

# ***Data Link Layer – Part II***

*Computer Networks, Winter 2012/2013*

# Link Layer

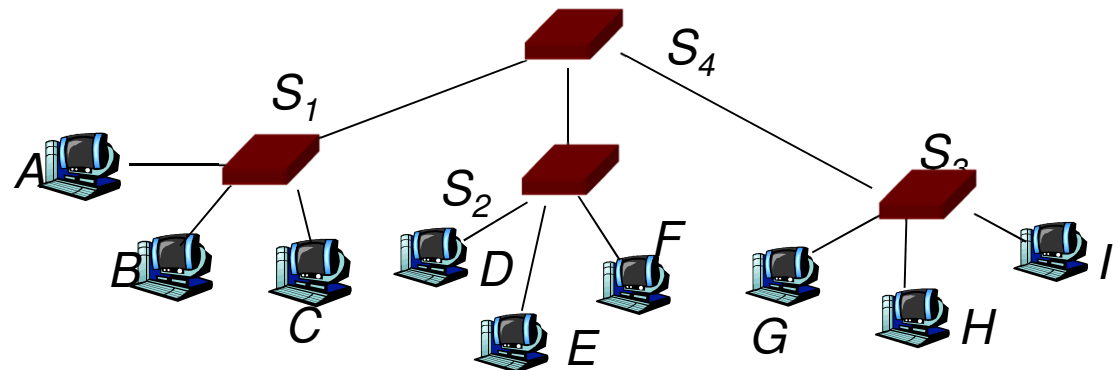
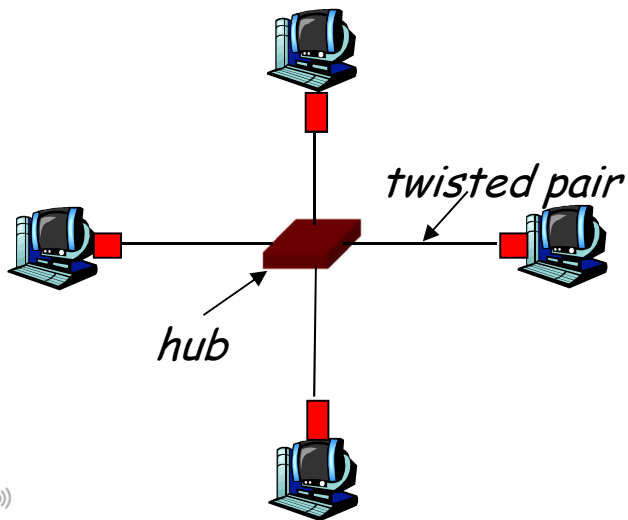
- *5.1 Introduction and services*
- *5.2 Error detection and correction*
- *5.3 Multiple access protocols*
- *5.4 Link-layer Addressing*
- *5.5 Ethernet*
- *5.6 Link-layer switches*
- *5.7 PPP*
- *5.8 Wireless links / Wi-Fi*
- *5.9 Link Virtualization: ATM, MPLS*

# Hubs

... physical-layer (“dumb”) repeaters:

- bits coming in one link go out **all** other links at same rate
- all nodes connected to hub can collide with one another
- no frame buffering
- no CSMA/CD at hub: host NICs detect collisions

Don't really exist anymore

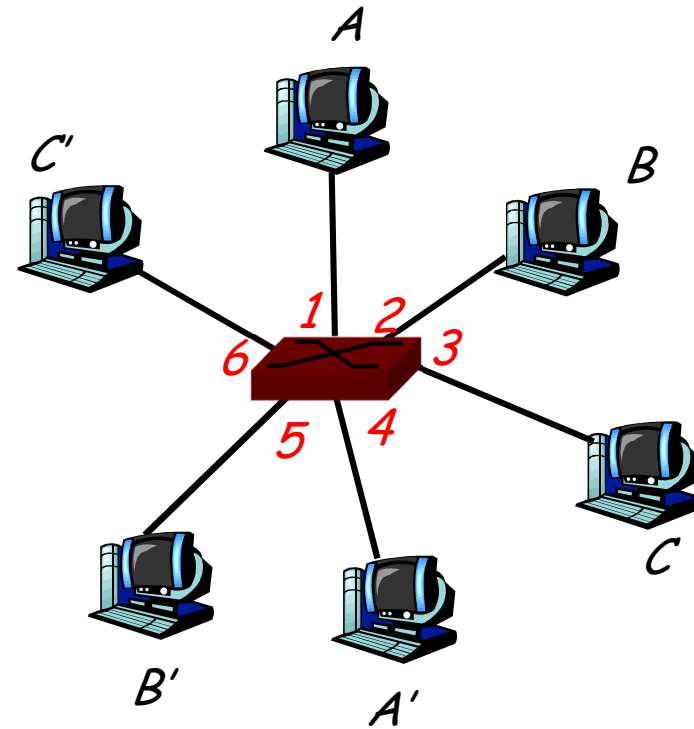


# Bridge and Switch

- *link-layer device: smarter than hubs, take active role*
  - *store, forward Ethernet frames*
  - *examine incoming frame's MAC address, **selectively** forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment*
- **plug-and-play, self-learning**
  - *switches do not need to be configured*
- **transparent**
  - *hosts are unaware of presence of switches*

# Switch Table

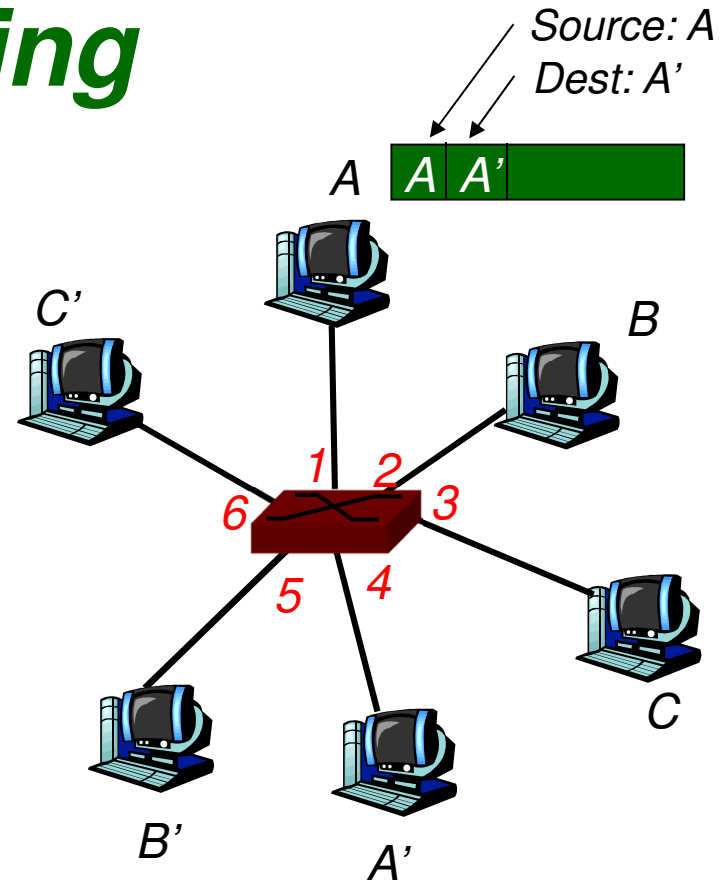
- **Q:** how does switch know that A' reachable via interface 4, B' reachable via interface 5?
- **A:** each switch has a **switch table**, each entry:
  - (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!
- **Q:** how are entries created, maintained in switch table?
  - something like a routing protocol?



switch with six interfaces  
(1,2,3,4,5,6)

# Switch: self-learning

- switch **learns** which hosts can be reached through which interfaces
  - when frame received, switch “learns” location of sender: incoming LAN segment
  - records sender/location pair in switch table

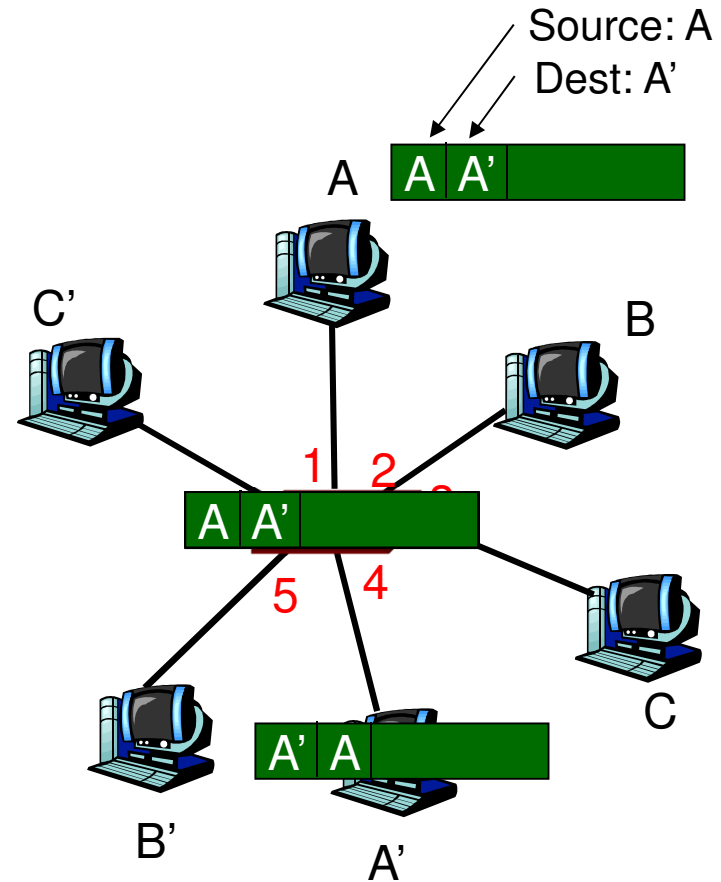


MAC addr	interface	TTL
A	1	60

Switch table  
(initially empty)

# Self-learning, forwarding: example

- frame destination unknown: *flood*
- destination A location known: *selective send*



MAC addr	interface	TTL
A	1	60
A'	4	61

Switch table  
(initially empty)

# Switch: frame filtering/forwarding

When frame received:

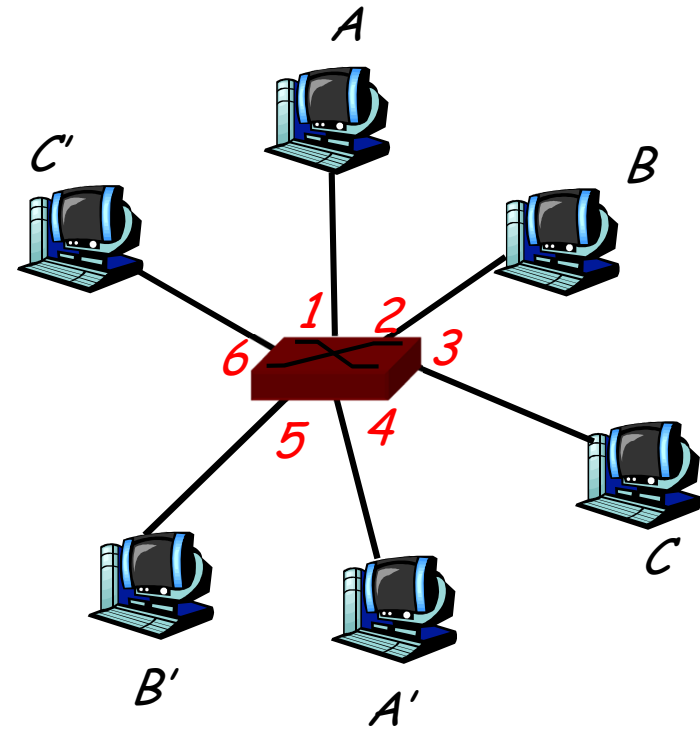
1. Add a new entry into the table with source address, interface number and current time. ( step 2)
2. Search the destination address in the switch table
3. **if** entry found for destination  
**then** {  
**if** dest on segment from which frame arrived  
**then** drop the frame(**filtering**)  
**else** forward the frame on interface indicated(**forwarding**/ step 4)  
}  
**else** flood ( step 3)

forward on all but the interface on which the frame arrived



# Switch: allows multiple simultaneous transmissions

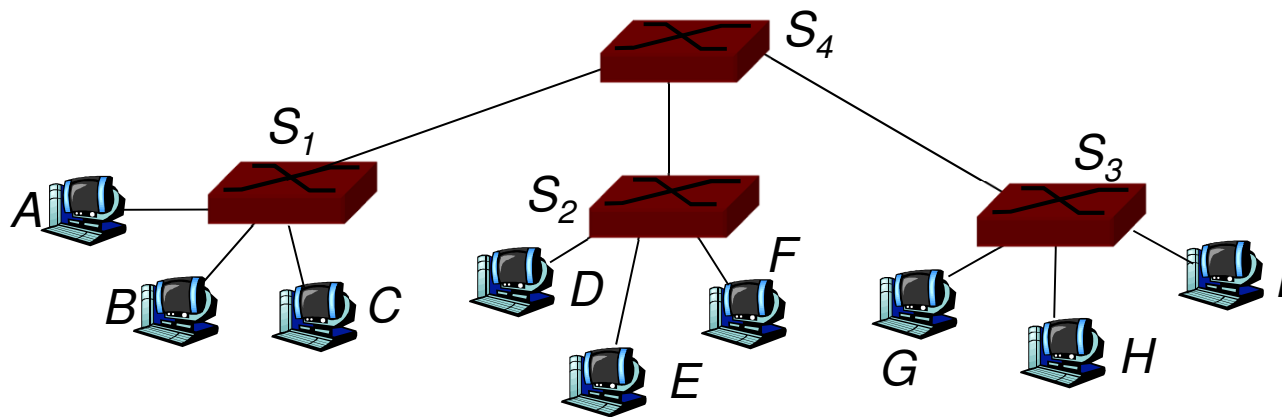
- *hosts have dedicated, direct connection to switch*
- *switches buffer packets*
- *Ethernet protocol used on each incoming link, but no collisions; full duplex*
  - *each link is its own collision domain*
- **switching:** *A-to-A' and B-to-B' simultaneously, without collisions*
  - *not possible with dumb hub*



switch with six interfaces  
(1,2,3,4,5,6)

