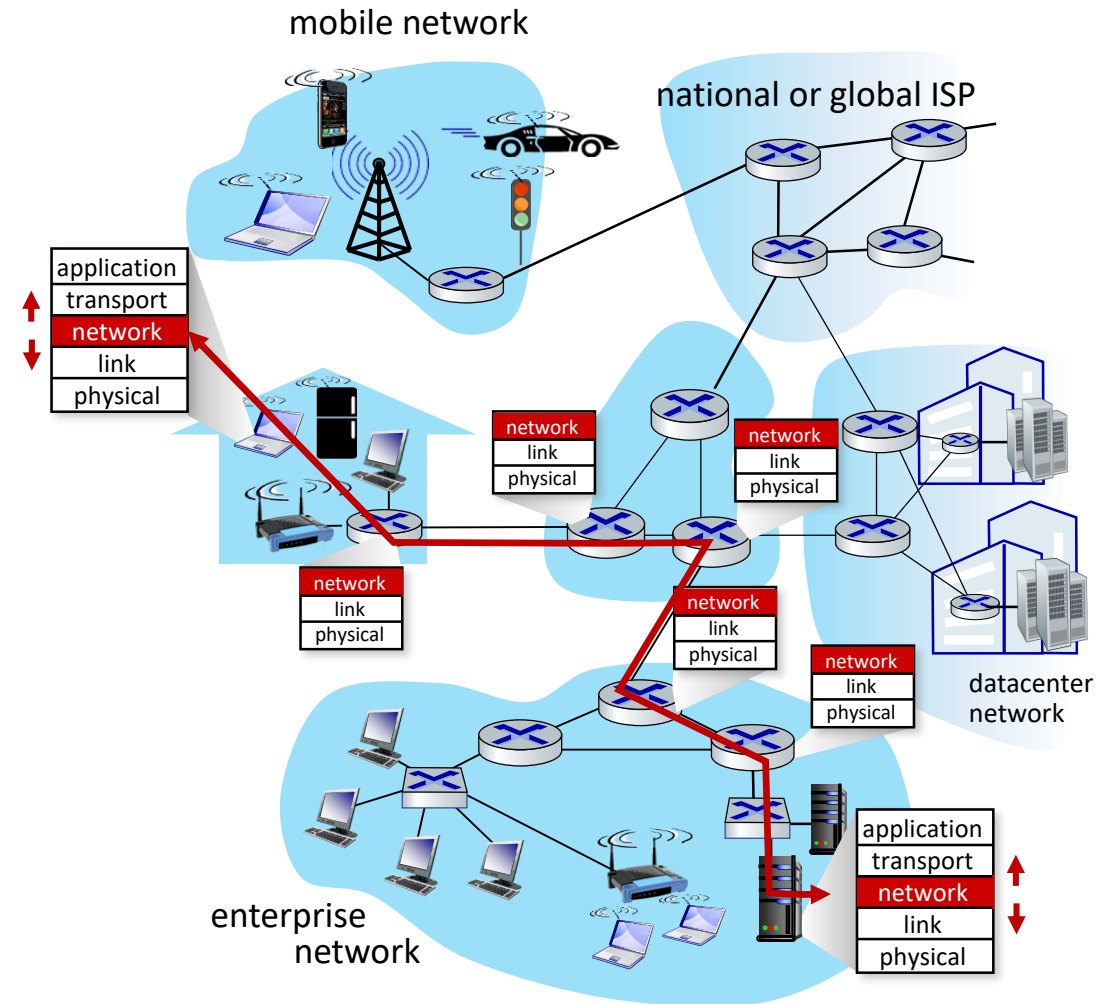
8<sup>th</sup> Edition, Global Edition

Jim Kurose, Keith Ross

Copyright © 2022 Pearson Education Ltd

# Network-layer services and protocols

- transport segment from sending to receiving host
  - **sender:** encapsulates segments into datagrams, passes to link layer
  - **receiver:** delivers segments to transport layer protocol
- network layer protocols in *every Internet device*: hosts, routers
- **routers:**
  - examines header fields in all IP datagrams passing through it
  - moves datagrams from input ports to output ports to transfer datagrams along end-end path



# Two key network-layer functions

## network-layer functions:

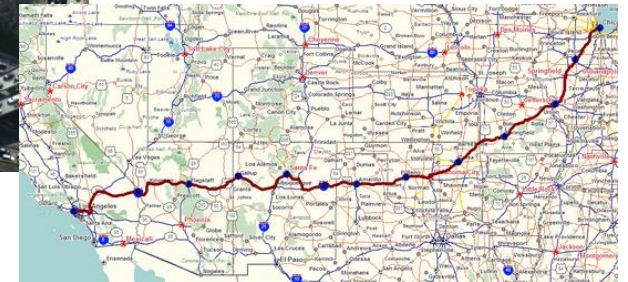
- *forwarding*: move packets from a router's input link to appropriate router output link
- *routing*: determine route taken by packets from source to destination
  - *routing algorithms*

## analogy: taking a trip

- *forwarding*: process of getting through single interchange
- *routing*: process of planning trip from source to destination



forwarding



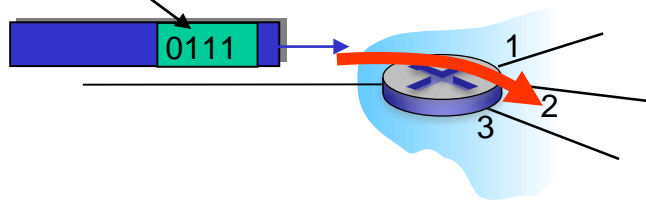
routing

# Network layer: data plane, control plane

## Data plane:

- *local*, per-router function
- determines how datagram arriving on router input port is forwarded to router output port

values in arriving  
packet header

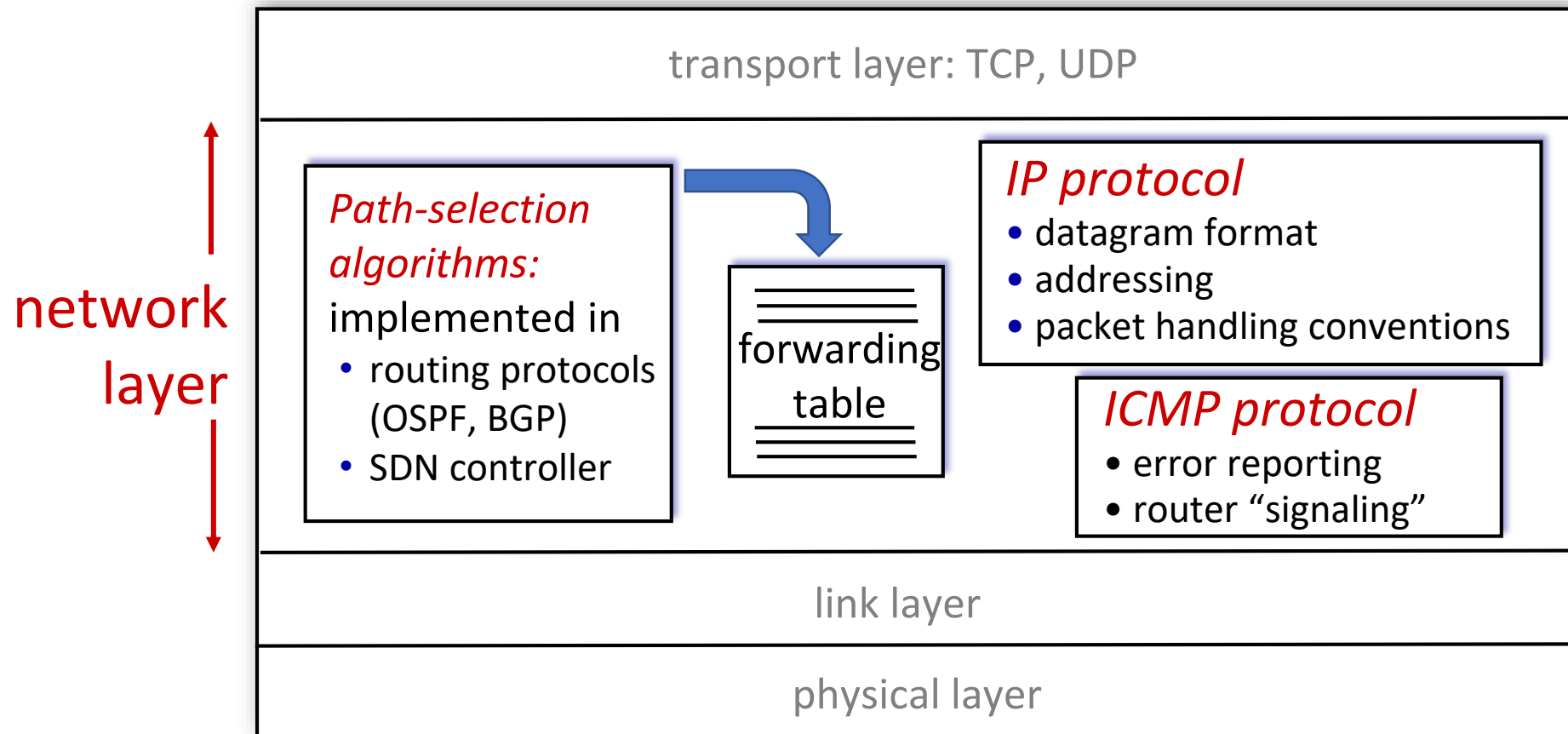


## Control plane

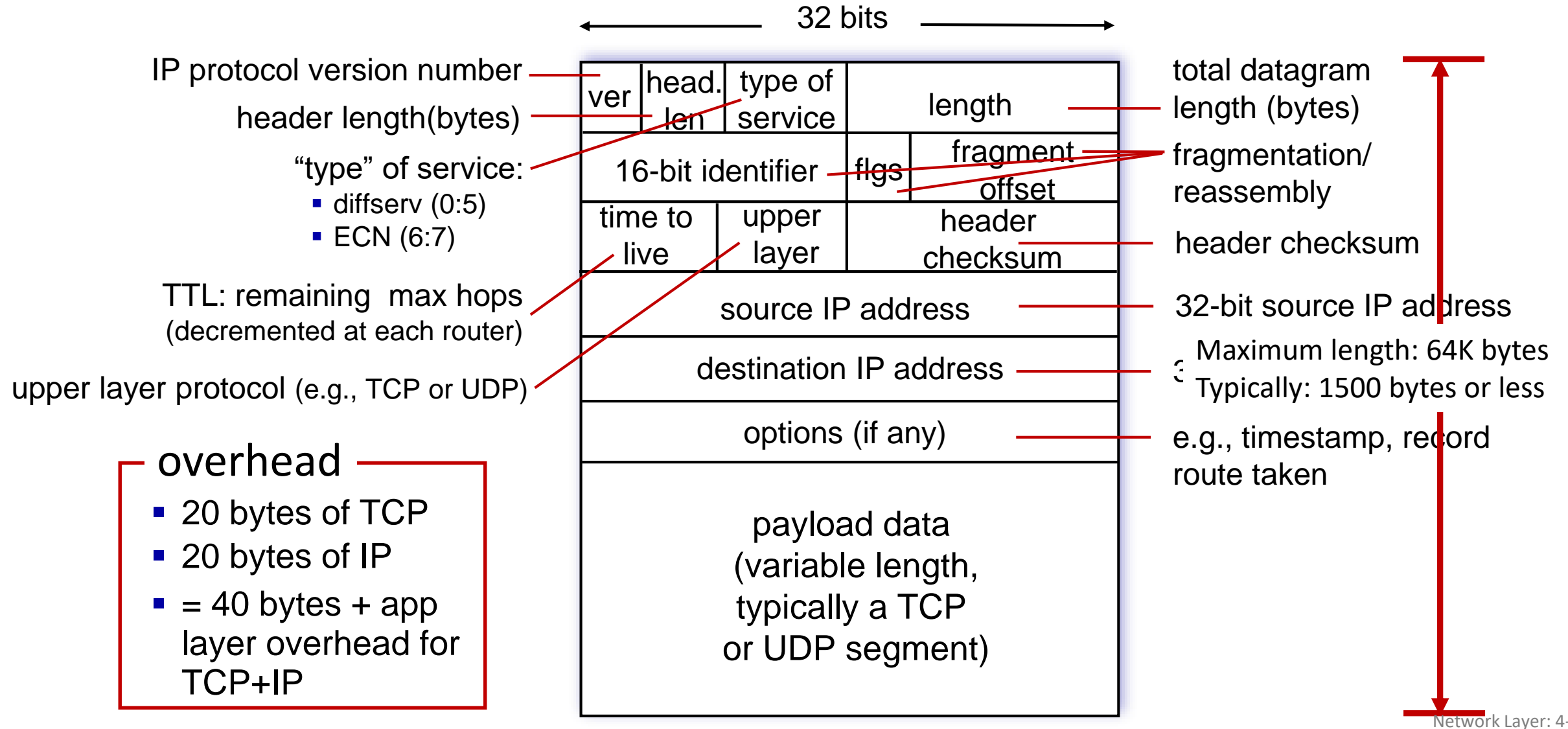
- *network-wide* logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- two control-plane approaches:
  - *traditional routing algorithms*: implemented in routers
  - *software-defined networking (SDN)*: implemented in (remote) servers