# Interconnecting switches

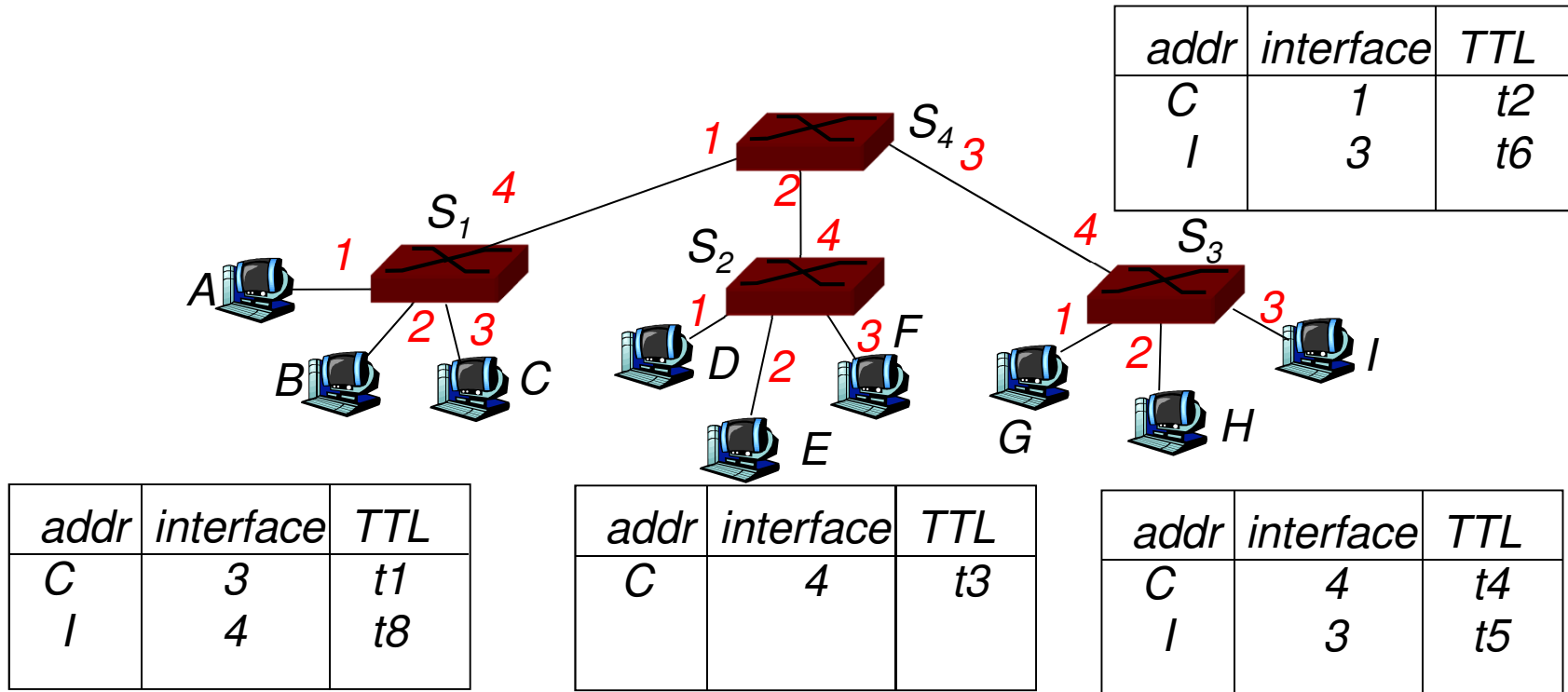
- switches can be connected together



- **Q:** sending from A to G - how does S<sub>1</sub> know to forward frame destined to F via S<sub>4</sub> and S<sub>3</sub>?
- **A:** self learning! (works exactly the same as in single-switch case!)

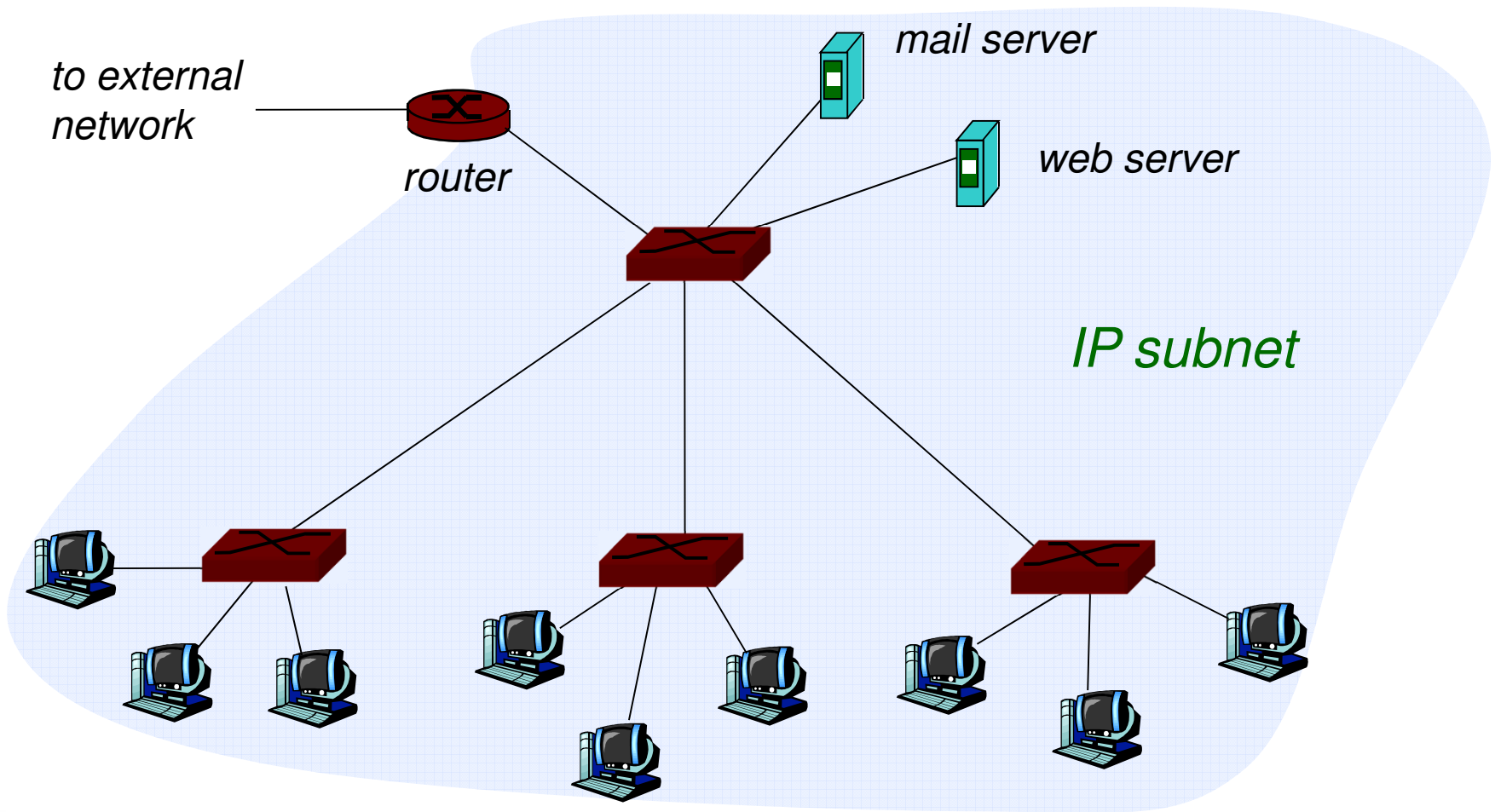
# Self-learning multi-switch example

Suppose C sends frame to I, I responds to C



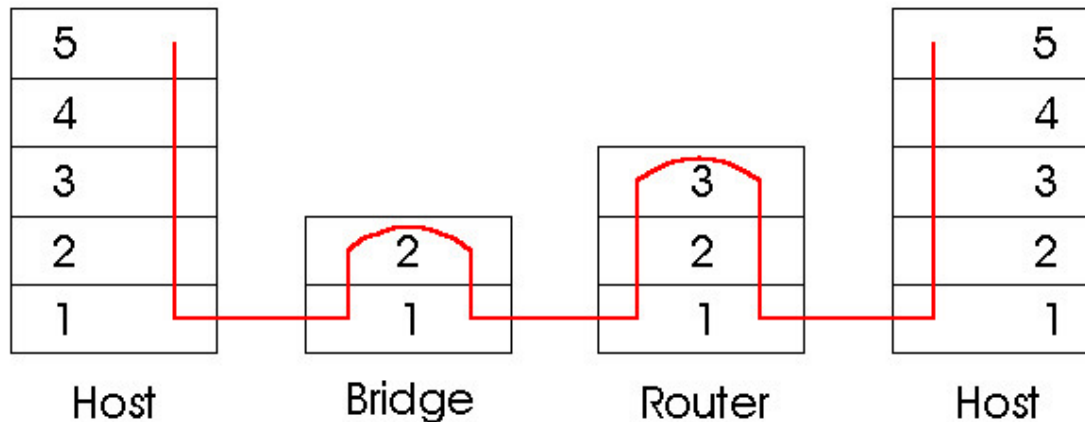
- Q: show switch tables and packet forwarding in S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>

# *Institutional network (e.g. GöNet)*



# Switches vs. Routers

- *both store-and-forward devices*
  - *routers: network layer devices (examine network layer headers)*
  - *switches are link layer devices*
- *routers maintain routing tables, implement routing algorithms*
- *switches maintain switch tables, implement filtering, learning algorithms*



# Link Layer

- *5.1 Introduction and services*
- *5.2 Error detection and correction*
- *5.3 Multiple access protocols*
- *5.4 Link-Layer Addressing*
- *5.5 Ethernet*
- *5.6 Link-layer switches*
- *5.7 PPP*
- *5.8 Wireless links / Wi-Fi*
- *5.9 Link Virtualization: ATM, MPLS*

# *Point to Point Data Link Control*

- *one sender, one receiver, one link: easier than broadcast link:*
  - *no Multiple Access Protocol*
  - *no need for explicit MAC addressing*
  - *e.g., dialup link, ISDN line*
- *popular point-to-point DLC protocols:*
  - *PPP (point-to-point protocol)*
  - *HDLC: High level data link control (Data link used to be considered “high layer” in protocol stack!)*

# PPP Design Requirements [RFC 1557]

- *packet framing*: encapsulation of network-layer datagram in data link frame
  - carry network layer data of any network layer protocol (not just IP) at same time
  - ability to demultiplex upwards
- *bit transparency*: must carry any bit pattern in the data field
- *error detection* (no correction)
- *connection liveness*: detect, signal link failure to network layer
- *network layer address negotiation*: endpoint can learn/configure each other's network address



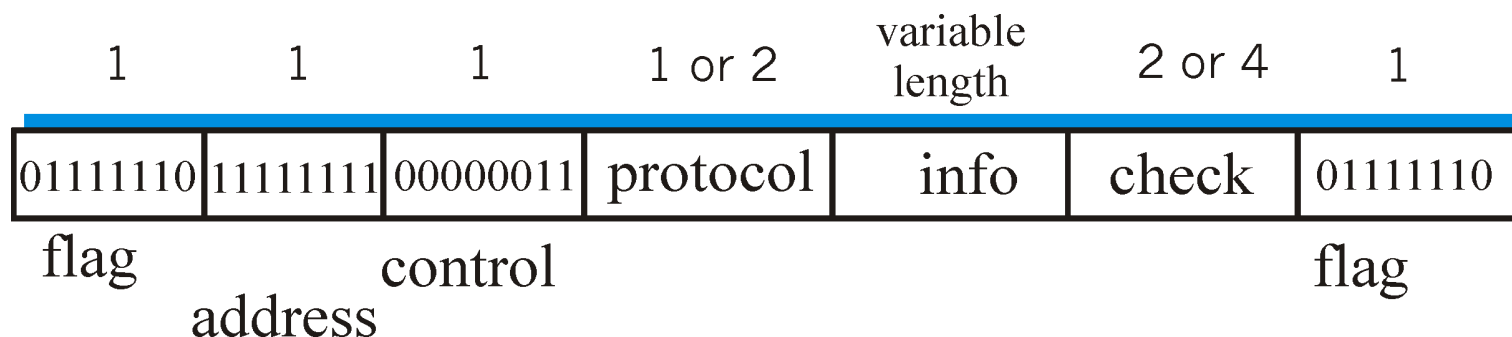
# *PPP non-requirements*

- *no error correction/recovery*
- *no flow control*
- *out of order delivery OK*
- *no need to support multipoint links (e.g., polling)*

*Error recovery, flow control, data re-ordering  
all relegated to higher layers!*