# Network Layer: Internet

host, router network layer functions:

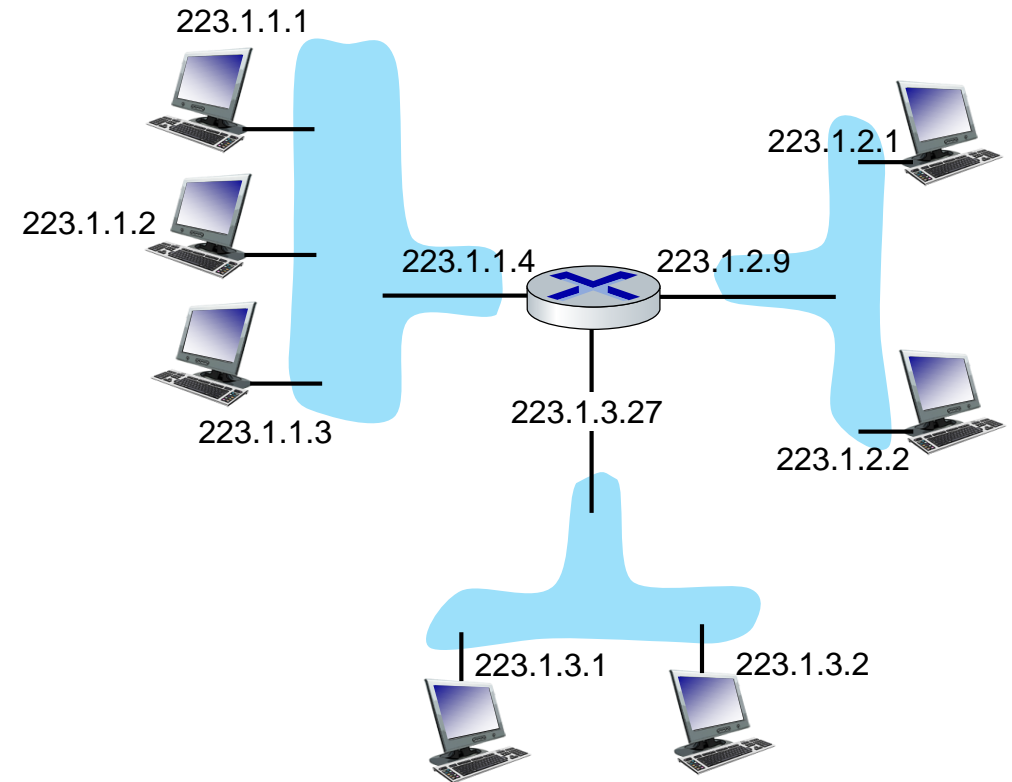


# IP Datagram format



# IP addressing: introduction

- **IP address:** 32-bit identifier associated with each host or router *interface*
- **interface:** connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)



dotted-decimal IP address notation:

223.1.1.1 = 11011111 00000001 00000001 00000001

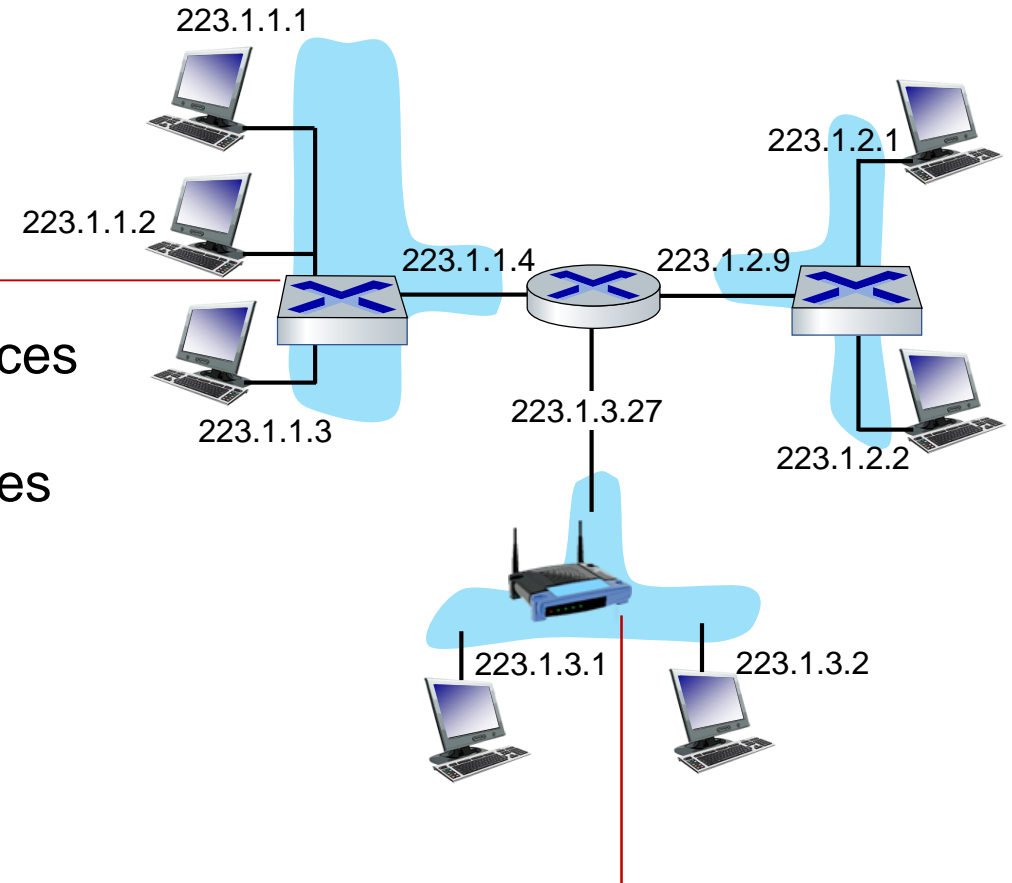
223                      1                      1                      1

Network Layer: 4-7

# IP addressing: introduction

Q: how are interfaces actually connected?

A: wired  
Ethernet interfaces  
connected by  
Ethernet switches

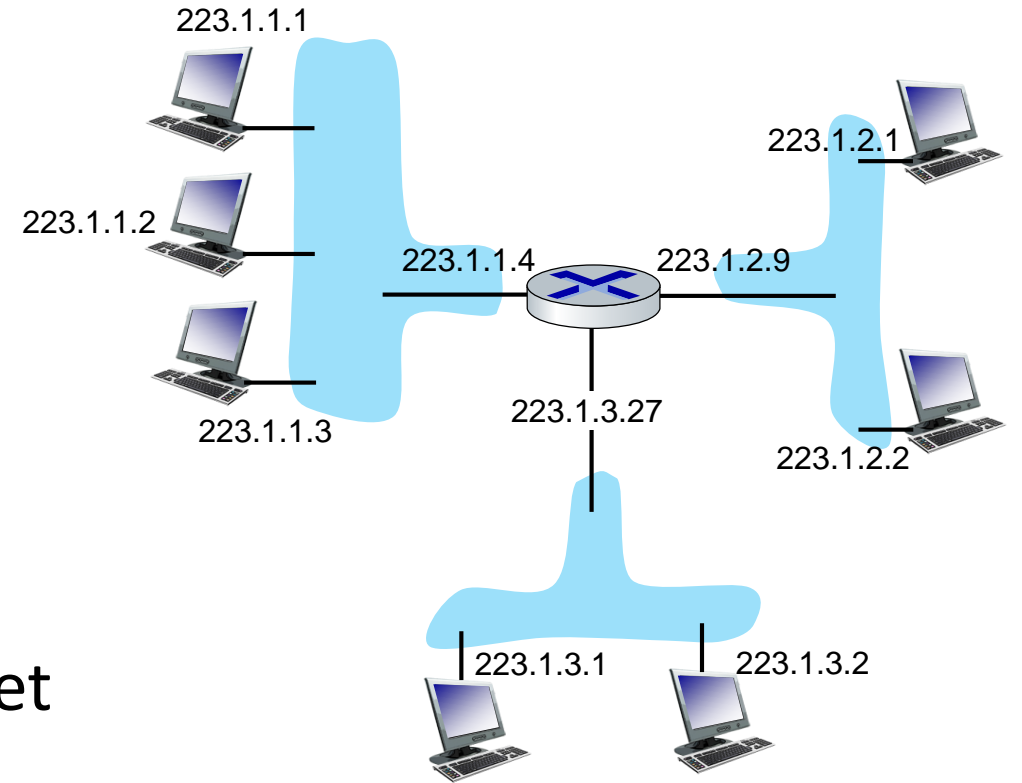


A: wireless WiFi interfaces  
connected by WiFi base station



# Subnets

- *What's a subnet ?*
  - device interfaces that can physically reach each other **without passing through an intervening router**
- IP addresses have structure:
  - **subnet part:** devices in same subnet have common high order bits
  - **host part: remaining** low order bits

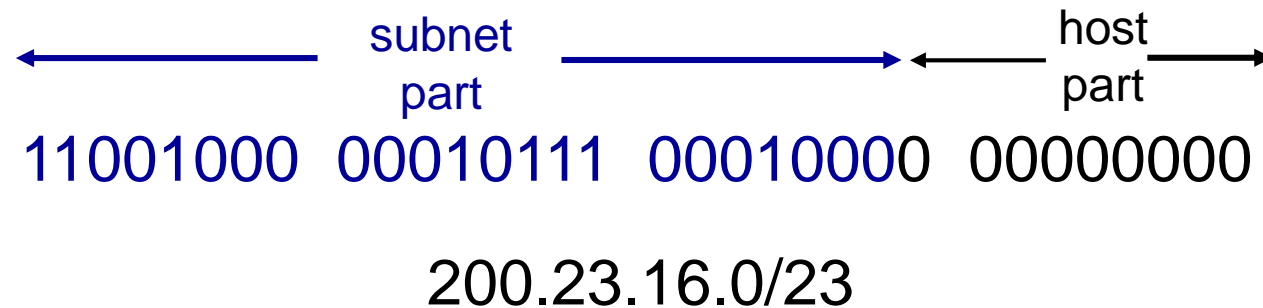


network consisting of 3 subnets

# IP addressing: CIDR

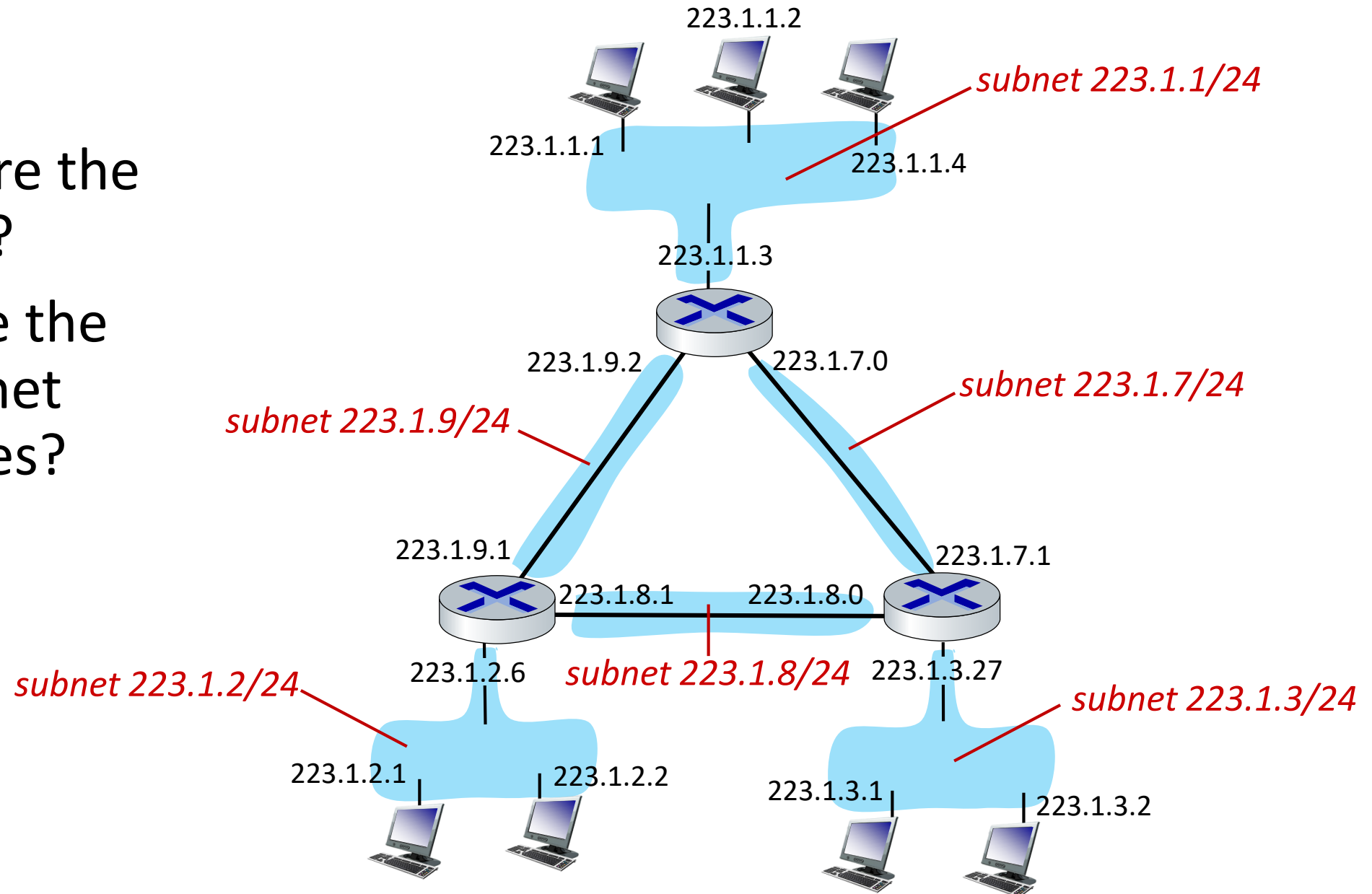
**CIDR: Classless InterDomain Routing** (pronounced “cider”)

- subnet portion of address of arbitrary length
- address format: **a.b.c.d/x**, where x is # bits in subnet portion of address

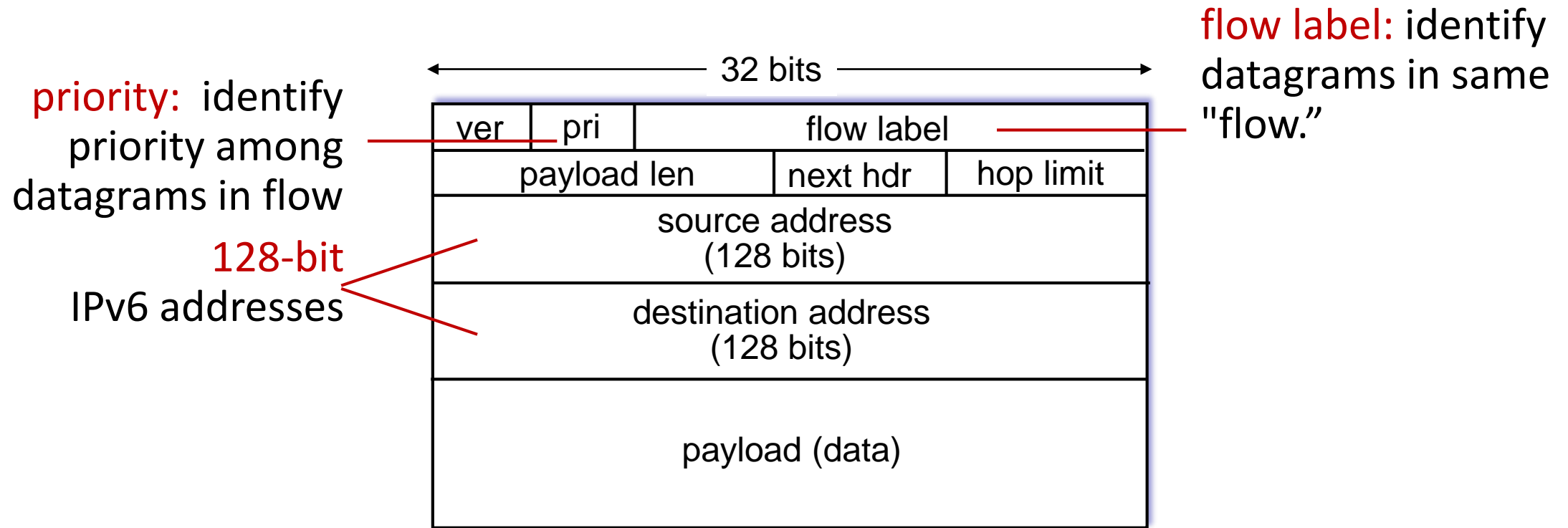


# Subnets

- where are the subnets?
- what are the /24 subnet addresses?



# IPv6 datagram format

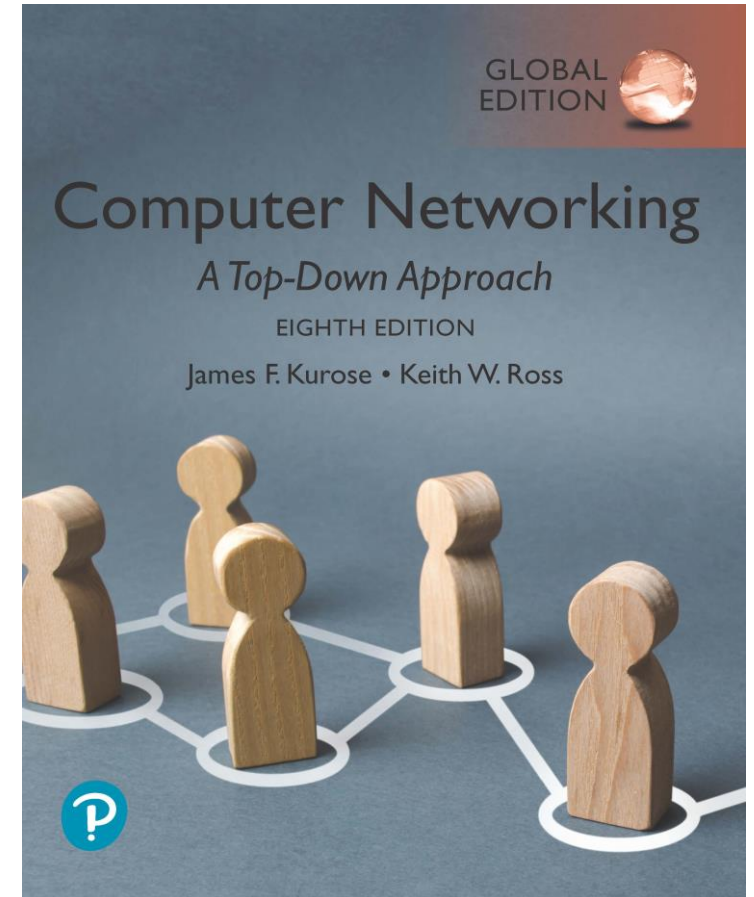


What's missing (compared with IPv4):

- no checksum (to speed processing at routers)
- no fragmentation/reassembly
- no options (available as upper-layer, next-header protocol at router)

# Chapter 5

## Network Layer: Control Plane



### *Computer Networking: A Top-Down Approach*

8<sup>th</sup> Edition, Global Edition

Jim Kurose, Keith Ross

Copyright © 2022 Pearson Education Ltd

# Network-layer functions

- **forwarding**: move packets from router's input to appropriate router output
- **routing**: determine route taken by packets from source to destination

*data plane*

*control plane*

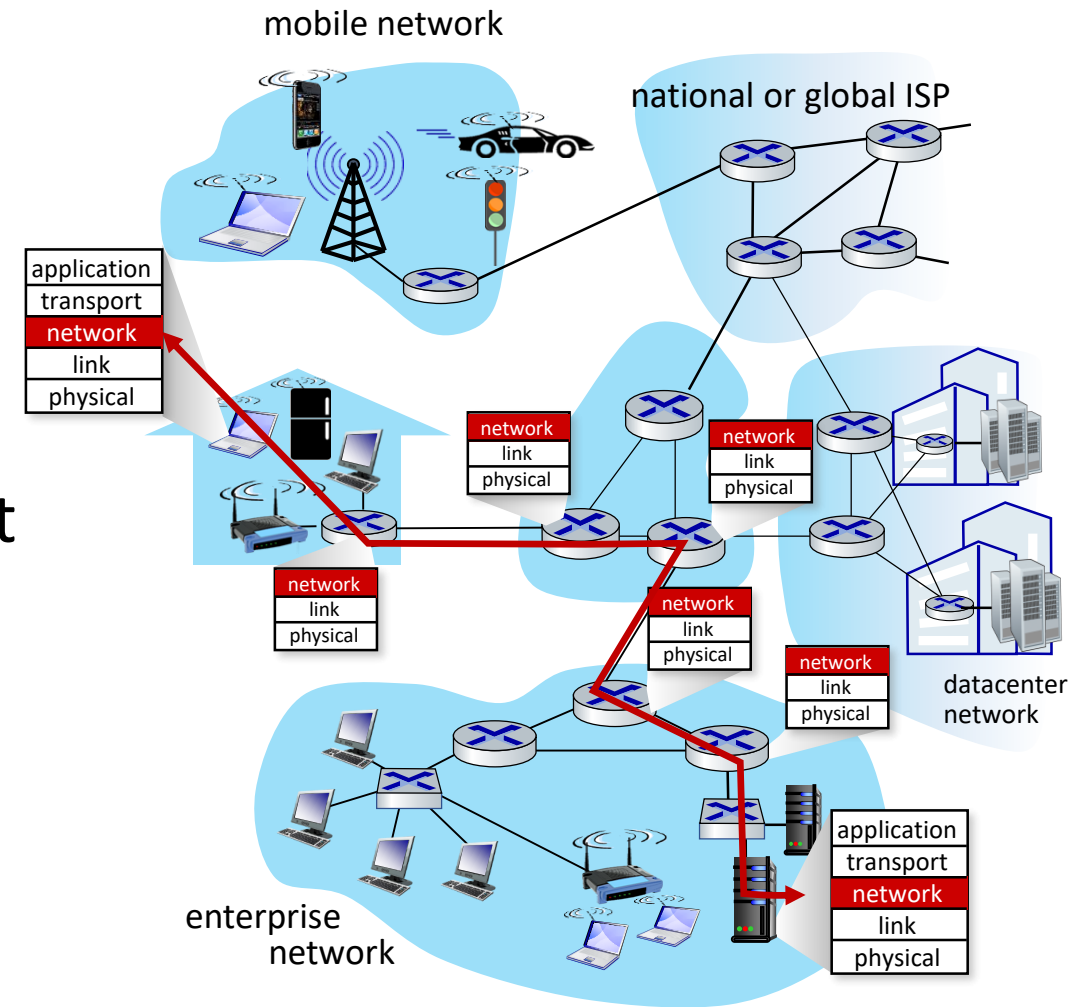
## Two approaches to structuring network control plane:

- per-router control (traditional)
- logically centralized control (software defined networking)

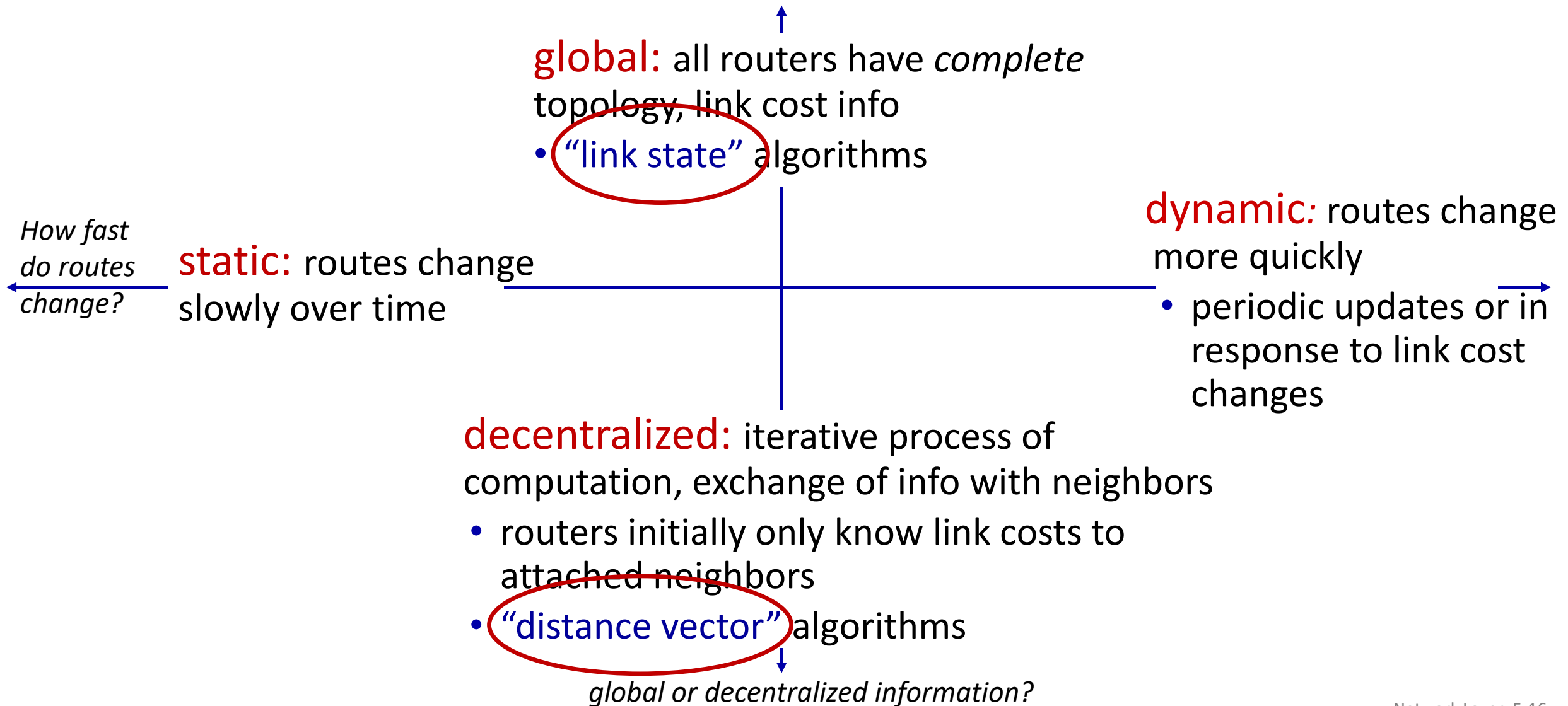
# Routing protocols

**Routing protocol goal:** determine “good” paths (equivalently, routes), from sending hosts to receiving host, through network of routers

- **path:** sequence of routers packets traverse from given initial source host to final destination host
- **“good”:** least “cost”, “fastest”, “least congested”
- routing: a “top-10” networking challenge!