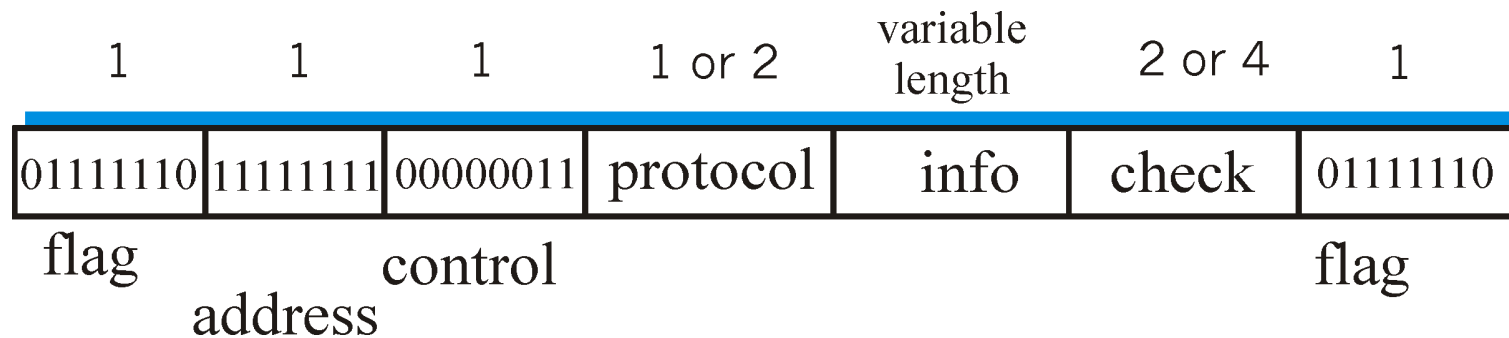
# PPP Data Frame

- *Flag: delimiter (framing)*
- *Address: does nothing (only one option)*
- *Control: does nothing; in the future possible multiple control fields*
- *Protocol: upper layer protocol to which frame delivered (eg, PPP-LCP, IP, IPCP, etc)*



# PPP Data Frame

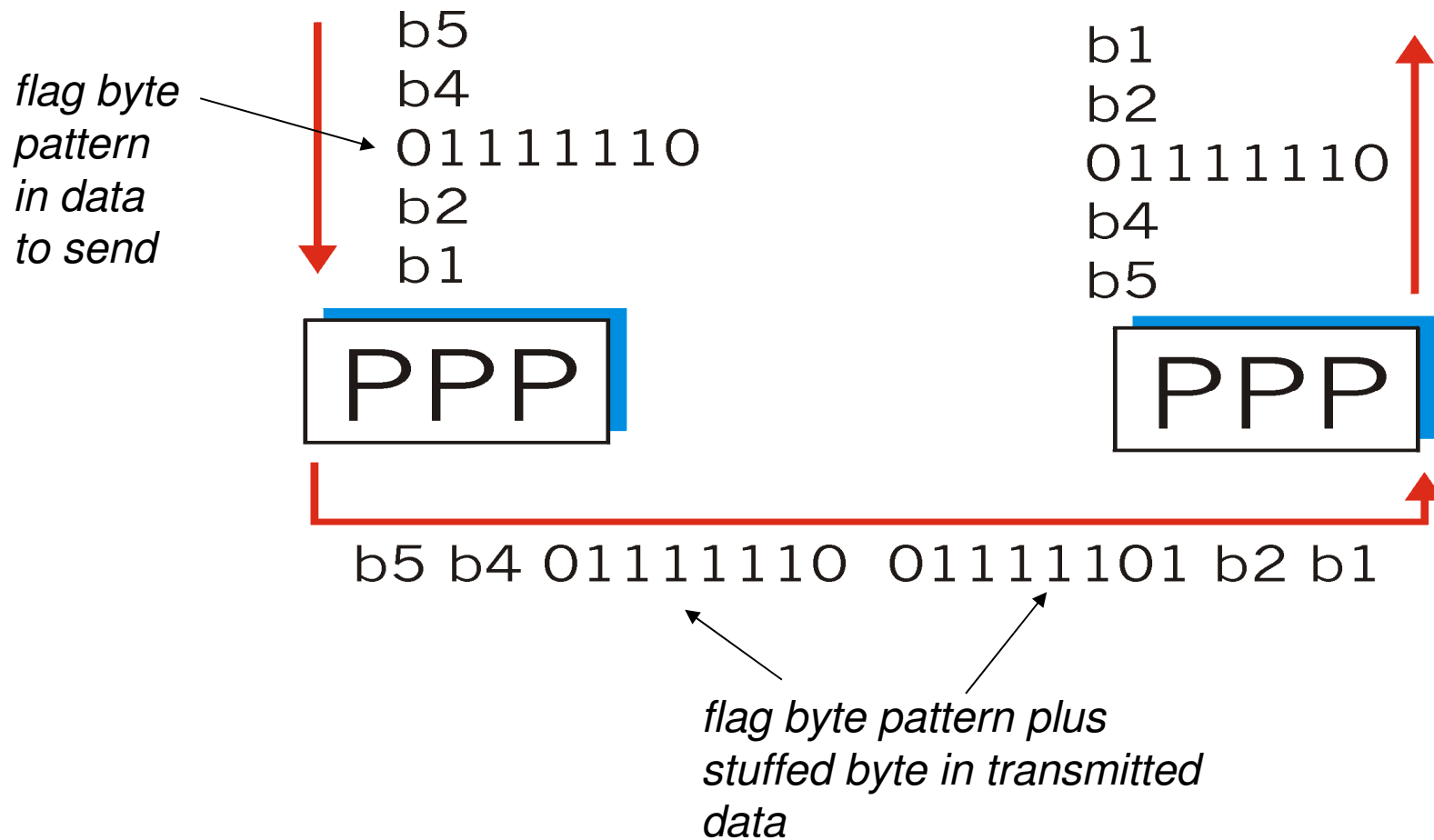
- *info*: upper layer data being carried
- *check*: cyclic redundancy check for error detection



# Byte Stuffing

- “data transparency” requirement: data field must be allowed to include flag pattern  $\langle 01111110 \rangle$ 
  - *Q*: is received  $\langle 01111110 \rangle$  data or flag?
- *Sender*: adds (“stuffs”) extra  $\langle 01111110 \rangle$  byte after each  $\langle 01111110 \rangle$  data byte
- *Receiver*:
  - two  $01111110$  bytes in a row: discard first byte, continue data reception
  - single  $01111110$ : flag byte

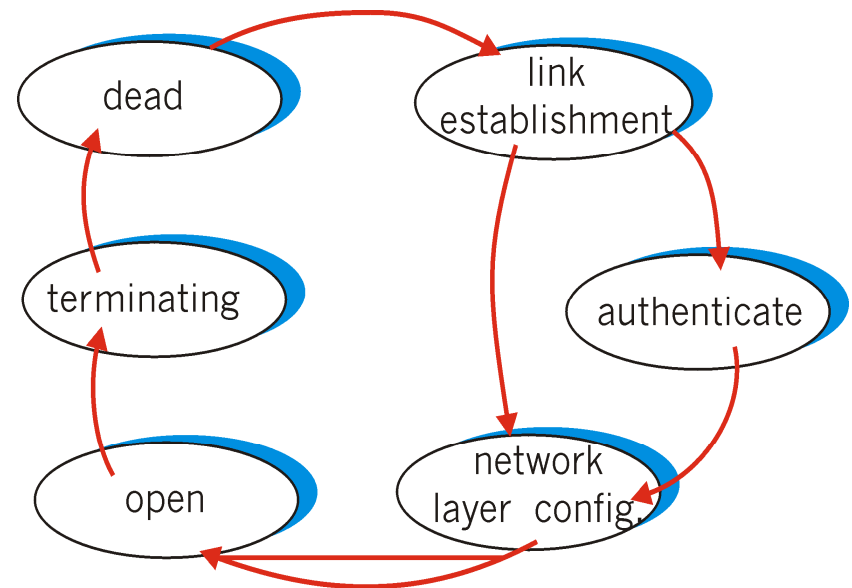
# Byte Stuffing



# PPP Data Control Protocol

Before exchanging network-layer data, data link peers must

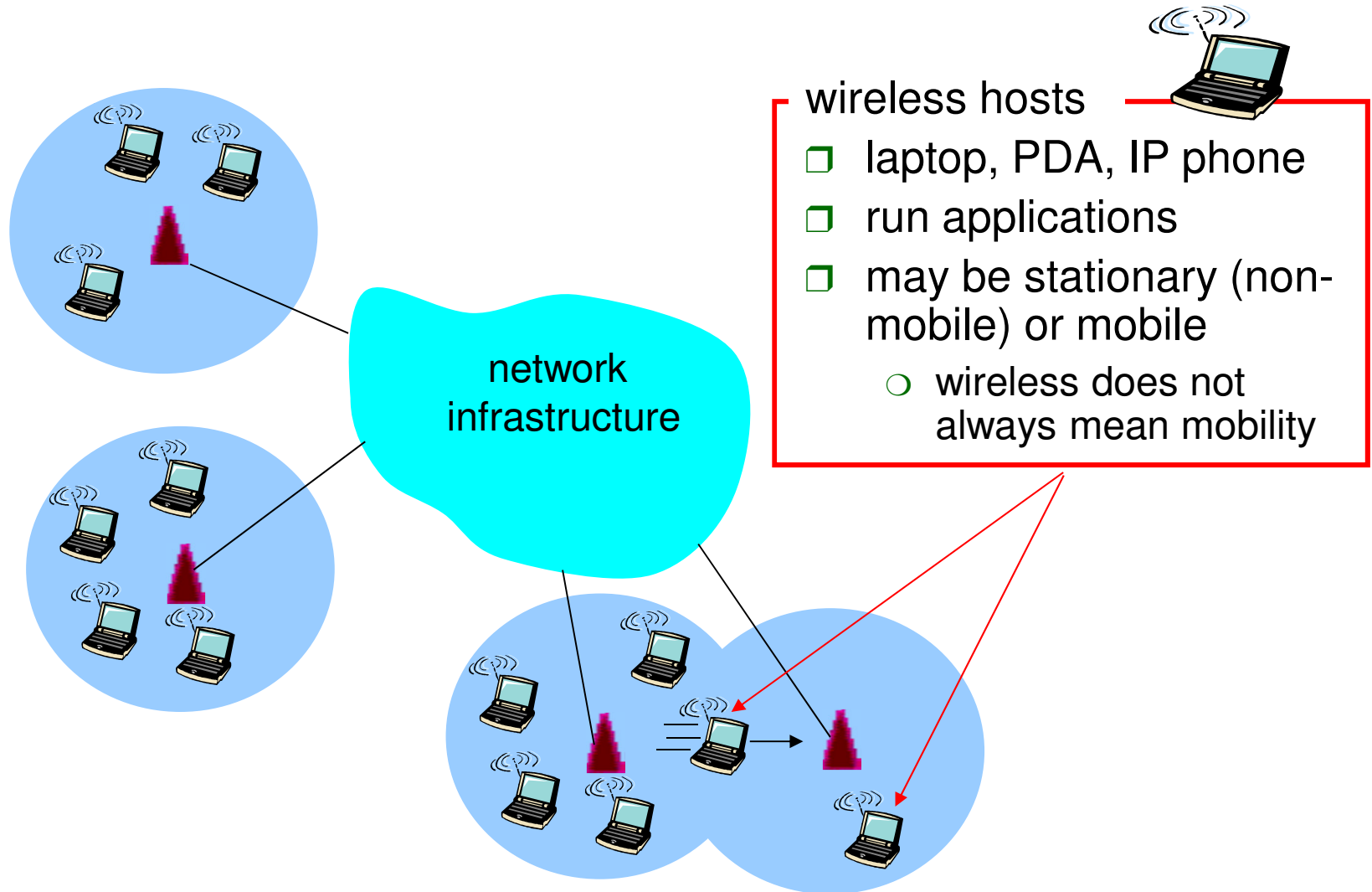
- *configure PPP link (max. frame length, authentication)*
- *learn/configure network layer information*
  - *for IP: carry IP Control Protocol (IPCP) msgs (protocol field: 8021) to configure/learn IP address*