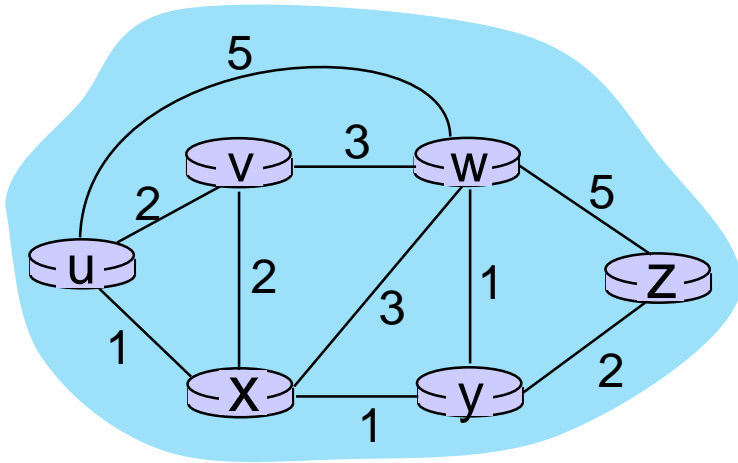


# Routing algorithm classification





# Graph abstraction: link costs



$c_{a,b}$ : cost of *direct* link connecting  $a$  and  $b$

e.g.,  $c_{w,z} = 5$ ,  $c_{u,z} = \infty$

cost defined by network operator:  
could always be 1, or inversely related  
to bandwidth, or inversely related to  
congestion

graph:  $G = (N, E)$

$N$ : set of routers =  $\{ u, v, w, x, y, z \}$

$E$ : set of links =  $\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

# Dijkstra's link-state routing algorithm

- **centralized:** network topology, link costs known to *all* nodes
  - accomplished via “link state broadcast”
  - all nodes have same info
- computes least cost paths from one node (“source”) to all other nodes
  - gives *forwarding table* for that node
- **iterative:** after  $k$  iterations, know least cost path to  $k$  destinations

## notation

- $c_{x,y}$ : direct link cost from node  $x$  to  $y$ ;  $= \infty$  if not direct neighbors
- $D(v)$ : *current* estimate of cost of least-cost-path from source to destination  $v$
- $p(v)$ : predecessor node along path from source to  $v$
- $N'$ : set of nodes whose least-cost-path *definitively* known

# Dijkstra's link-state routing algorithm

1 *Initialization:*

2  $N' = \{u\}$  /\* compute least cost path from u to all other nodes \*/

3 for all nodes  $v$

4 if  $v$  adjacent to  $u$  /\*  $u$  initially knows direct-path-cost only to direct neighbors \*/

5 then  $D(v) = c_{u,v}$  /\* but may not be *minimum* cost! \*/

6 else  $D(v) = \infty$

7



8 *Loop*

9 find  $w$  not in  $N'$  such that  $D(w)$  is a minimum

10 add  $w$  to  $N'$

11 update  $D(v)$  for all  $v$  adjacent to  $w$  and not in  $N'$ :

12  **$D(v) = \min ( D(v), D(w) + c_{w,v} )$**

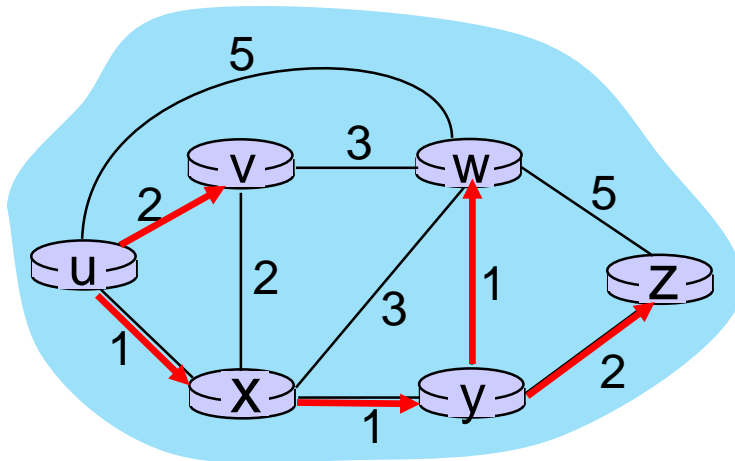
13 /\* new least-path-cost to  $v$  is either old least-cost-path to  $v$  or known

14 least-cost-path to  $w$  plus direct-cost from  $w$  to  $v$  \*/

15 *until all nodes in  $N'$*

# Dijkstra's algorithm: an example

Step	$N'$	$D(v), p(v)$	$D(w), p(w)$	$D(x), p(x)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2, u	5, u	1, u	$\infty$	$\infty$
1	ux	2, u	4, x		2, x	$\infty$
2	uxy	2, u	3, y			4, y
3	uxyv		3, y			4, y
4	uxyvw					4, y
5	uxyvwz					



Initialization (step 0): For all  $a$ : if  $a$  adjacent to then  $D(a) = c_{u,a}$

find  $a$  not in  $N'$  such that  $D(a)$  is a minimum

add  $a$  to  $N'$

update  $D(b)$  for all  $b$  adjacent to  $a$  and not in  $N'$ :

$$D(b) = \min ( D(b), D(a) + c_{a,b} )$$

# Distance vector algorithm

Based on *Bellman-Ford* (BF) equation (dynamic programming):

Bellman-Ford equation

Let  $D_x(y)$ : cost of least-cost path from  $x$  to  $y$ .

Then:

$$D_x(y) = \min_v \{ c_{x,v} + D_v(y) \}$$

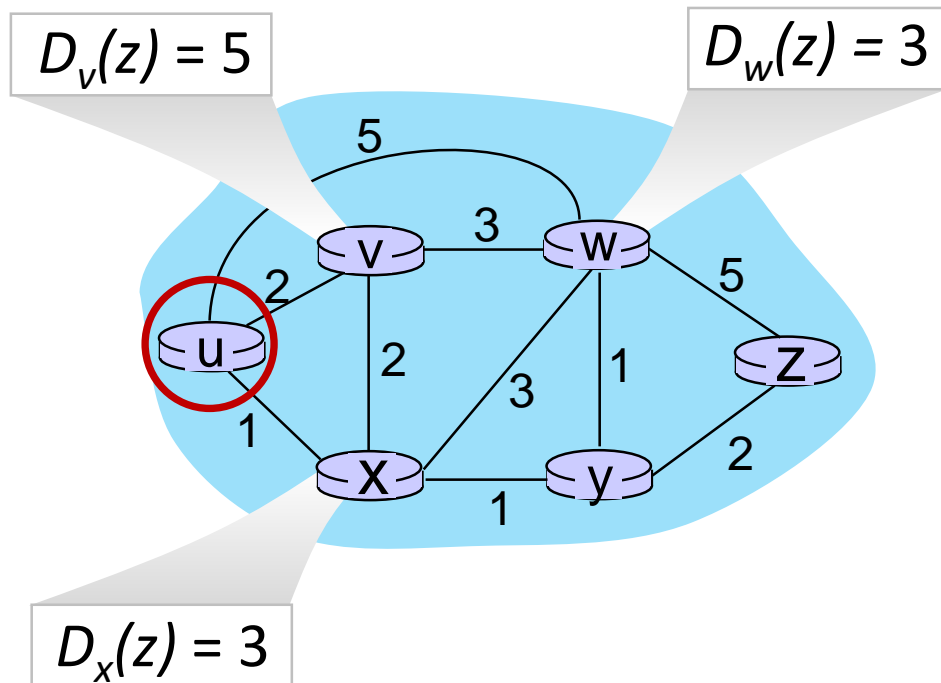
$v$ 's estimated least-cost-path cost to  $y$

$\min$  taken over all neighbors  $v$  of  $x$

direct cost of link from  $x$  to  $v$

# Bellman-Ford Example

Suppose that  $u$ 's neighboring nodes,  $x, v, w$ , know that for destination  $z$ :



Bellman-Ford equation says:

$$\begin{aligned} D_u(z) &= \min \{ c_{u,v} + D_v(z), \\ &\quad c_{u,x} + D_x(z), \\ &\quad c_{u,w} + D_w(z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4 \end{aligned}$$

*node achieving minimum ( $x$ ) is next hop on estimated least-cost path to destination ( $z$ )*

# Distance vector algorithm

## key idea:

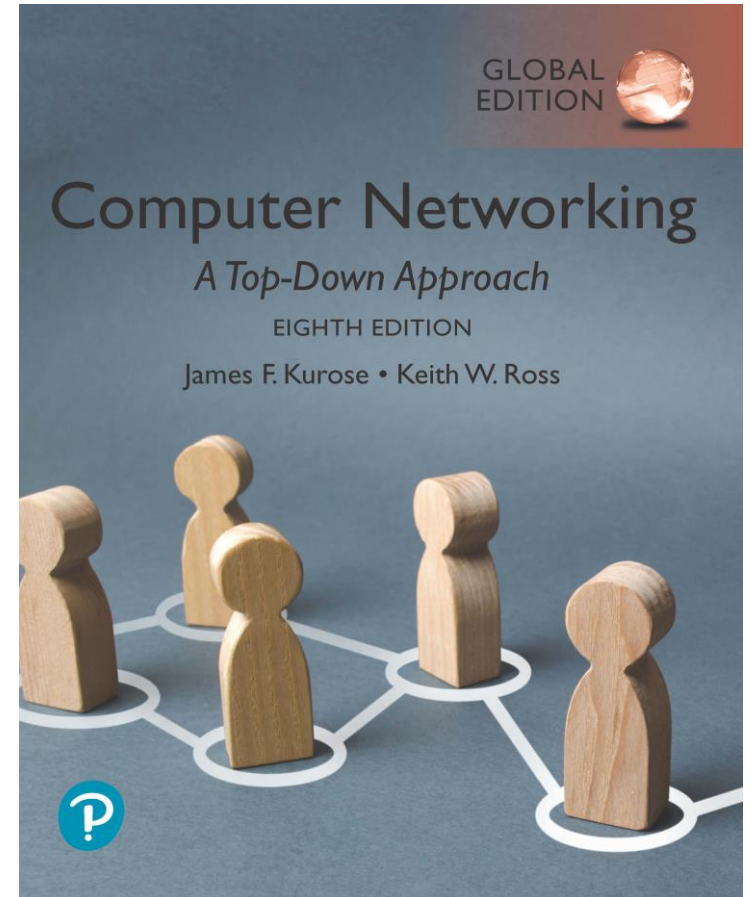
- from time-to-time, each node sends its own distance vector estimate to neighbors
- when  $x$  receives new DV estimate from any neighbor, it updates its own DV using B-F equation:

$$D_x(y) \leftarrow \min_v \{c_{x,v} + D_v(y)\} \text{ for each node } y \in N$$

- under minor, natural conditions, the estimate  $D_x(y)$  converge to the actual least cost  $d_x(y)$