# Link Layer

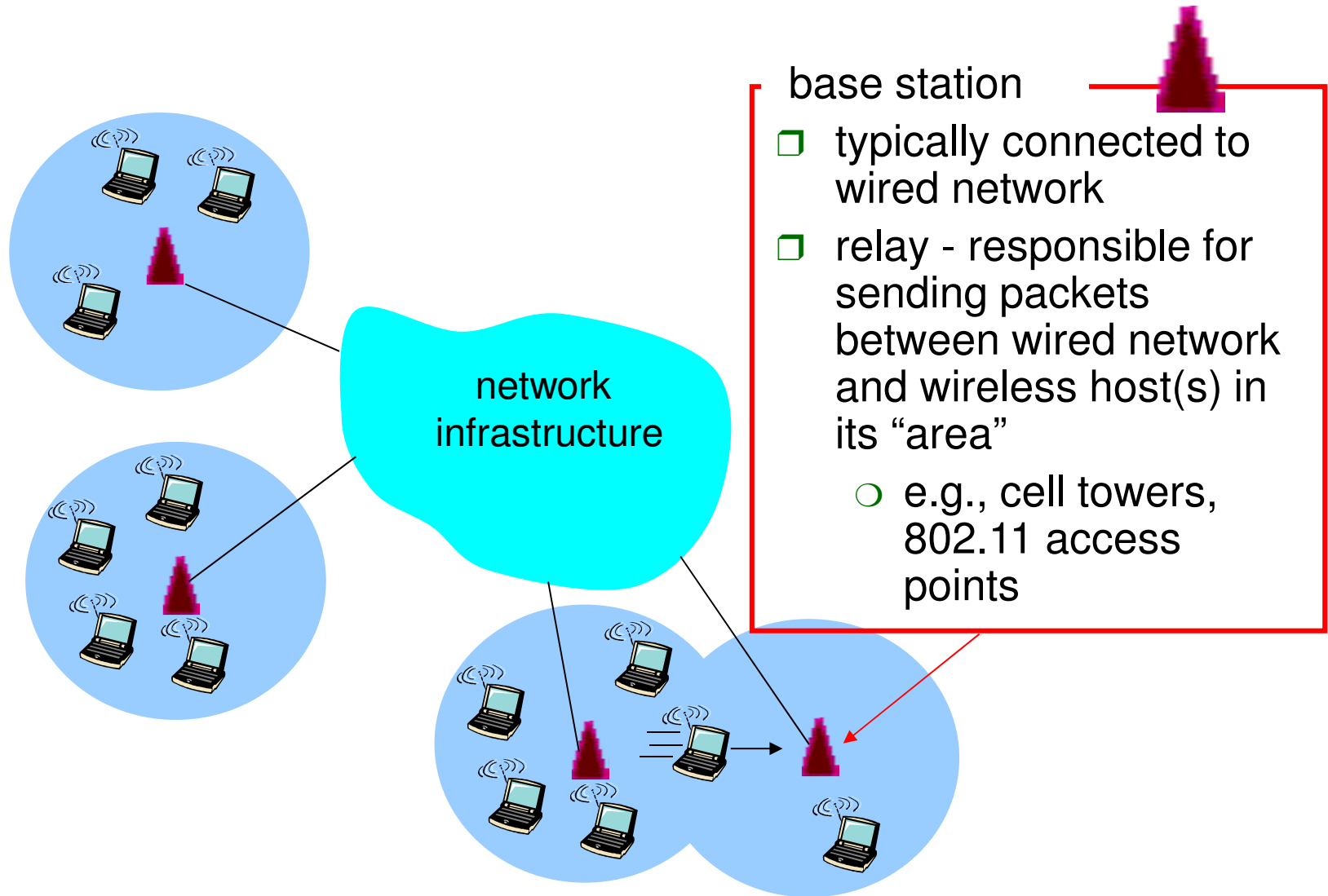
- *5.1 Introduction and services*
- *5.2 Error detection and correction*
- *5.3 Multiple access protocols*
- *5.4 Link-Layer Addressing*
- *5.5 Ethernet*
- *5.6 Link-layer switches*
- *5.7 PPP*
- *5.8 Wireless links / Wi-Fi*
- *5.9 Link Virtualization: ATM, MPLS*

# Elements of a wireless network





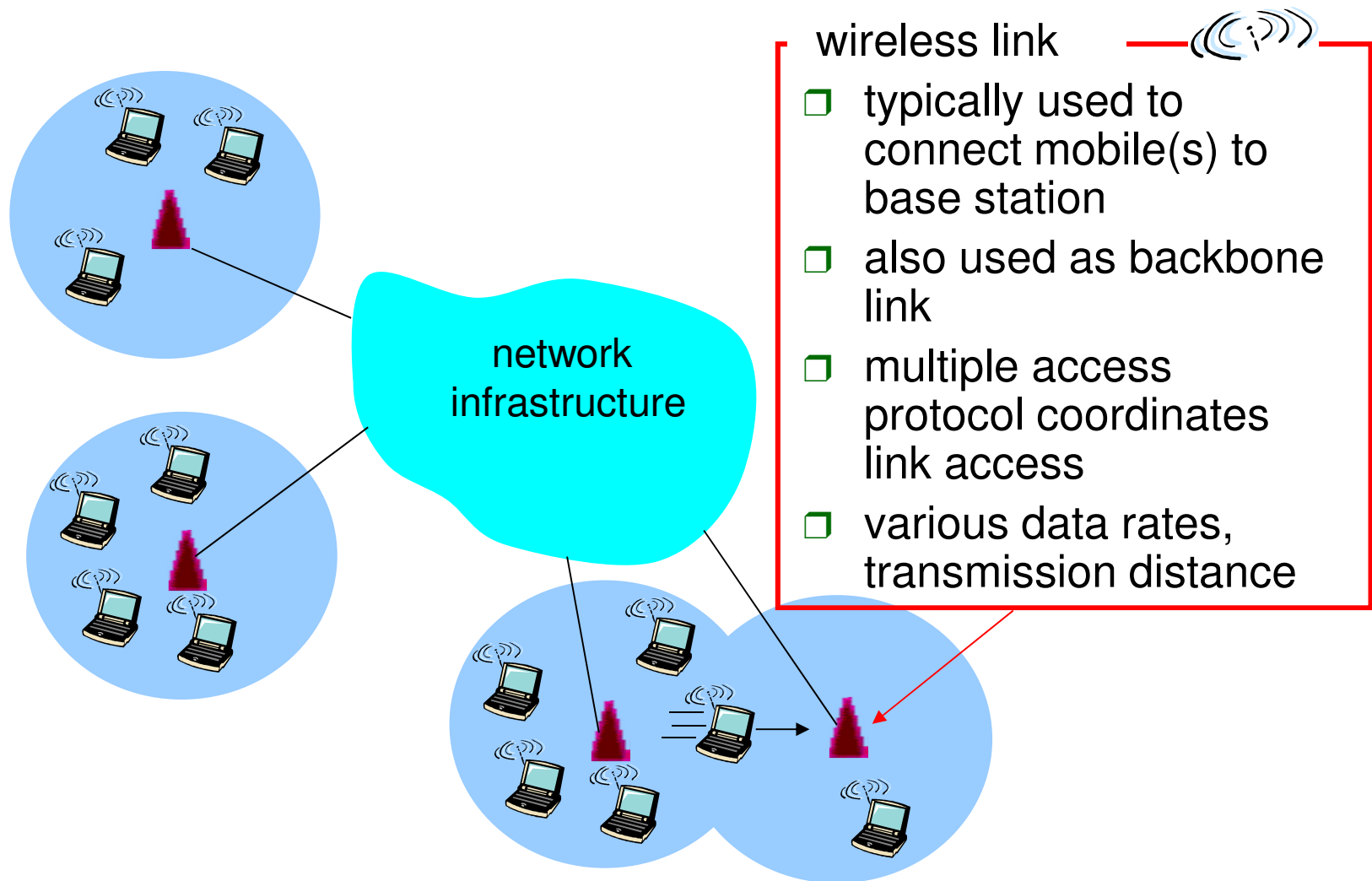
# Elements of a wireless network



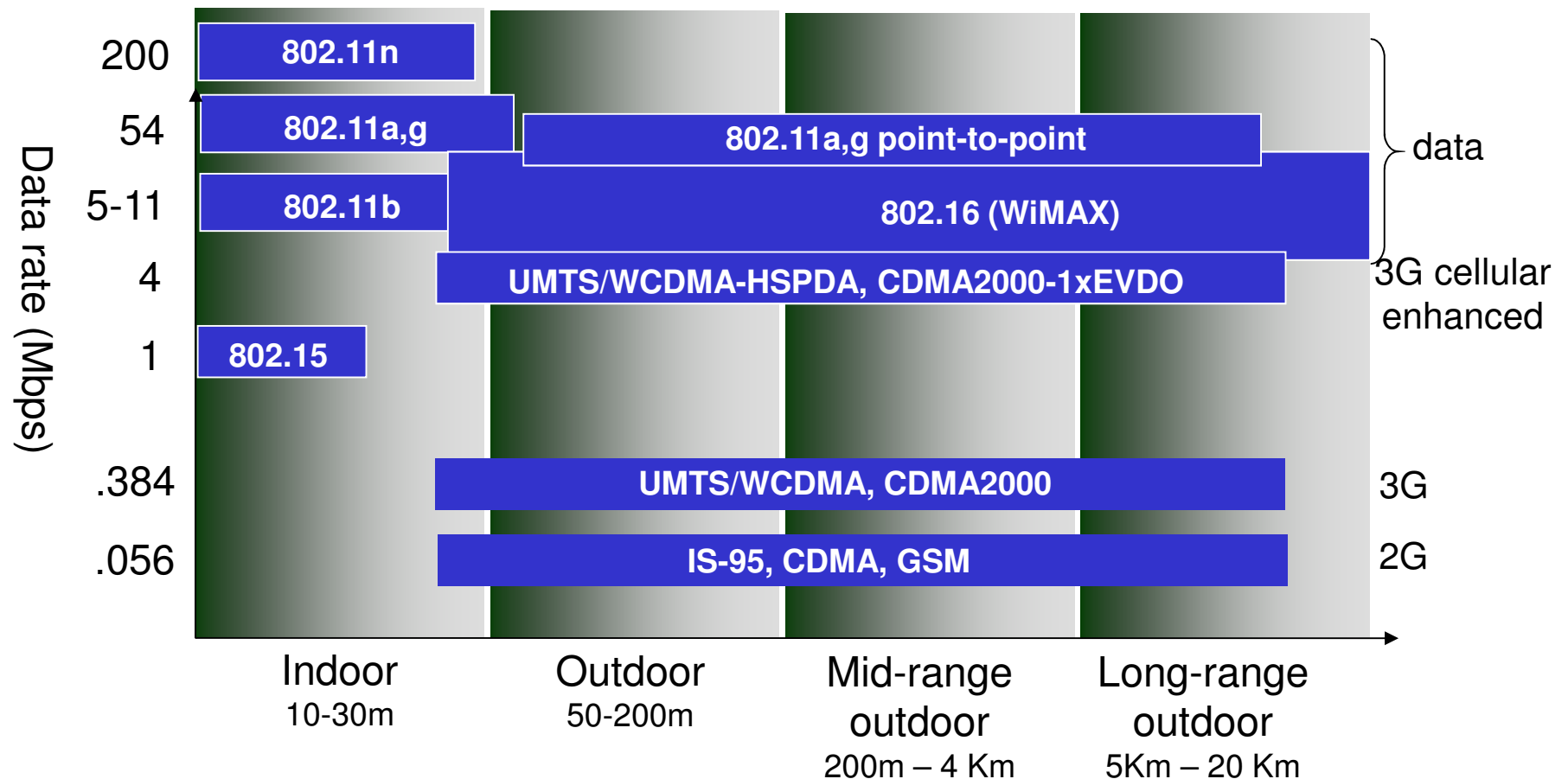
base station

- 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



# Characteristics of selected wireless link standards

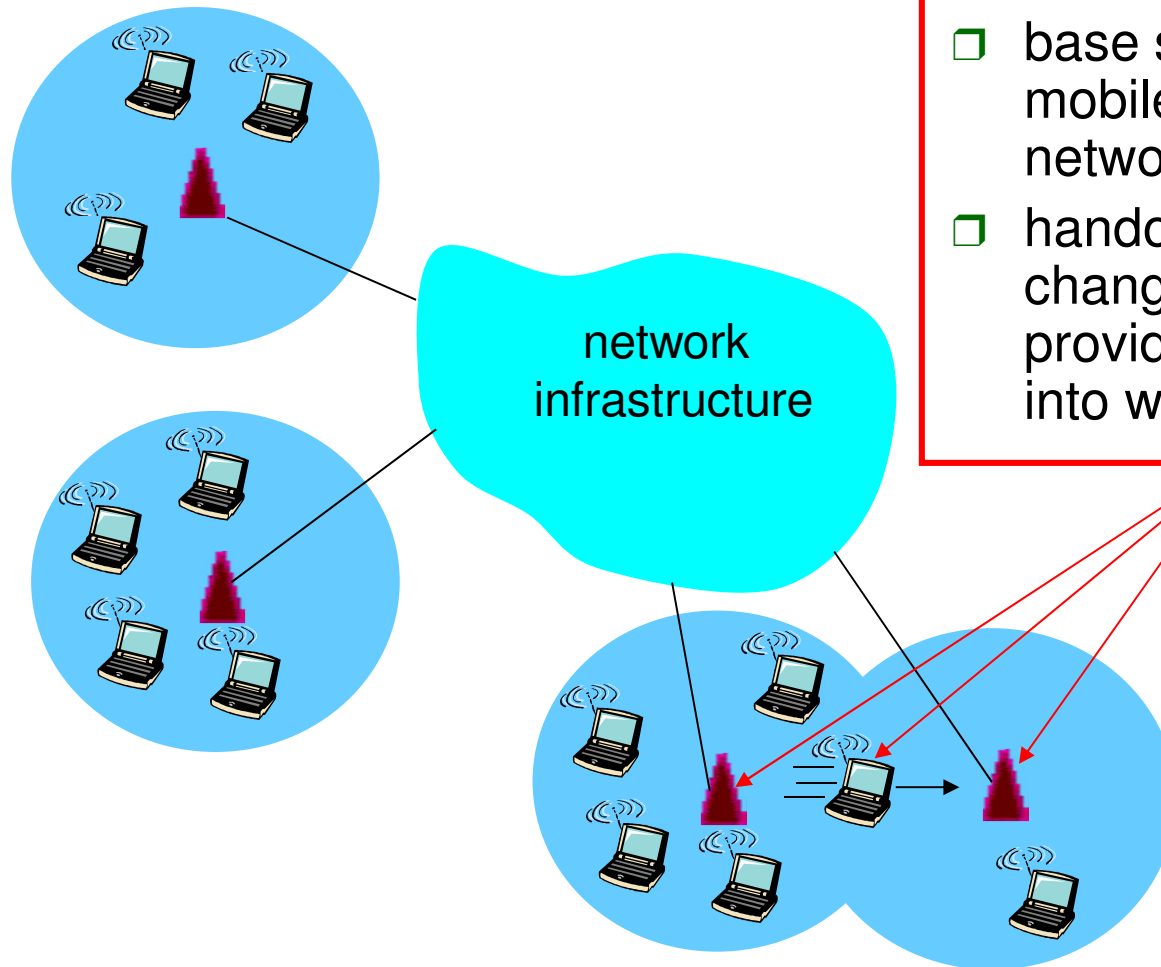


# IEEE 802.11 Wireless LAN

- **802.11b**
  - 2.4-5 GHz unlicensed spectrum
  - up to 11 Mbps
  - direct sequence spread spectrum (DSSS) in physical layer
    - all hosts use same chipping code
- **802.11a**
  - 5-6 GHz range
  - up to 54 Mbps
- **802.11g**
  - 2.4-5 GHz range
  - up to 54 Mbps
- **802.11n: multiple antennae**
  - 2.4-5 GHz range
  - up to 200 Mbps