# Chapter 6

## The Link Layer and LANs



### *Computer Networking: A Top-Down Approach*

8<sup>th</sup> Edition, Global Edition

Jim Kurose, Keith Ross

Copyright © 2022 Pearson Education Ltd

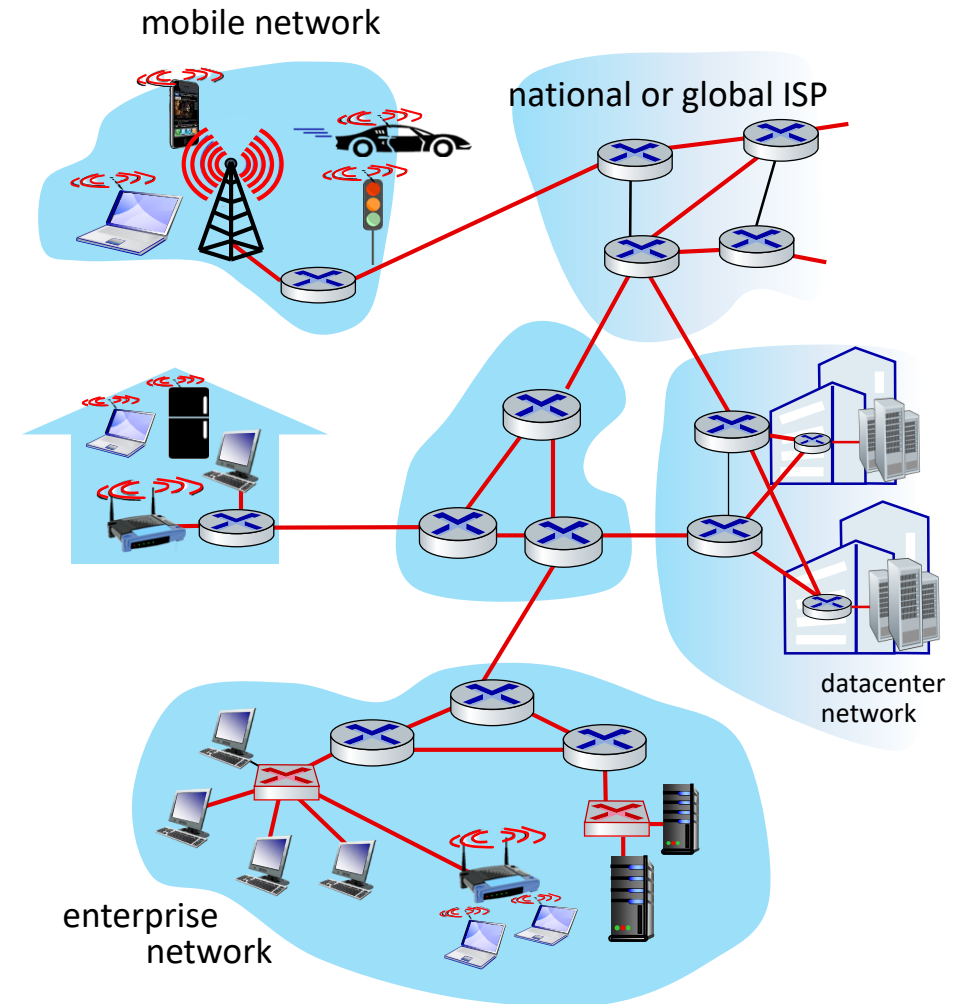


# Link layer: introduction

terminology:

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired
  - wireless
  - LANs
- layer-2 packet: *frame*, encapsulates datagram

*link layer* has responsibility of transferring datagram from one node to *physically adjacent* node over a link



# Link layer: context

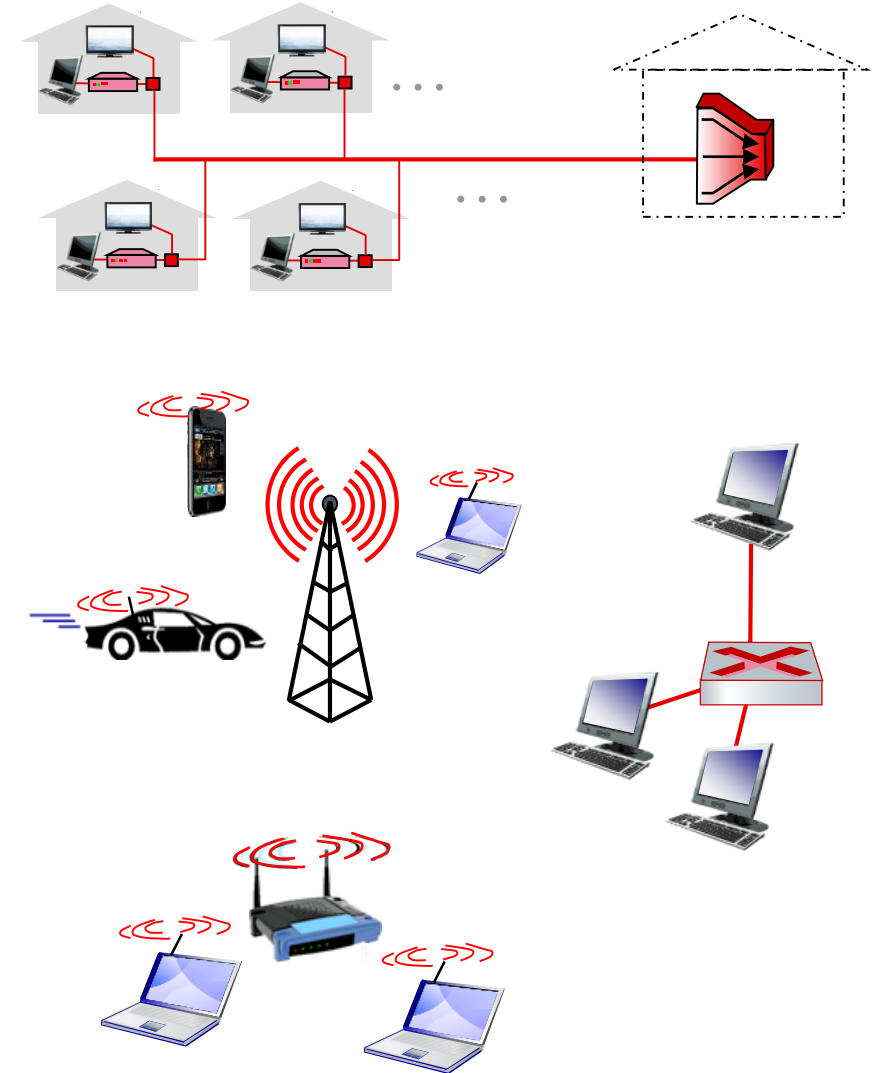
- datagram transferred by different link protocols over different links:
  - e.g., WiFi on first link, Ethernet on next link
- each link protocol provides different services
  - e.g., may or may not provide reliable data transfer over link

## transportation analogy:

- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = **datagram**
- transport segment = **communication link**
- transportation mode = **link-layer protocol**
- travel agent = **routing algorithm**

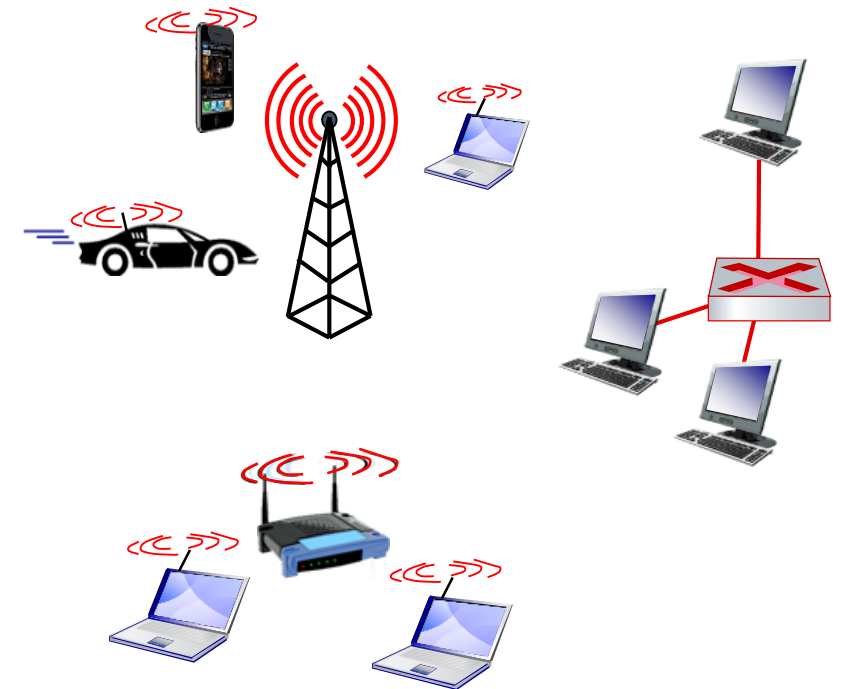
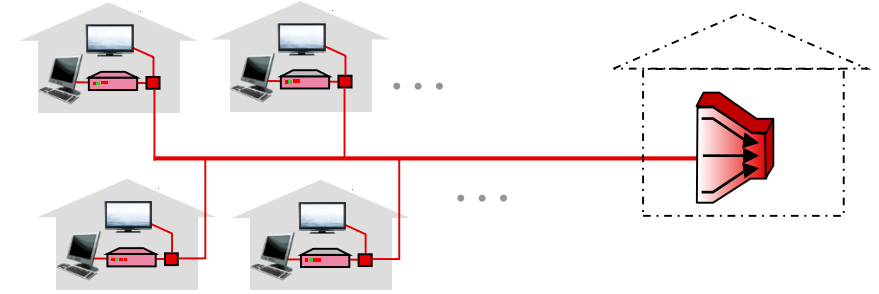
# Link layer: services

- **framing, link access:**
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - “MAC” addresses in frame headers identify source, destination (different from IP address!)
- **reliable delivery between adjacent nodes**
  - seldom used on low bit-error links
  - wireless links: high error rates



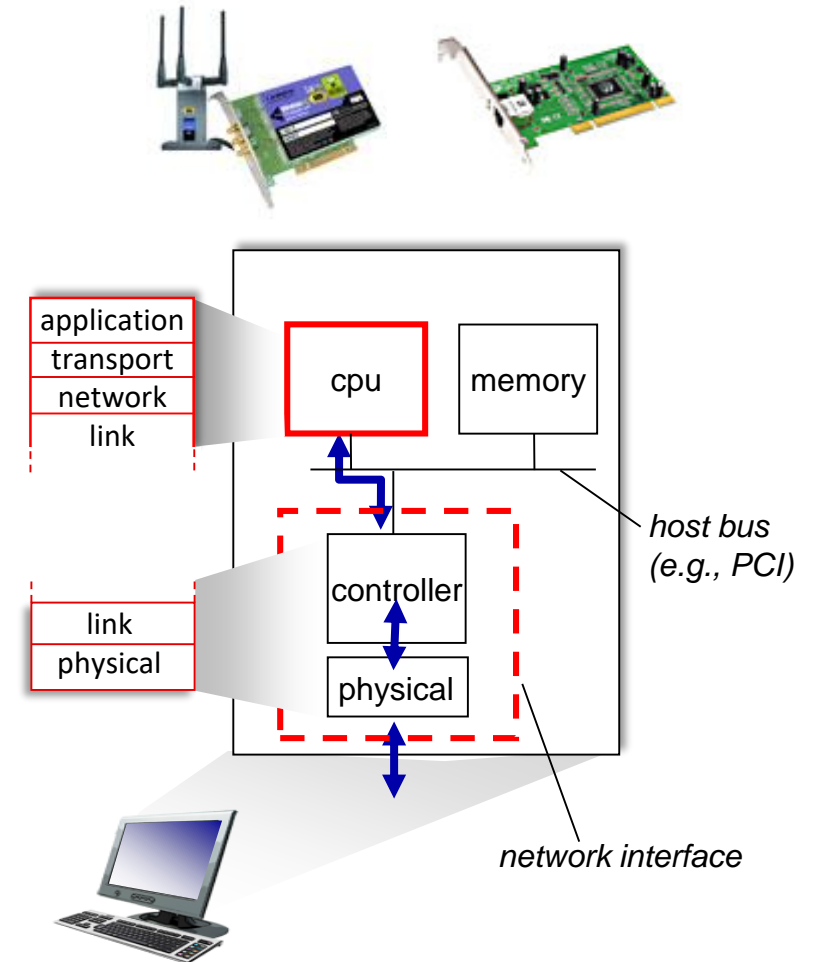
# Link layer: services (more)

- **flow control:**
  - pacing between adjacent sending and receiving nodes
- **error detection:**
  - errors caused by signal attenuation, noise.
  - receiver detects errors, signals retransmission, or drops frame
- **error correction:**
  - receiver identifies *and corrects* bit error(s) without retransmission
- **half-duplex and full-duplex:**
  - with half duplex, nodes at both ends of link can transmit, but not at same time



# Where is the link layer implemented?

- in each-and-every host
- link layer implemented in *network interface card* (NIC) or on a chip
  - Ethernet, WiFi card or chip
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



# Multiple access links, protocols

two types of “links”:

- point-to-point
  - point-to-point link between Ethernet switch, host
  - PPP for dial-up access
- **broadcast (shared wire or medium)**
  - old-fashioned Ethernet
  - upstream HFC in cable-based access network
  - 802.11 wireless LAN, 4G/4G. satellite



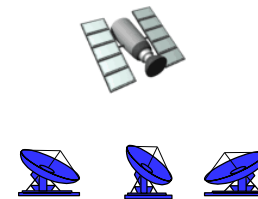
shared wire (e.g.,  
cabled Ethernet)



shared radio: 4G/5G



shared radio: WiFi



shared radio: satellite



humans at a cocktail party  
(shared air, acoustical)

# Multiple access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - *collision* if node receives two or more signals at the same time

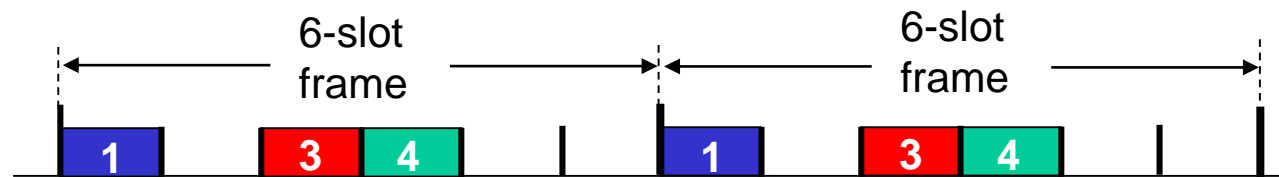
## multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