- 
- all use CSMA/CA for multiple access
  - all have base-station and ad-hoc network versions

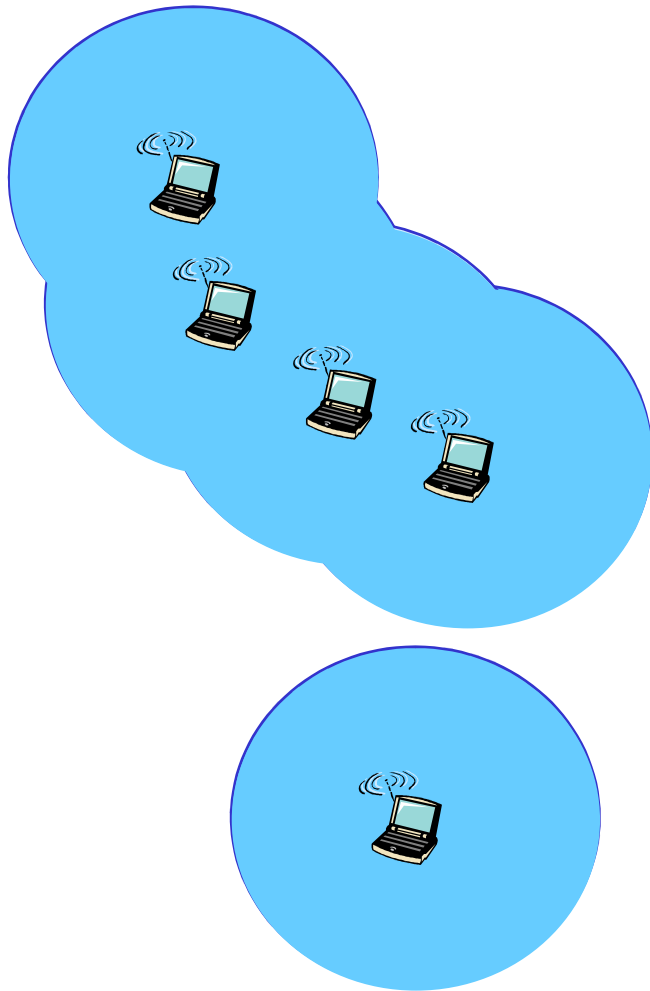
# Elements of a wireless network



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

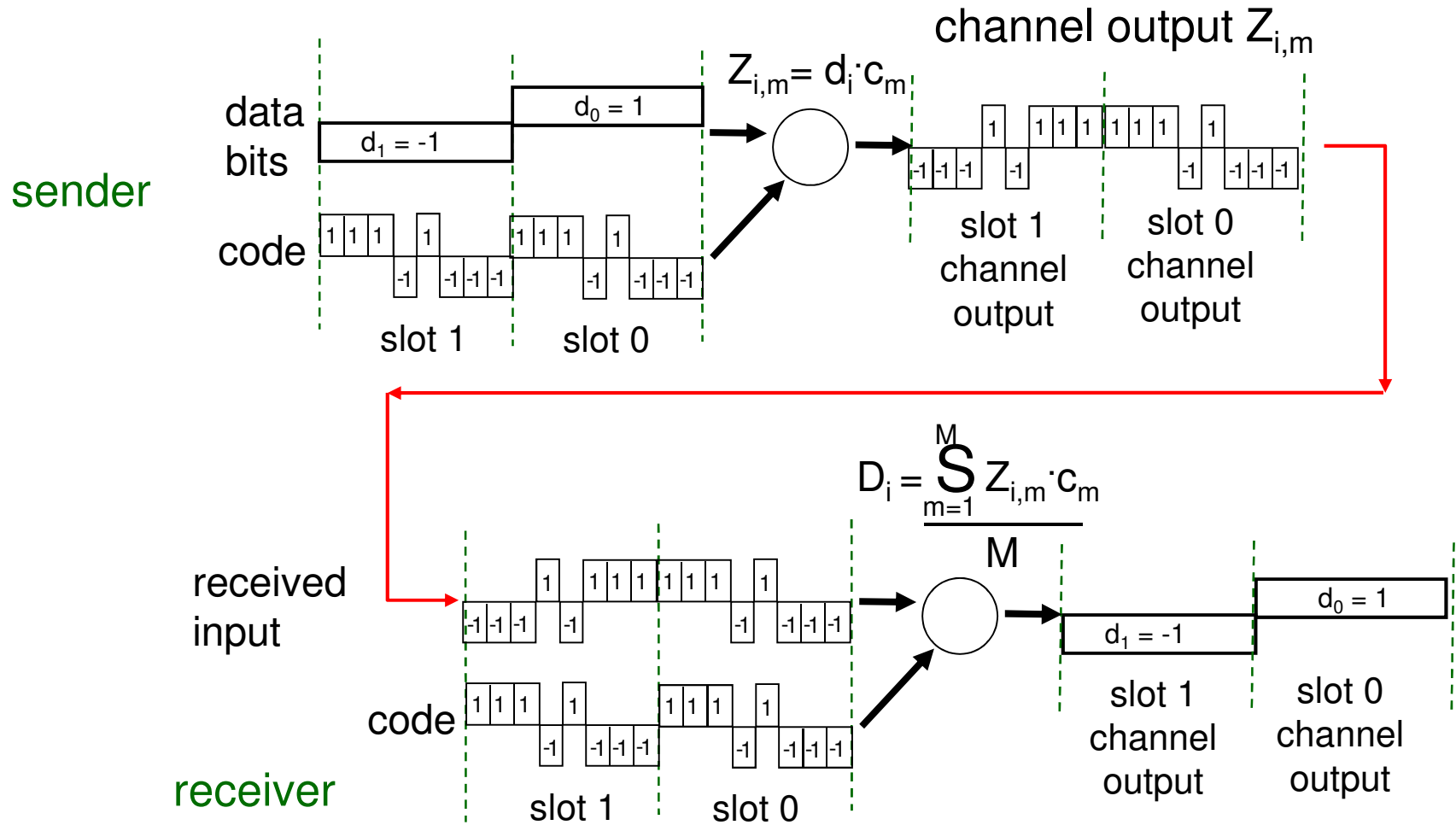
	single hop	multiple hops
infrastructure (e.g., APs)	host connects to base station (WiFi, WiMAX, cellular) which connects to larger Internet	host may have to relay through several wireless nodes to connect to larger Internet: mesh net
no infrastructure	no base station, no connection to larger Internet (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

# Code Division Multiple Access (CDMA)

- *used in several wireless broadcast channels (cellular, satellite, etc) standards*
- *unique “code” assigned to each user; i.e., code set partitioning*
- *all users share same frequency, but each user has own “chipping” sequence (i.e., code) to encode data*
- **encoded signal** = *(original data) X (chipping sequence)*
- **decoding:** *inner-product of encoded signal and chipping sequence*
- *allows multiple users to “coexist” and transmit simultaneously with minimal interference (if codes are “orthogonal”)*

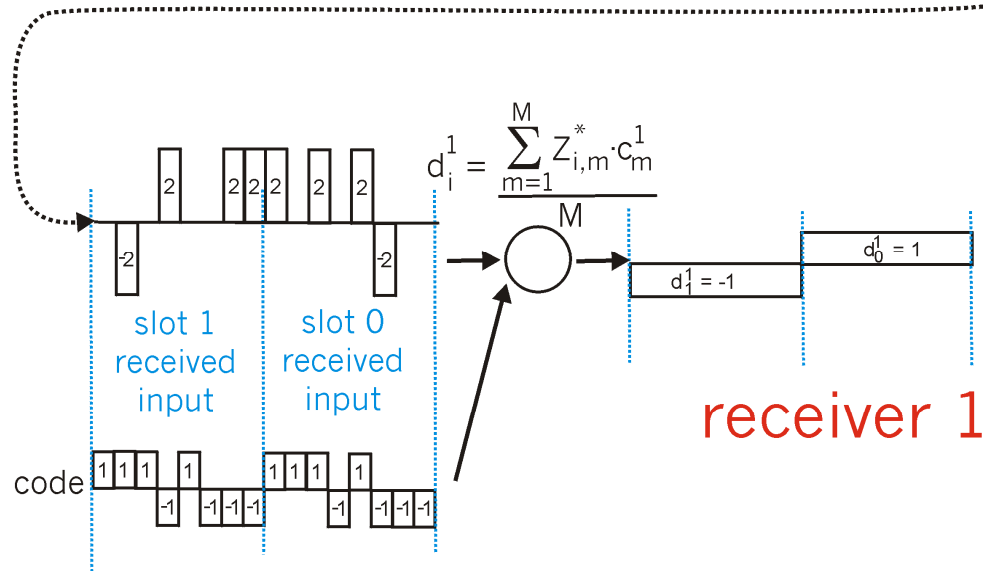
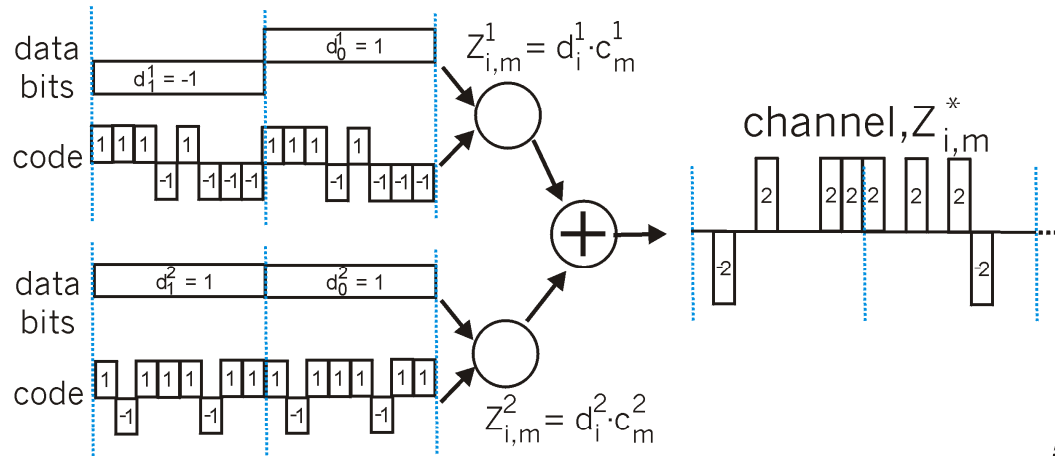


# CDMA Encode/Decode



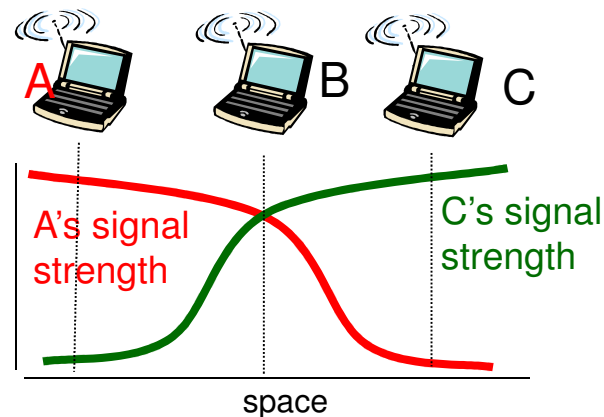
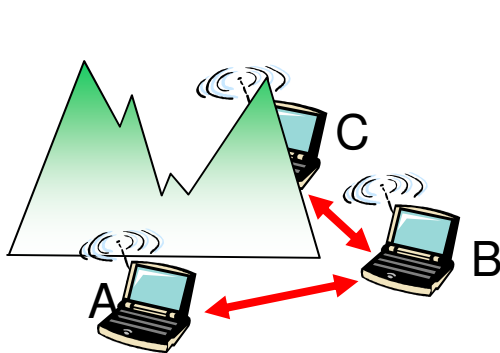
# CDMA: two-sender interference

senders



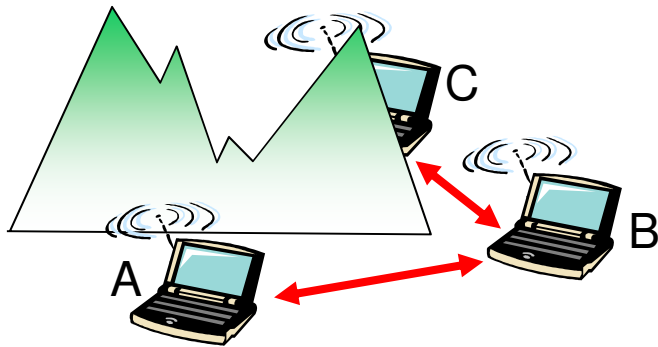
# IEEE 802.11: multiple access

- *avoid collisions: 2+ nodes transmitting at same time*
- *802.11: CSMA - sense before transmitting*
  - *don't collide with ongoing transmission by other node*
- *802.11: no collision detection!*
  - *difficult to receive (sense collisions) when transmitting due to weak received signals (fading)*
  - *can't sense all collisions in any case: hidden terminal, fading*
  - *goal: avoid collisions: CSMA/C(ollision)A(voidance)*