# Channel partitioning MAC protocols: TDMA

## TDMA: time division multiple access

- access to channel in “rounds”
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle

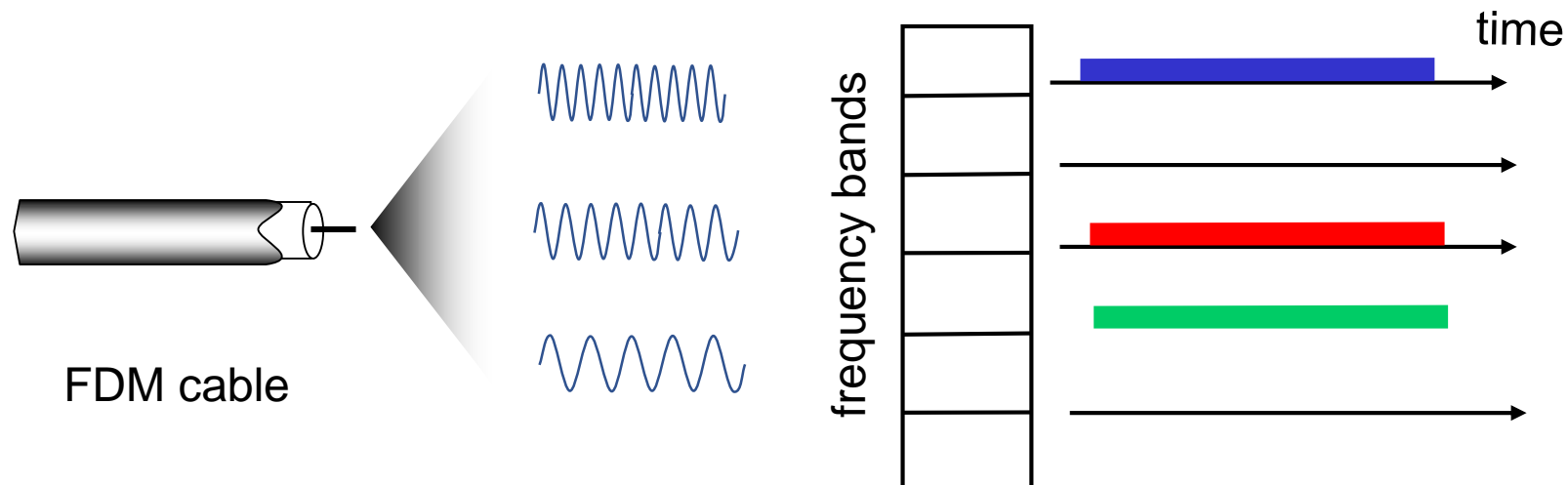




# Channel partitioning MAC protocols: FDMA

## FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have packet to send, frequency bands 2,5,6 idle



# Random access protocols

- when node has packet to send
  - transmit at full channel data rate  $R$ .
  - no *a priori* coordination among nodes
- two or more transmitting nodes: “collision”
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- examples of random access MAC protocols:
  - ALOHA, slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

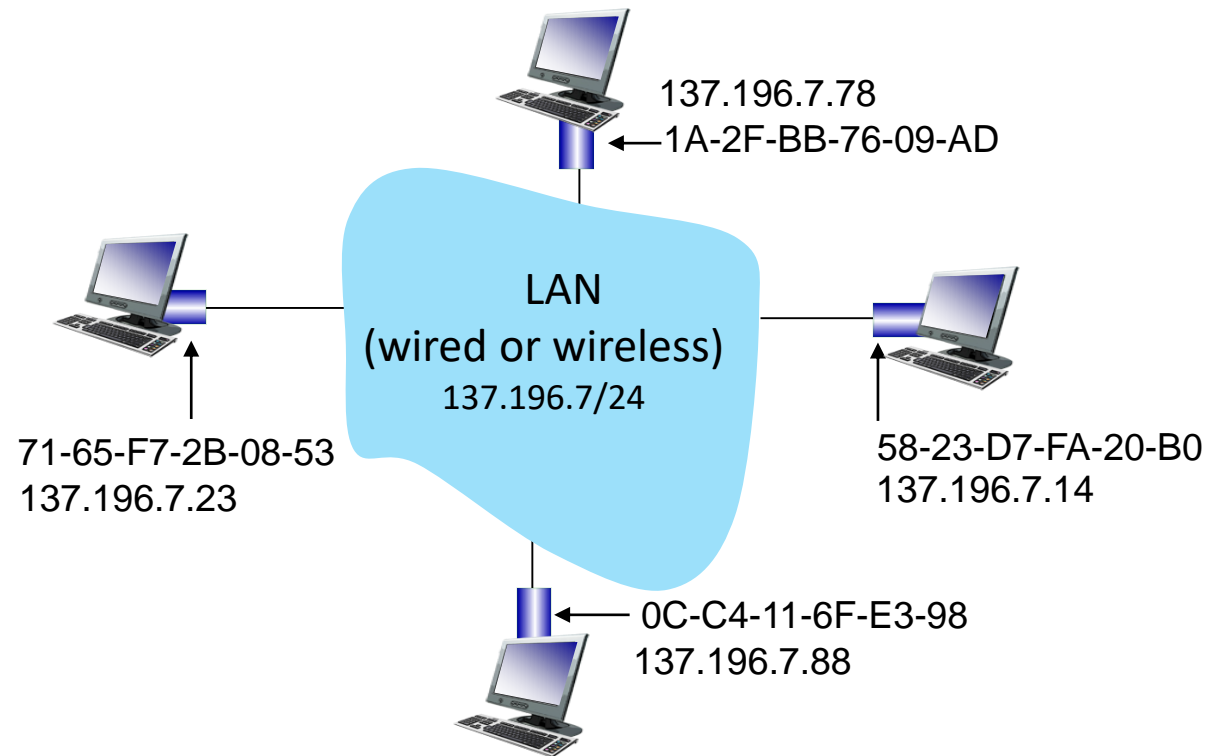
# MAC addresses

- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
  - e.g.: 128.119.40.136
- MAC (or LAN or physical or Ethernet) address:
  - function: used “locally” to get frame from one interface to another physically-connected interface (same subnet, in IP-addressing sense)
  - 48-bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD
    - hexadecimal (base 16) notation  
(each “numeral” represents 4 bits)

# MAC addresses

each interface on LAN

- has unique 48-bit **MAC** address
- has a locally unique 32-bit IP address (as we've seen)



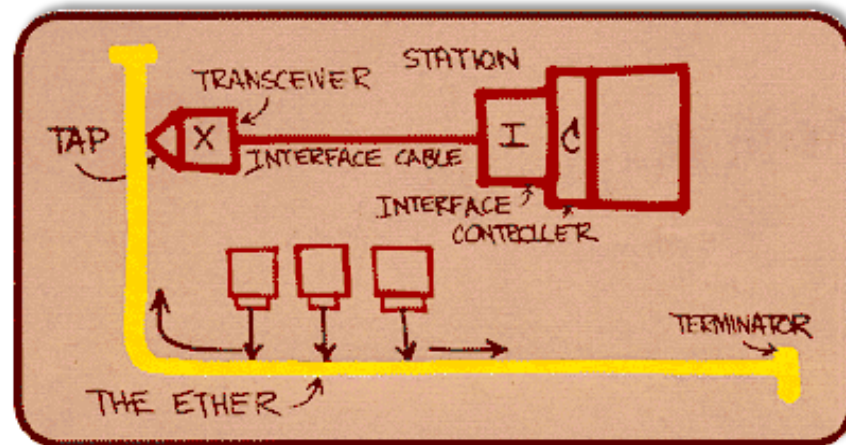
# MAC addresses

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address: portability
  - can move interface from one LAN to another
  - recall IP address *not* portable: depends on IP subnet to which node is attached

# Ethernet

“dominant” wired LAN technology:

- first widely used LAN technology
- simpler, cheap
- kept up with speed race: 10 Mbps – 400 Gbps
- single chip, multiple speeds (e.g., Broadcom BCM5761)

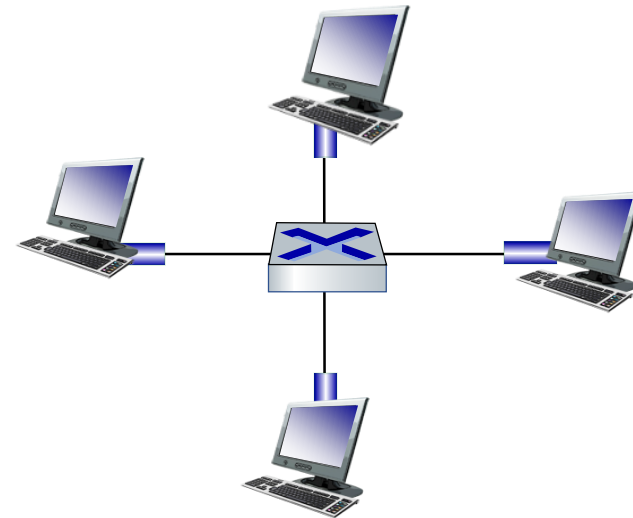
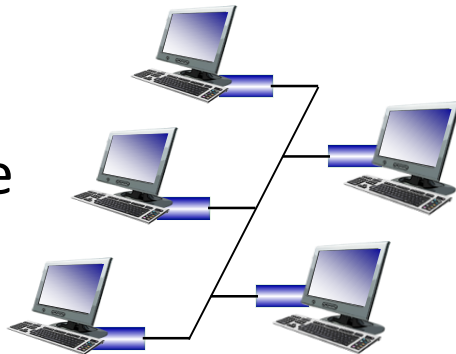


*Metcalfe's Ethernet sketch*

# Ethernet: physical topology

- **bus:** popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- **switched:** prevails today
  - active link-layer 2 *switch* in center
  - each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)

**bus:** coaxial cable



**switched**

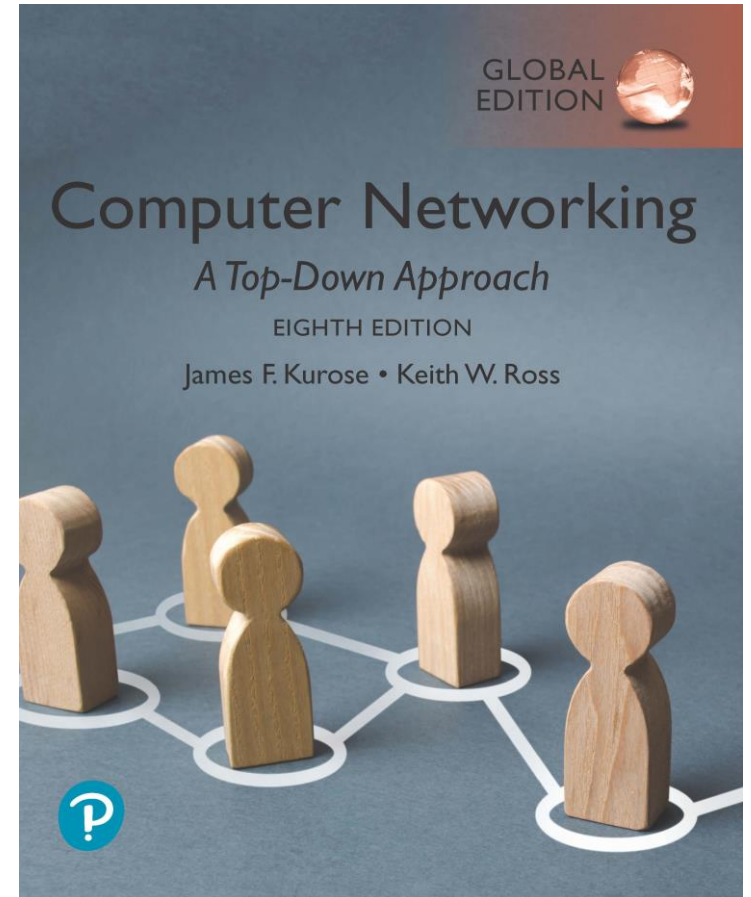
# Ethernet: unreliable, connectionless

- **connectionless**: no handshaking between sending and receiving NICs
- **unreliable**: receiving NIC doesn't send ACKs or NAKs to sending NIC
  - data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: unslotted **CSMA/CD with binary backoff**