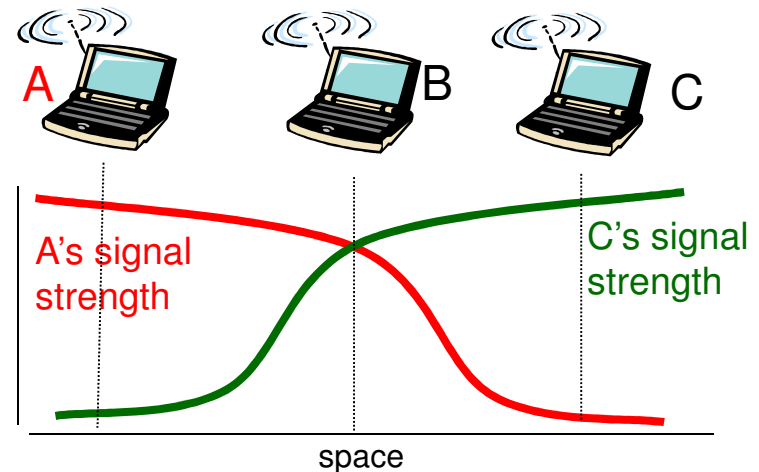
# Wireless network characteristics

*Multiple wireless senders and receivers create additional problems (beyond multiple access):*



## Hidden terminal problem

- B, A hear each other
  - B, C hear each other
  - A, C can not hear each other
- means A, C unaware of their interference at B



## Signal attenuation:

- B, A hear each other
- B, C hear each other
- A, C can not hear each other interfering at B

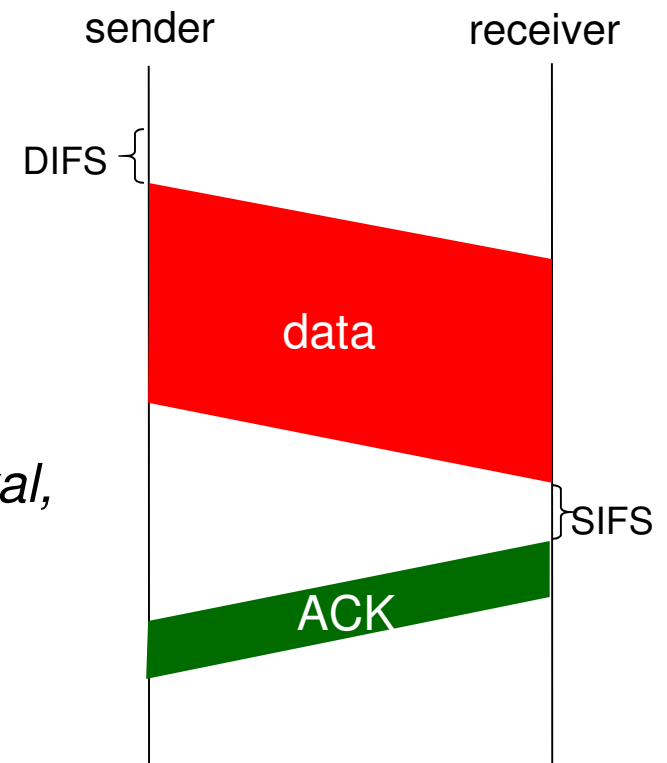
# IEEE 802.11 MAC Protocol: CSMA/CA

## 802.11 sender

- 1 if sense channel idle for **DIFS** then transmit entire frame (no CD)
- 2 if sense channel busy then start random backoff time  
timer counts down while channel idle  
transmit when timer expires  
if no ACK, increase random backoff interval, repeat 2

## 802.11 receiver

- if frame received OK  
return ACK after **SIFS** (ACK needed due to hidden terminal problem)



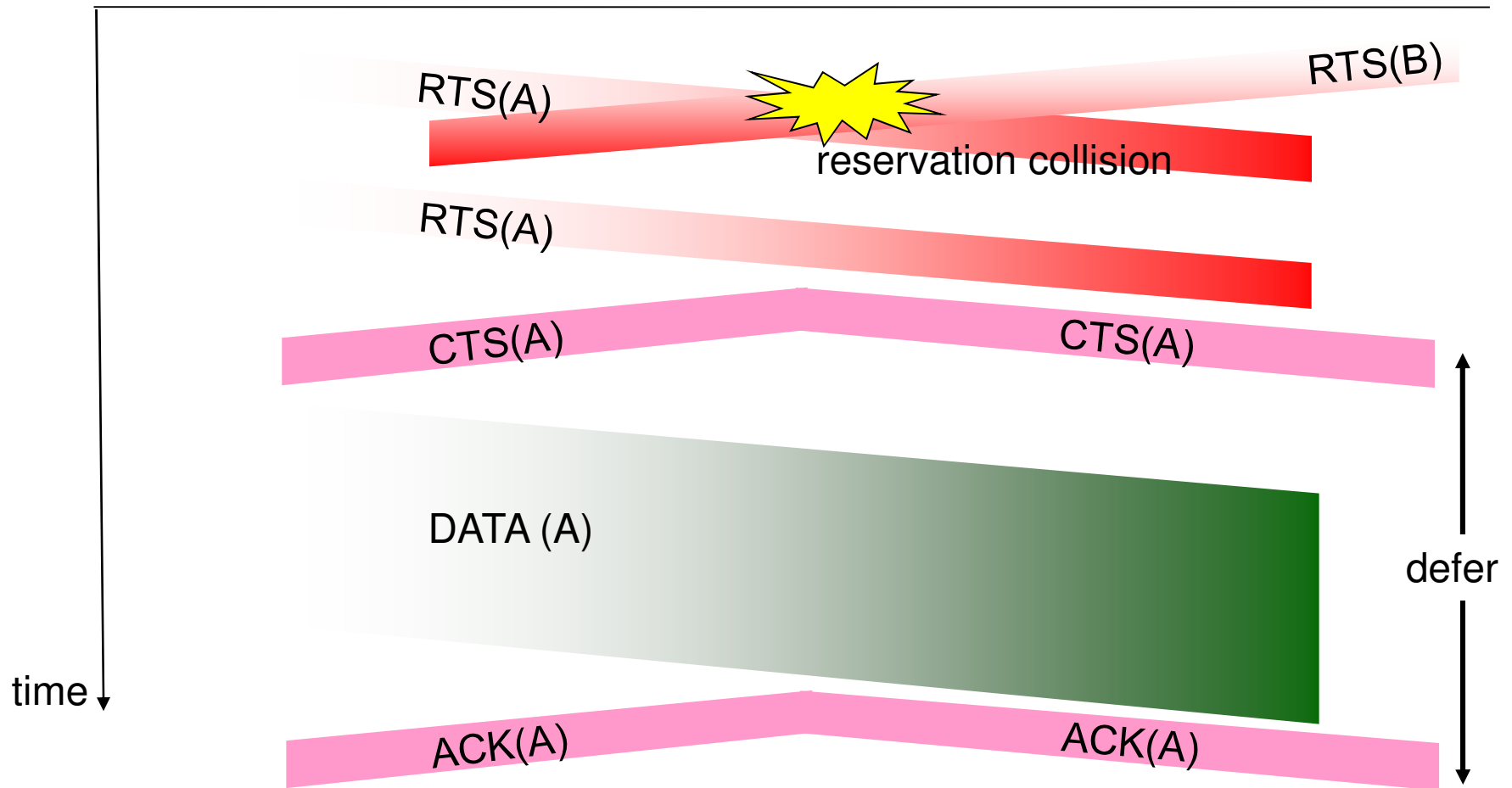
# Avoiding collisions (more)

**idea:** allow sender to “reserve” channel rather than random access of data frames: avoid collisions of long data frames

- sender first transmits small request-to-send (RTS) packets to BS using CSMA
  - RTSs may still collide with each other (but they’re short)
- BS broadcasts clear-to-send CTS in response to RTS
- CTS heard by all nodes
  - sender transmits data frame
  - other stations defer transmissions

avoid data frame collisions completely  
using small reservation packets!

# Collision Avoidance: RTS-CTS exchange



# Link Layer

- *5.1 Introduction and services*
- *5.2 Error detection and correction*
- *5.3 Multiple access protocols*
- *5.4 Link-Layer Addressing*
- *5.5 Ethernet*
- *5.6 Link-layer switches*
- *5.7 PPP*
- *5.8 Wireless links / Wi-Fi*
- *5.9 Link Virtualization: ATM, MPLS*



# Virtualization of networks

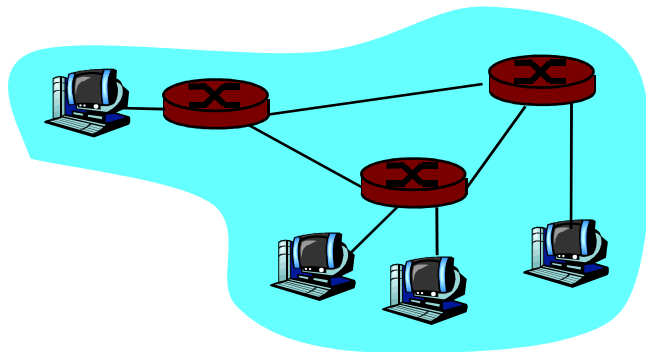
*Virtualization of resources: powerful abstraction in systems engineering:*

- *computing examples: virtual memory, virtual devices*
  - *Virtual machines: e.g., java*
  - *IBM VM os from 1960's/70's*
- *layering of abstractions: don't sweat the details of the lower layer, only deal with lower layers abstractly*

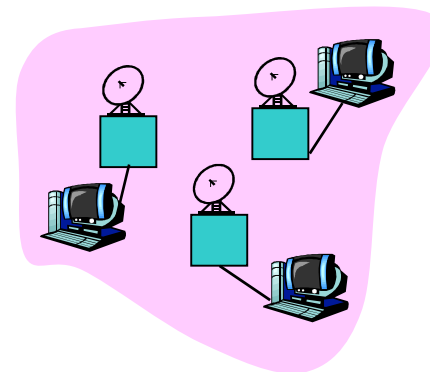
# The Internet: virtualizing networks

1974: multiple unconnected nets ... differing in:

- ARPAnet
- data-over-cable networks
- packet satellite network (Aloha)
- packet radio network
- addressing conventions
- packet formats
- error recovery
- routing



ARPAnet



satellite net

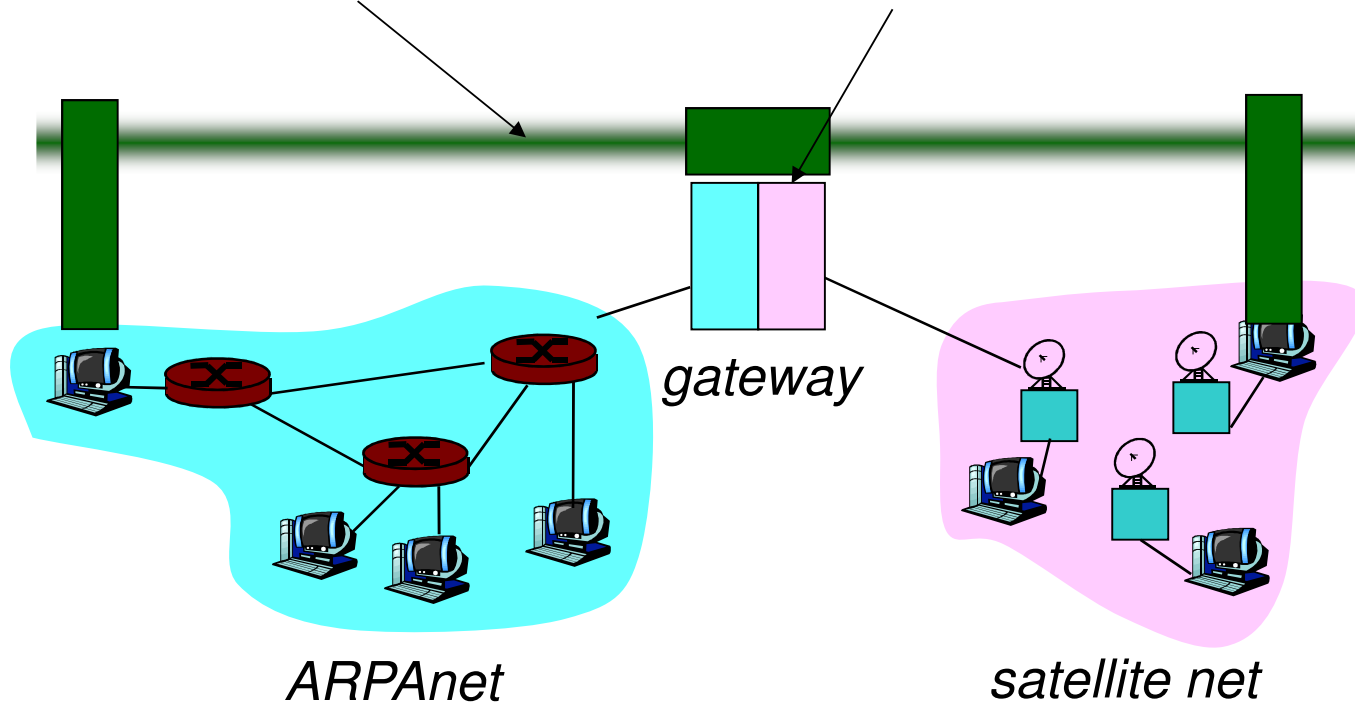
# The Internet: virtualizing networks

## Internetwork layer (IP):

- addressing: internetwork appears as single, uniform entity, despite underlying local network heterogeneity
- network of networks

## Gateway:

- “embed internetwork packets in local packet format or extract them”
- route (at internetwork level) to next gateway



# *Cerf & Kahn's Internetwork Architecture*

## *What is virtualized?*

- *two layers of addressing: internetwork and local network*
  - *new layer (IP) makes everything homogeneous at internetwork layer*
  - *underlying local network technology*
    - *cable*
    - *satellite*
    - *56K telephone modem*
    - *today: ATM, MPLS*
- ... “invisible” at internetwork layer. Looks like a link layer technology to IP!*

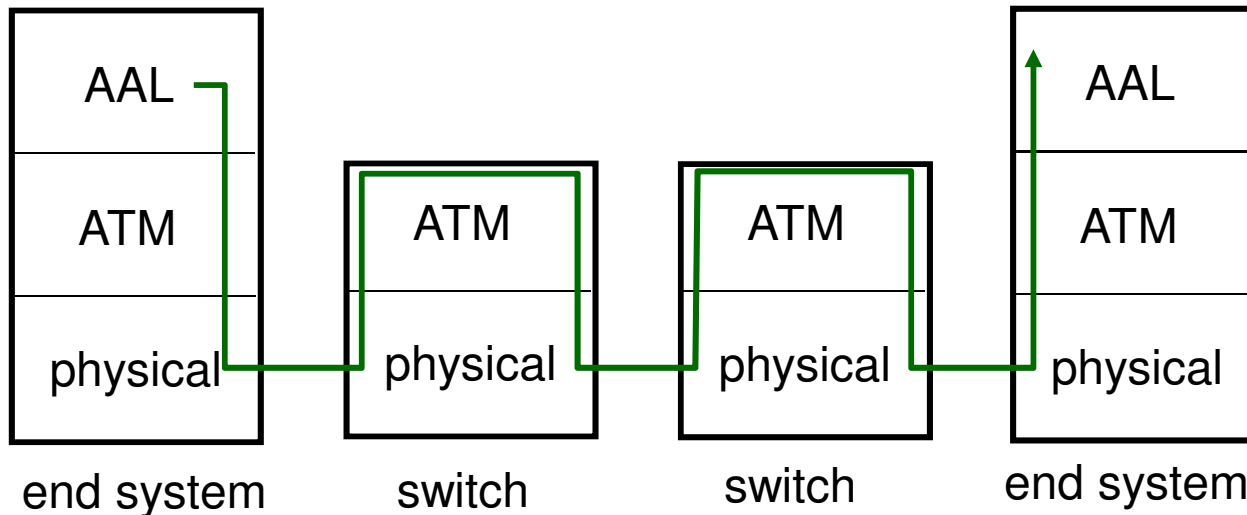
# ***ATM and MPLS***

- *ATM, MPLS separate networks in their own right*
  - *different service models, addressing, routing from Internet*
- *viewed by Internet as logical link connecting IP routers*
  - *just like dialup link is really part of separate network (telephone network)*
- *ATM, MPLS: of technical interest in their own right*

# Asynchronous Transfer Mode: ATM

- **1990's/00 standard for high-speed** (155Mbps to 622 Mbps and higher) **Broadband Integrated Service Digital Network architecture**
- **Goal:** integrated, end-end transport of carry voice, video, data
  - *meeting timing/QoS requirements of voice, video (versus Internet best-effort model)*
  - *“next generation” telephony: technical roots in telephone world*
  - *packet-switching (fixed length packets, called “cells”) using virtual circuits*

# ATM architecture



- *adaptation layer: only at edge of ATM network*
  - *data segmentation/reassembly*
  - *roughly analagous to Internet transport layer*
- *ATM layer: “network” layer*
  - *cell switching, routing*
- *physical layer*

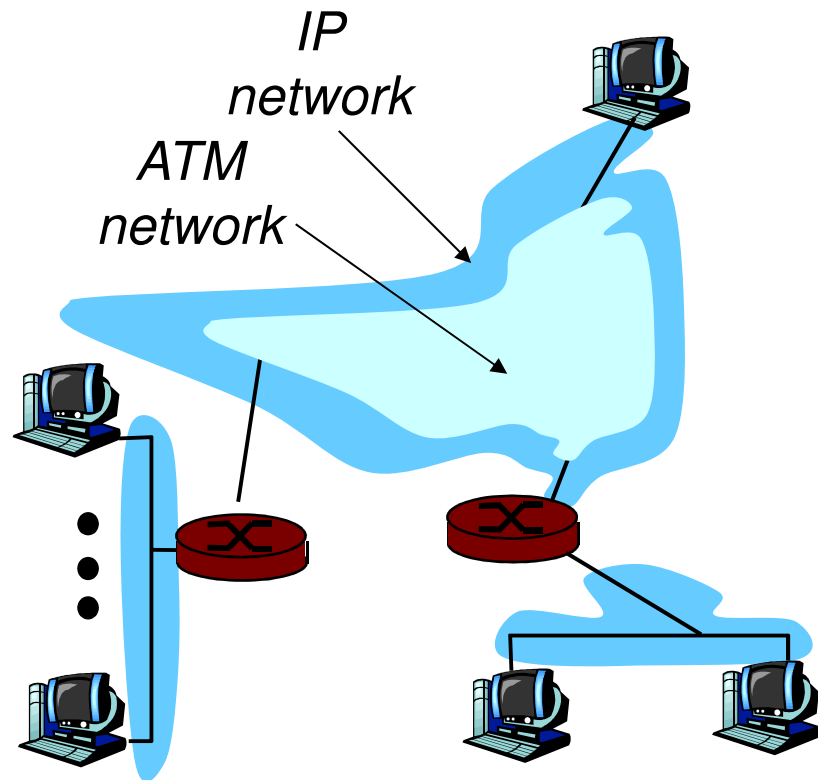
# ATM: network or link layer?

*Vision: end-to-end transport: “ATM from desktop to desktop”*

- *ATM is a network technology*

*Reality: used to connect IP backbone routers*

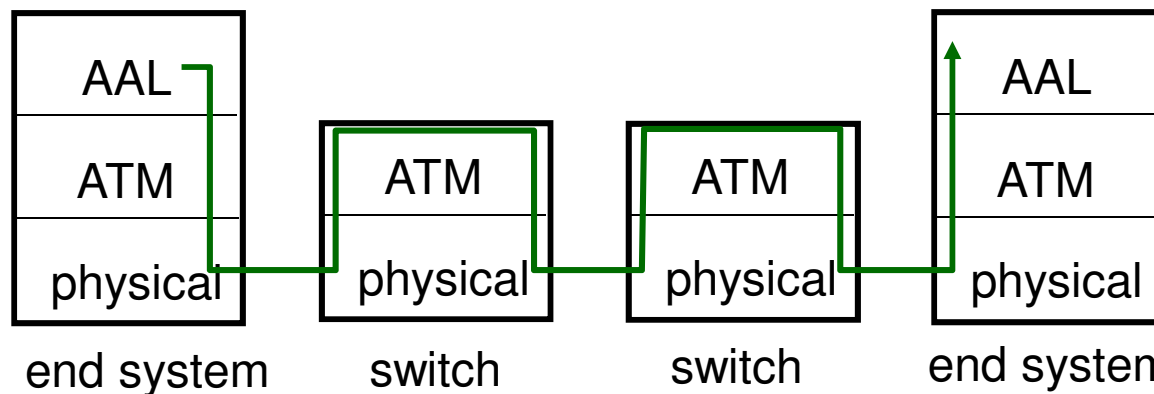
- *“IP over ATM”*
- *ATM as switched link layer, connecting IP routers*





# ATM Adaptation Layer (AAL)

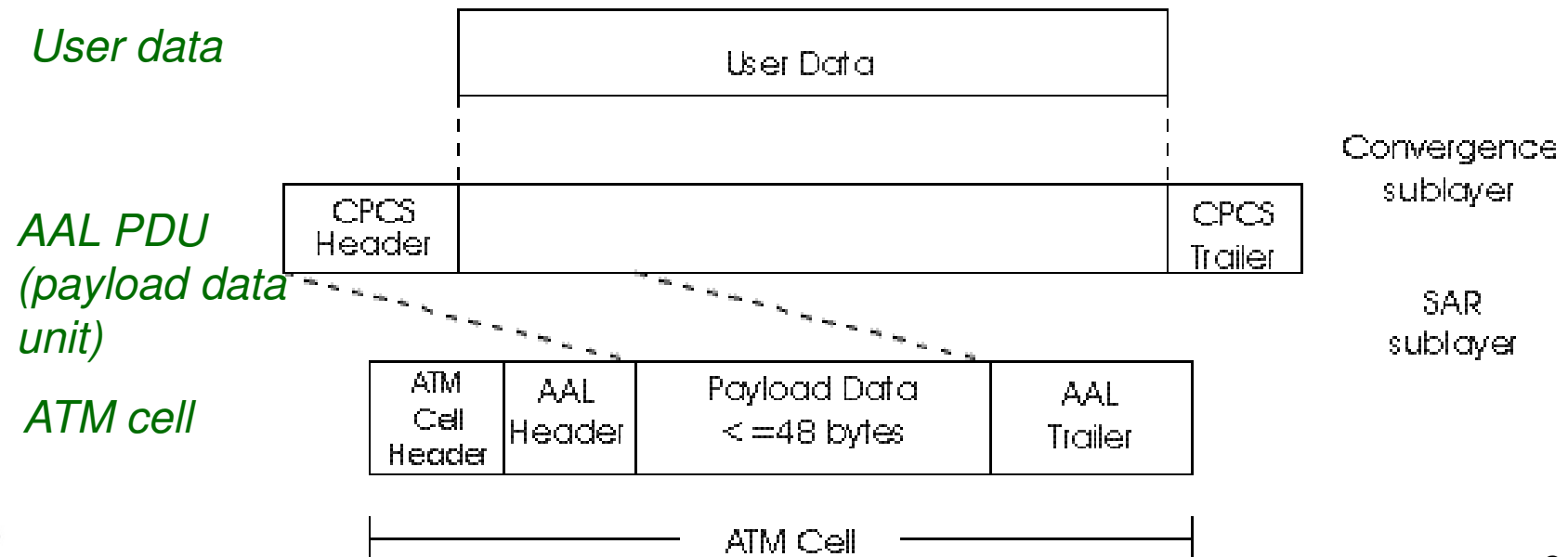
- *ATM Adaptation Layer (AAL): “adapts” upper layers (IP or native ATM applications) to ATM layer below*
- *AAL present **only in end systems**, not in switches*
- *AAL layer segment (header/trailer fields, data) fragmented across multiple ATM cells*
  - *analogy: TCP segment in many IP packets*



# ATM Adaptation Layer (AAL) [more]

*Different versions of AAL layers, depending on ATM service class:*

- *AAL1: for CBR (Constant Bit Rate) services, e.g. circuit emulation*
- *AAL2: for VBR (Variable Bit Rate) services, e.g., MPEG video*
- *AAL5: for data (eg, IP datagrams)*



# ATM Layer

*Service: transport cells across ATM network*

- *analogous to IP network layer*
- *very different services than IP network layer*

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

# ATM Layer: Virtual Circuits

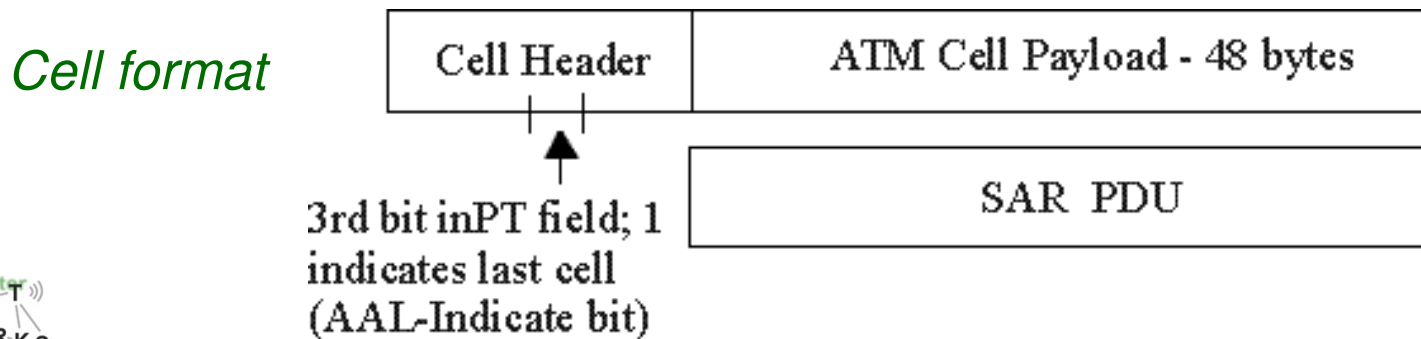
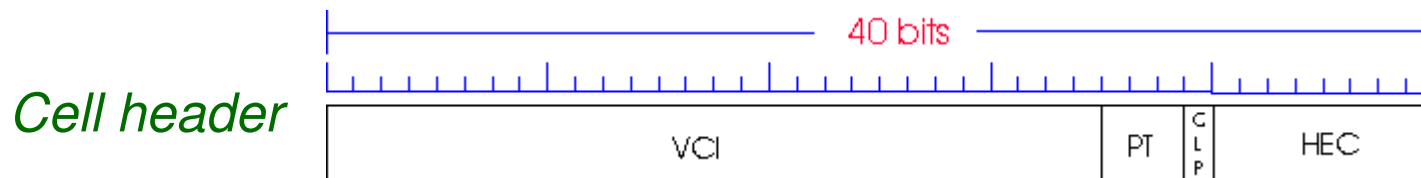
- *VC transport: cells carried on VC from source to dest*
  - *call setup, teardown for each call before data can flow*
  - *each packet carries VC identifier (not destination ID)*
  - *every switch on source-dest path maintain “state” for each passing connection*
  - *link, switch resources (bandwidth, buffers) may be allocated to VC: to get circuit-like perf.*
- *Permanent VCs (PVCs)*
  - *long lasting connections*
  - *typically: “permanent” route between to IP routers*
- *Switched VCs (SVC):*
  - *dynamically set up on per-call basis*