# Chapter 7

## Wireless and Mobile Networks



### *Computer Networking: A Top-Down Approach*

8<sup>th</sup> Edition, Global Edition

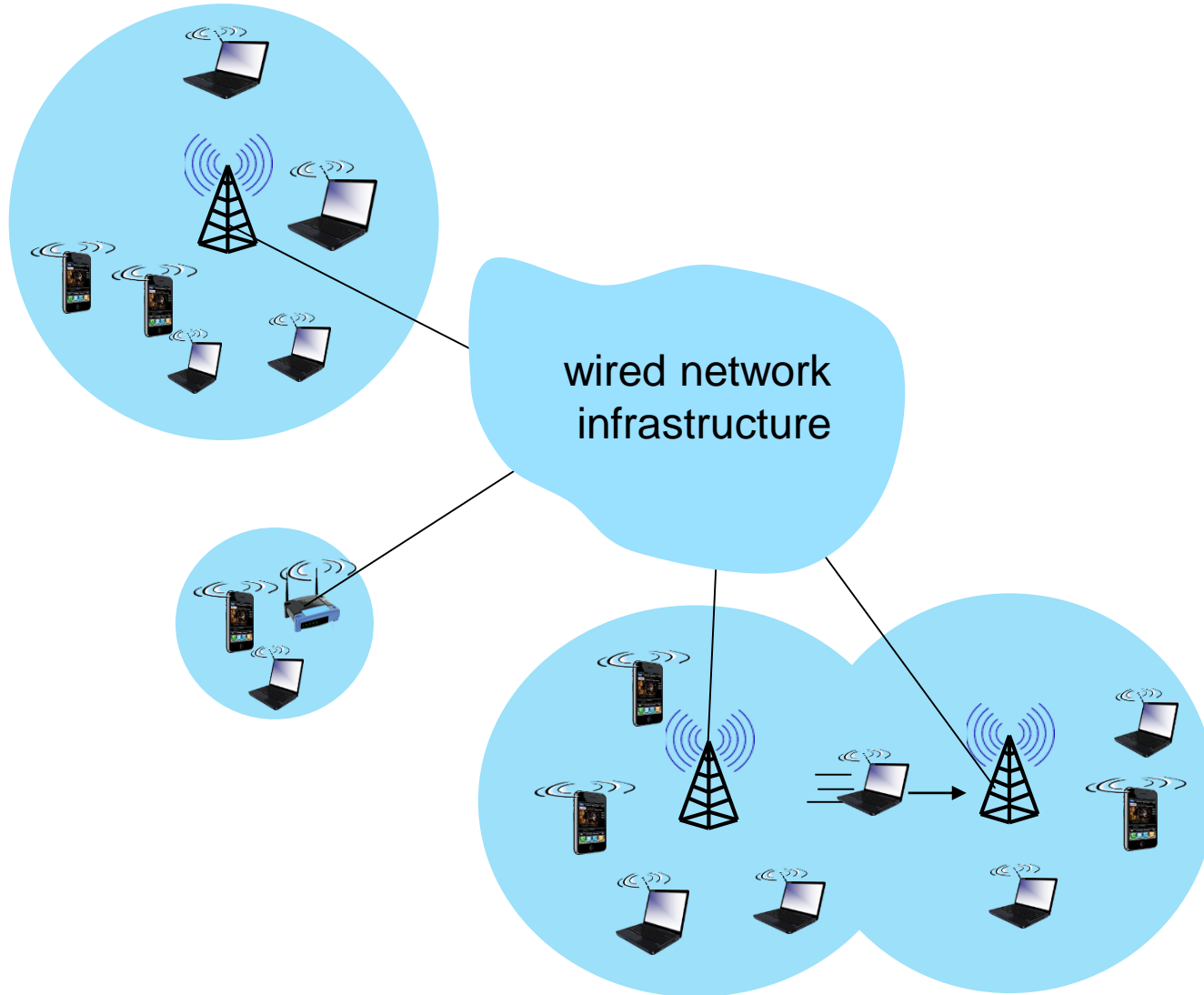
Jim Kurose, Keith Ross

Copyright © 2022 Pearson Education Ltd

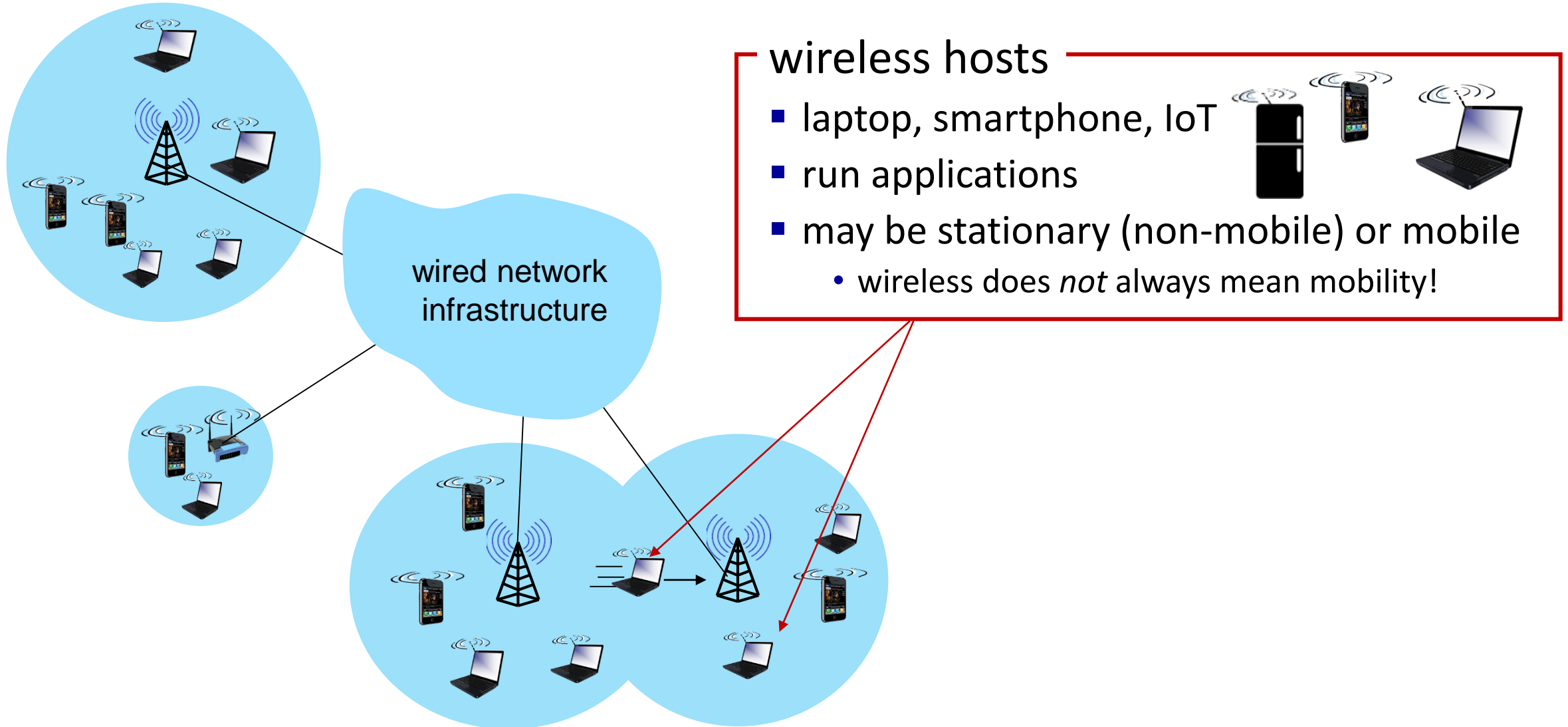
# Wireless and Mobile Networks: context

- more wireless (mobile) phone subscribers than fixed (wired) phone subscribers (10-to-1 in 2019)!
- more mobile-broadband-connected devices than fixed-broadband-connected devices (5-1 in 2019)!
  - 4G/5G cellular networks now embracing Internet protocol stack, including SDN
- two important (but different) challenges
  - **wireless**: communication over wireless link
  - **mobility**: handling the mobile user who changes point of attachment to network

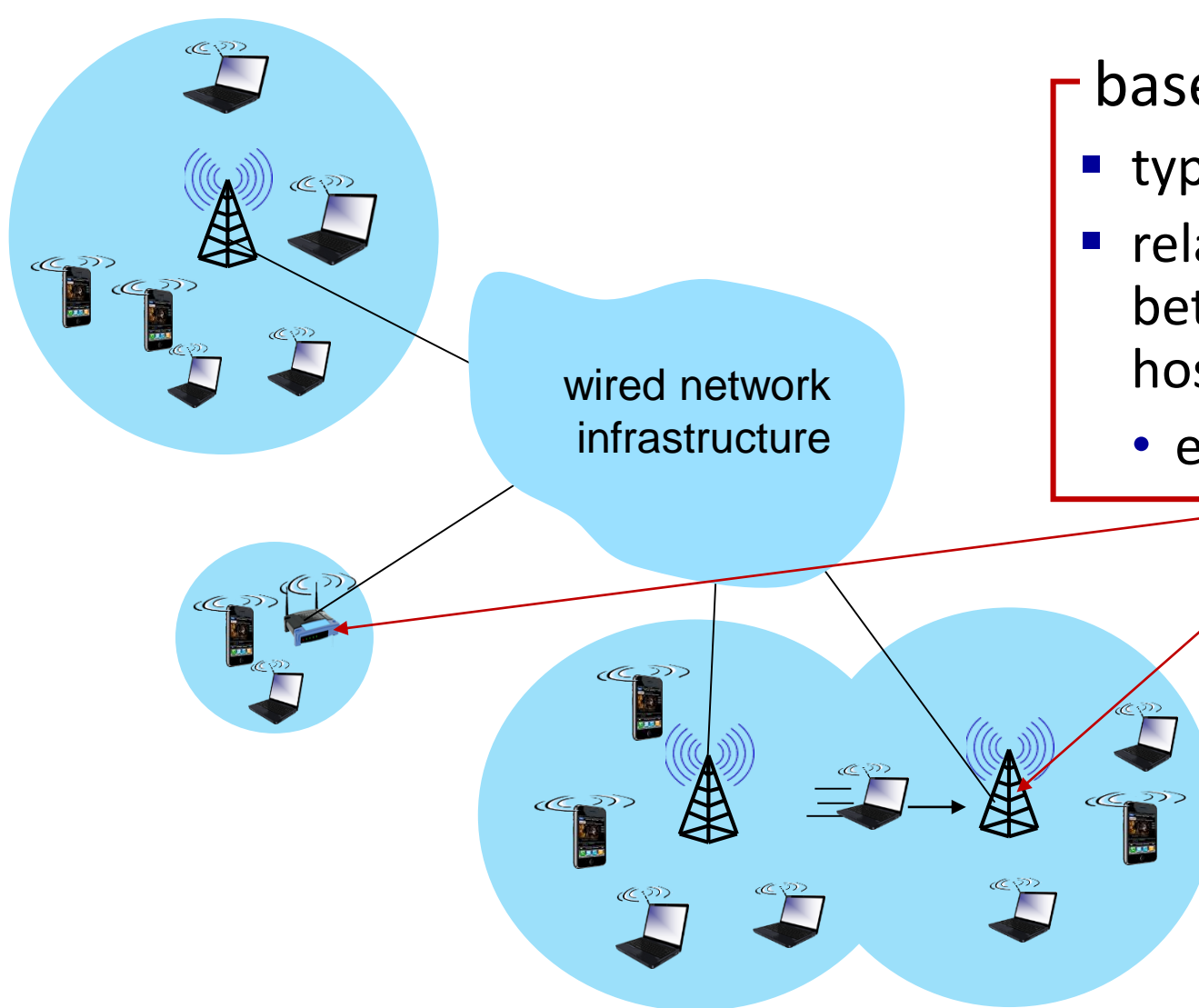
# Elements of a wireless network



# Elements of a wireless network



# Elements of a wireless network

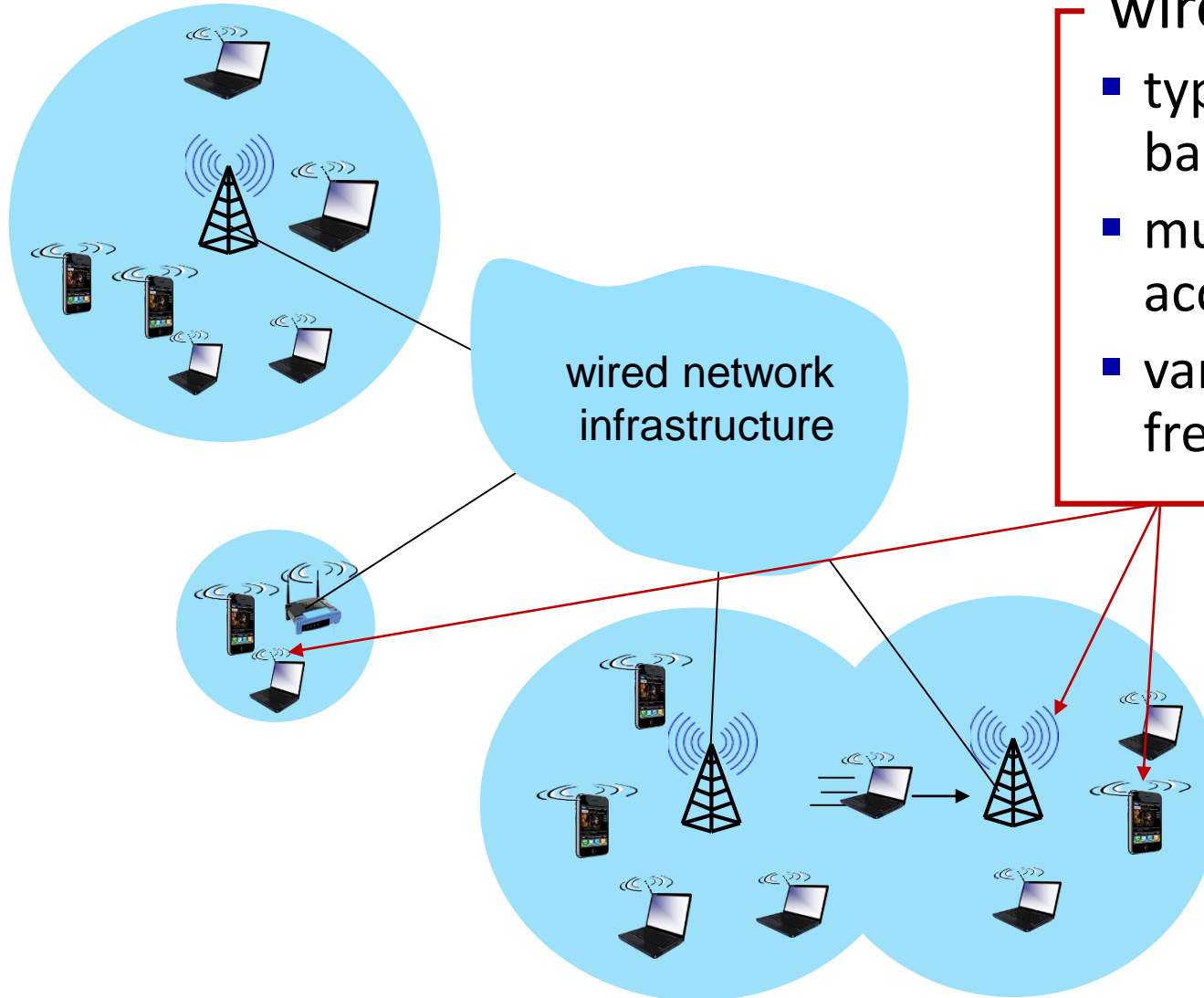


base station



- typically connected to wired network
- relay - responsible for sending packets between wired network and wireless host(s) in its “area”
  - e.g., cell towers, 802.11 access points

# Elements of a wireless network

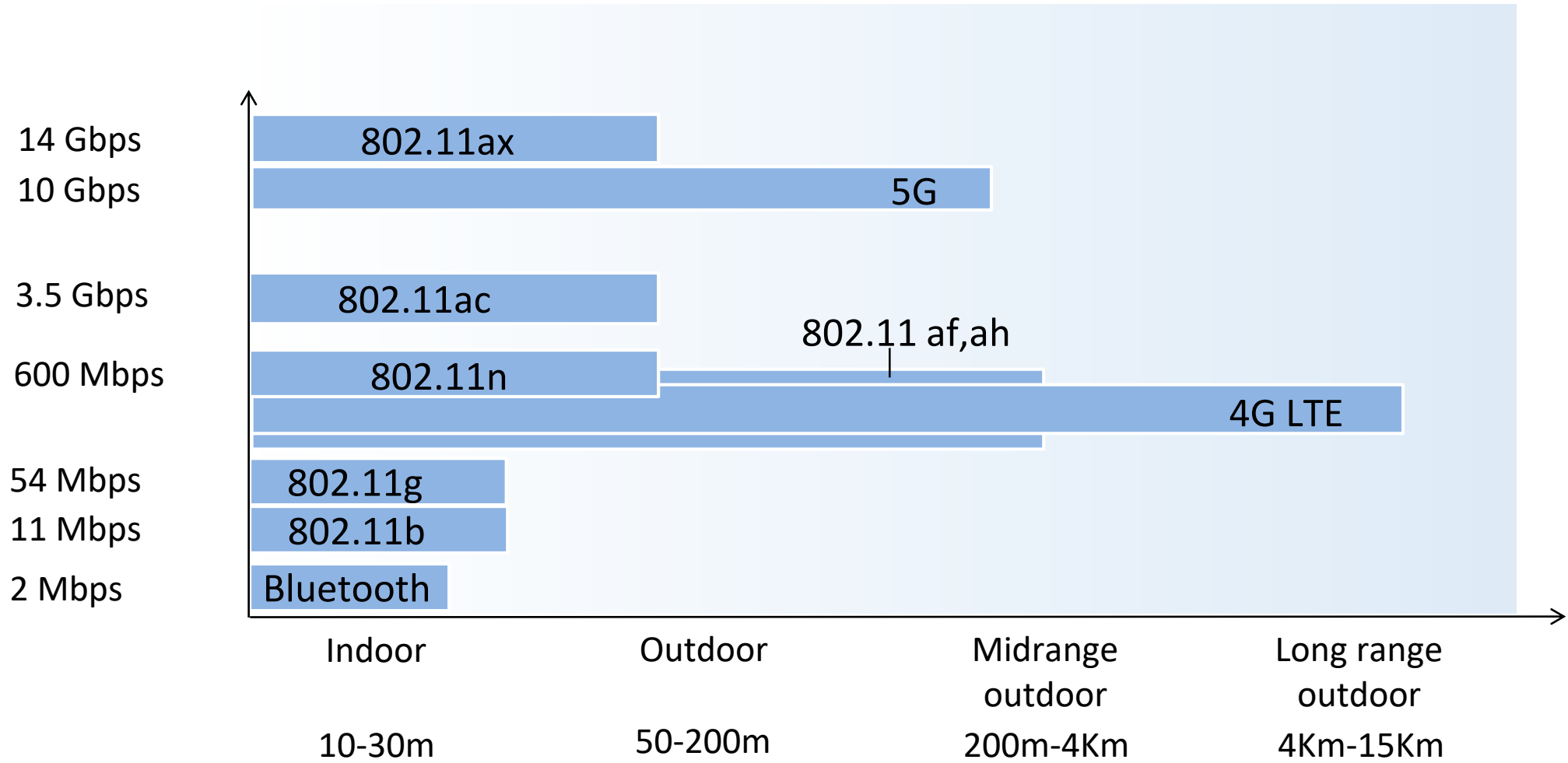


wireless link

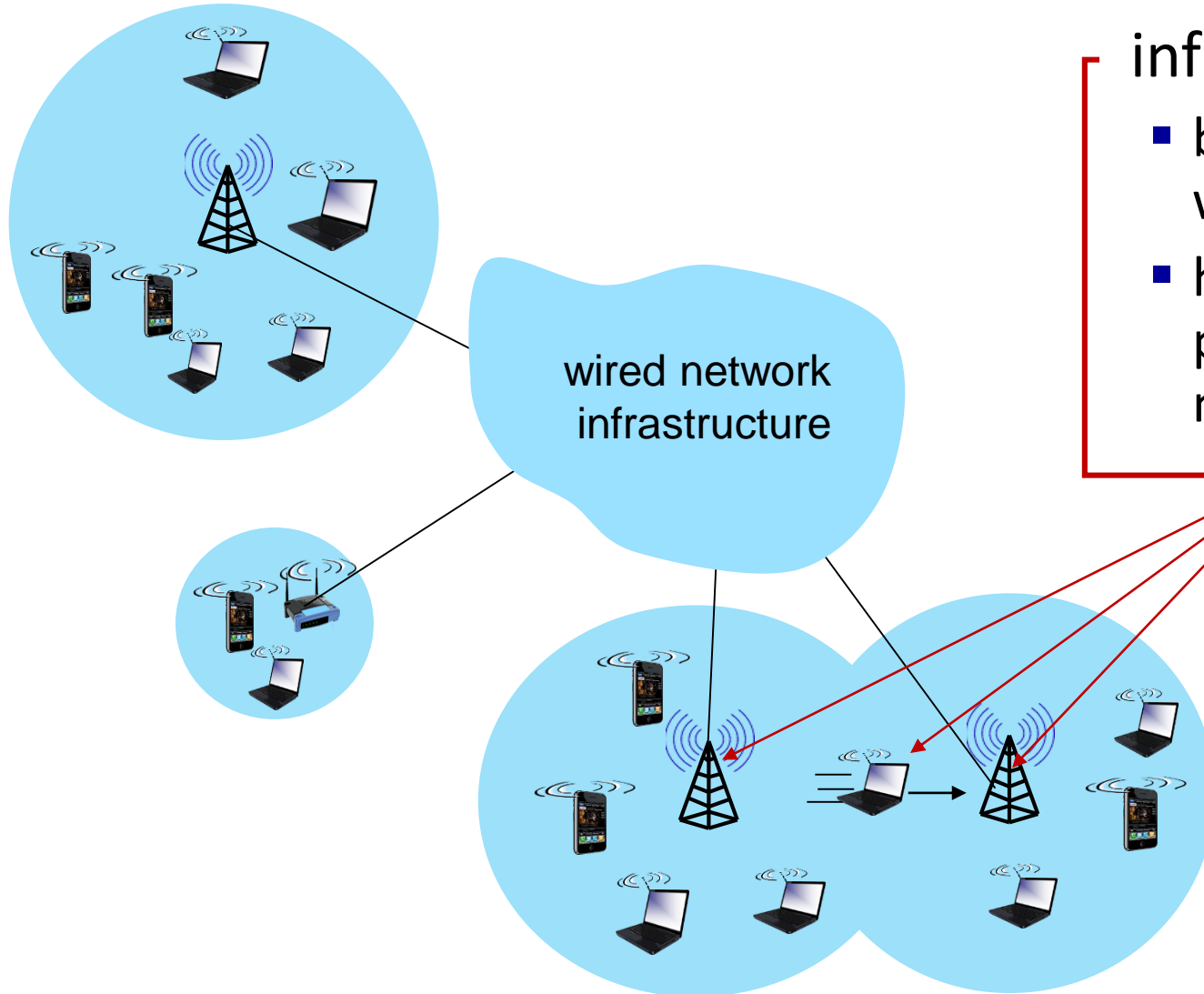


- typically used to connect mobile(s) to base station, also used as backbone link
- multiple access protocol coordinates link access
- various transmission rates and distances, frequency bands

# Characteristics of selected wireless links



# Elements of a wireless network

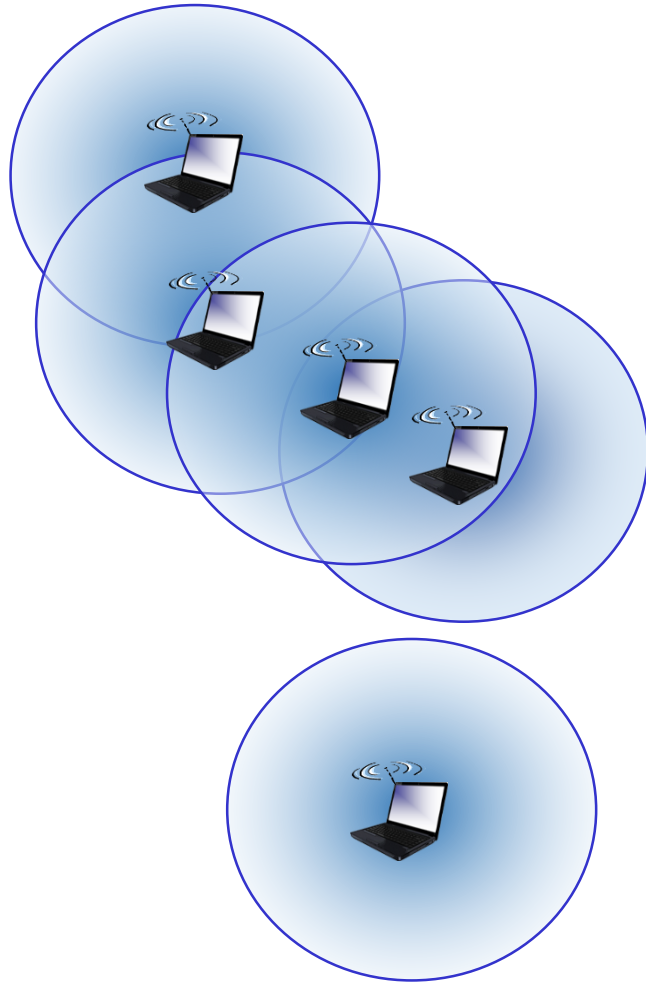


## infrastructure mode

- base station connects mobiles into wired network
- handoff: mobile changes base station providing connection into wired network



# Elements of a wireless network



## ad hoc mode

- no base stations
- nodes can only transmit to other nodes within link coverage
- nodes organize themselves into a network: route among themselves

# Wireless network taxonomy

	single hop	multiple hops
infrastructure (e.g., APs)	host connects to base station (WiFi, cellular) which connects to larger Internet	host may have to relay through several wireless nodes to connect to larger Internet: <i>mesh net</i>
<i>no infrastructure</i>	no base station, no connection to larger Internet (Bluetooth, ad hoc nets)	no base station, no connection to larger Internet. May have to relay to reach other a given wireless node MANET, VANET

# Wireless link characteristics

*important* differences from wired link ....

- **decreased signal strength:** radio signal attenuates as it propagates through matter (path loss)
- **interference from other sources:** wireless network frequencies (e.g., 2.4 GHz) shared by many devices (e.g., WiFi, cellular, motors): interference
- **multipath propagation:** radio signal reflects off objects ground, arriving at destination at slightly different times

.... make communication across (even a point to point) wireless link much more “difficult”