# ATM VCs

- *Advantages of ATM VC approach:*
  - *QoS performance guarantee for connection mapped to VC (bandwidth, delay, delay jitter)*
- *Drawbacks of ATM VC approach:*
  - *Inefficient support of datagram traffic*
  - *one PVC between each source/dest pair) does not scale ( $N*2$  connections needed)*
  - *SVC introduces call setup latency, processing overhead for short lived connections*

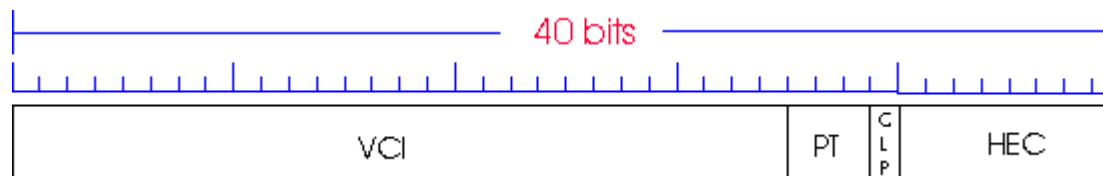
# ATM Layer: ATM cell

- 5-byte ATM cell header
- 48-byte payload
  - Why?: small payload -> short cell-creation delay for digitized voice
  - halfway between 32 and 64 (compromise!)



# ATM cell header

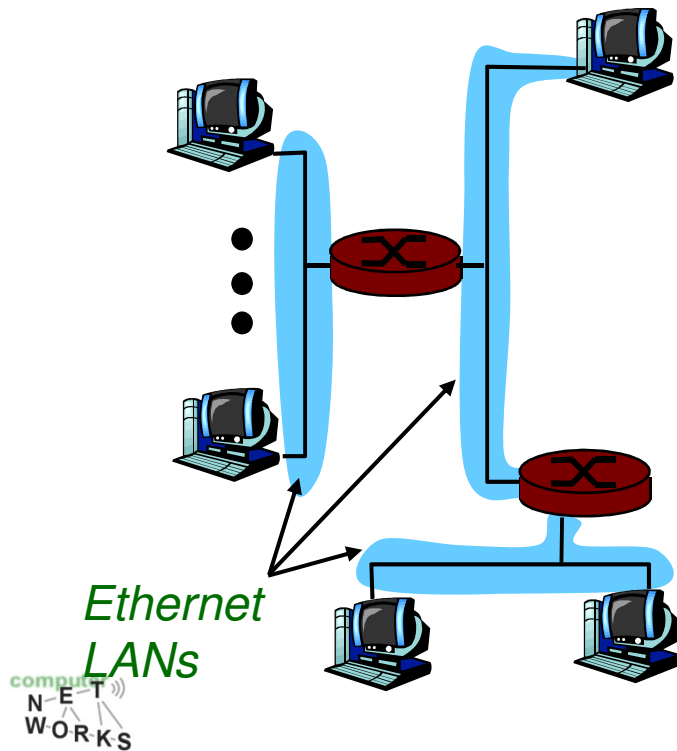
- **VCI:** virtual channel ID
  - will change from link to link thru net
- **PT:** Payload type (e.g. RM cell versus data cell)
- **CLP:** Cell Loss Priority bit
  - $CLP = 1$  implies low priority cell, can be discarded if congestion
- **HEC:** Header Error Checksum
  - cyclic redundancy check



# IP-Over-ATM

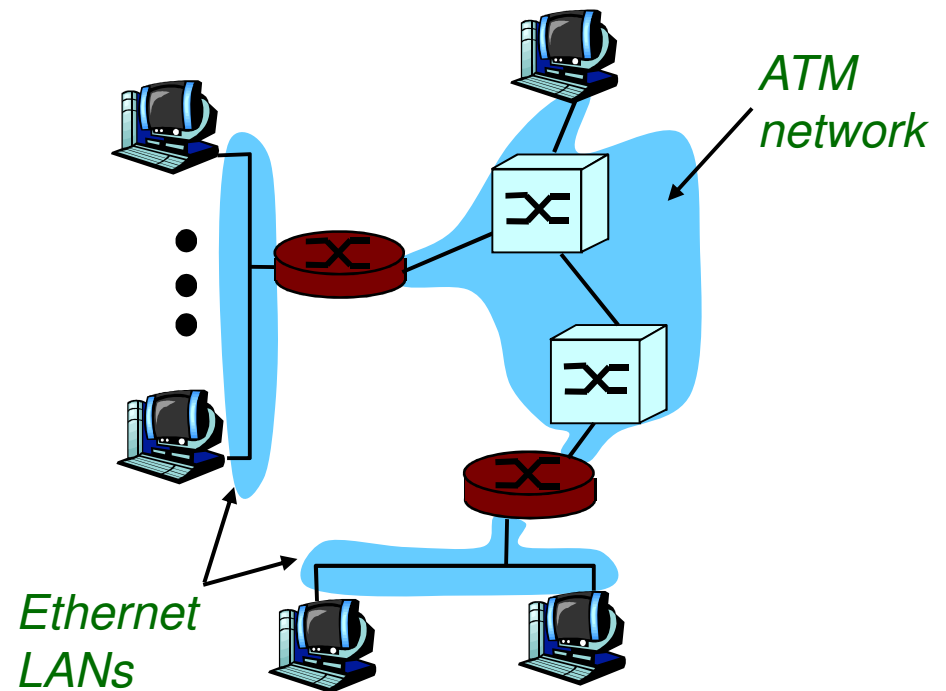
## Classic IP only

- 3 “networks” (e.g., LAN segments)
- MAC (802.3) and IP addresses



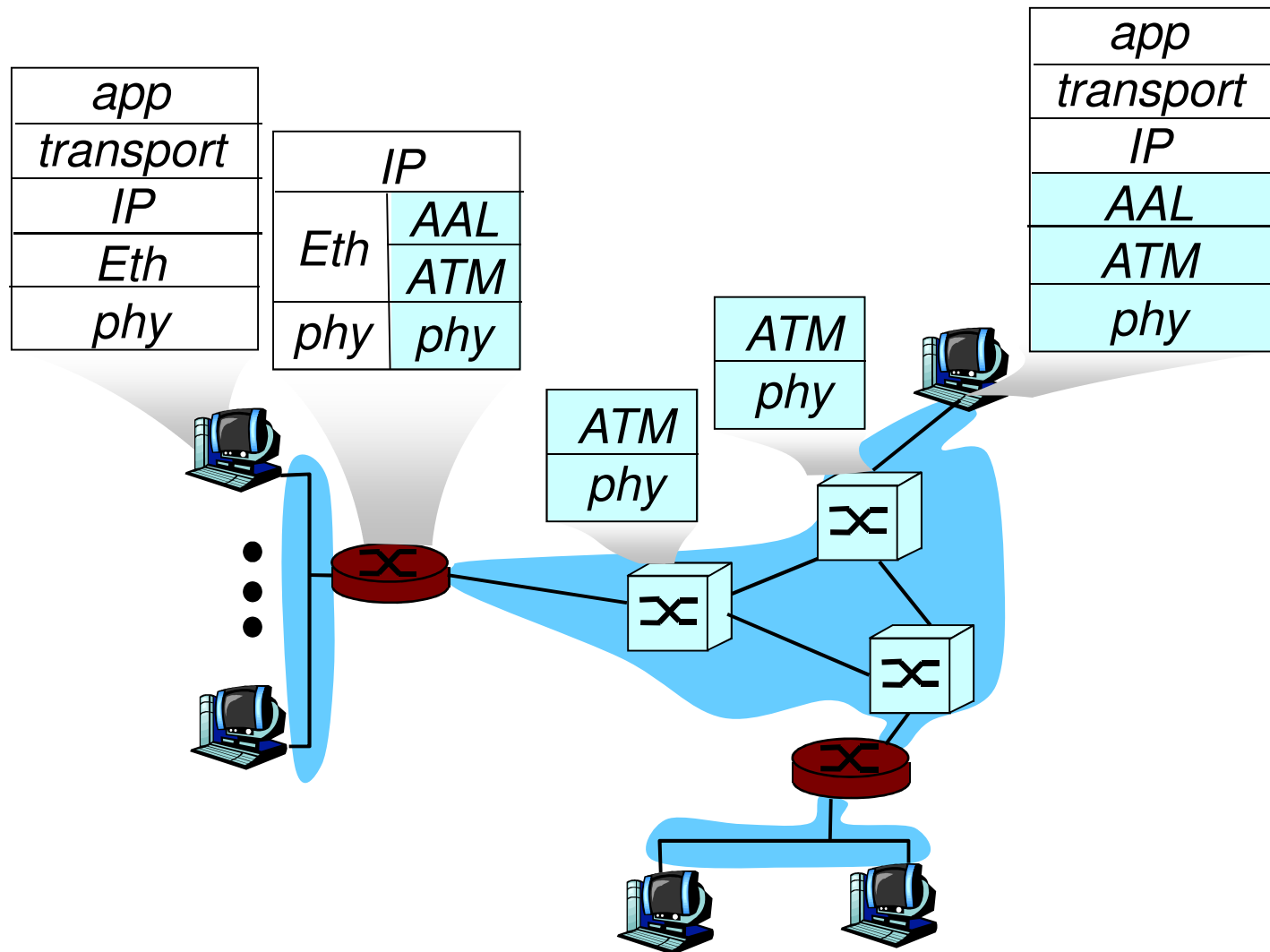
## IP over ATM

- replace “network” (e.g., LAN segment) with ATM network
- ATM addresses, IP addresses





# IP-Over-ATM



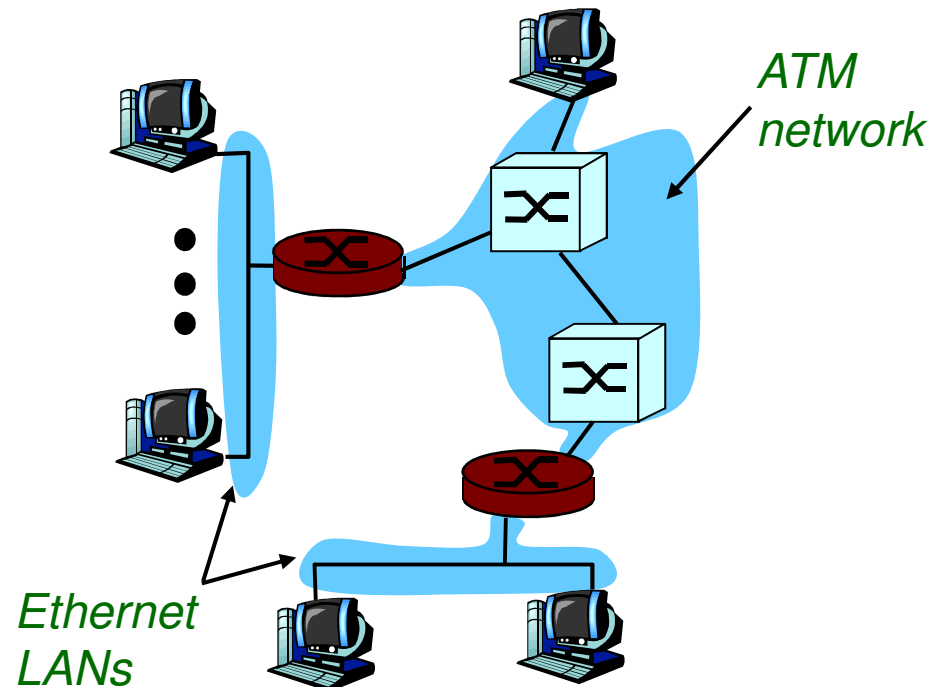
# *Datagram Journey in IP-over-ATM Network*

- *at Source Host:*
  - *IP layer maps between IP, ATM dest address (using ARP)*
  - *passes datagram to AAL5*
  - *AAL5 encapsulates data, segments cells, passes to ATM layer*
- *ATM network: moves cell along VC to destination*
- *at Destination Host:*
  - *AAL5 reassembles cells into original datagram*
  - *if CRC OK, datagram is passed to IP*

# IP-Over-ATM

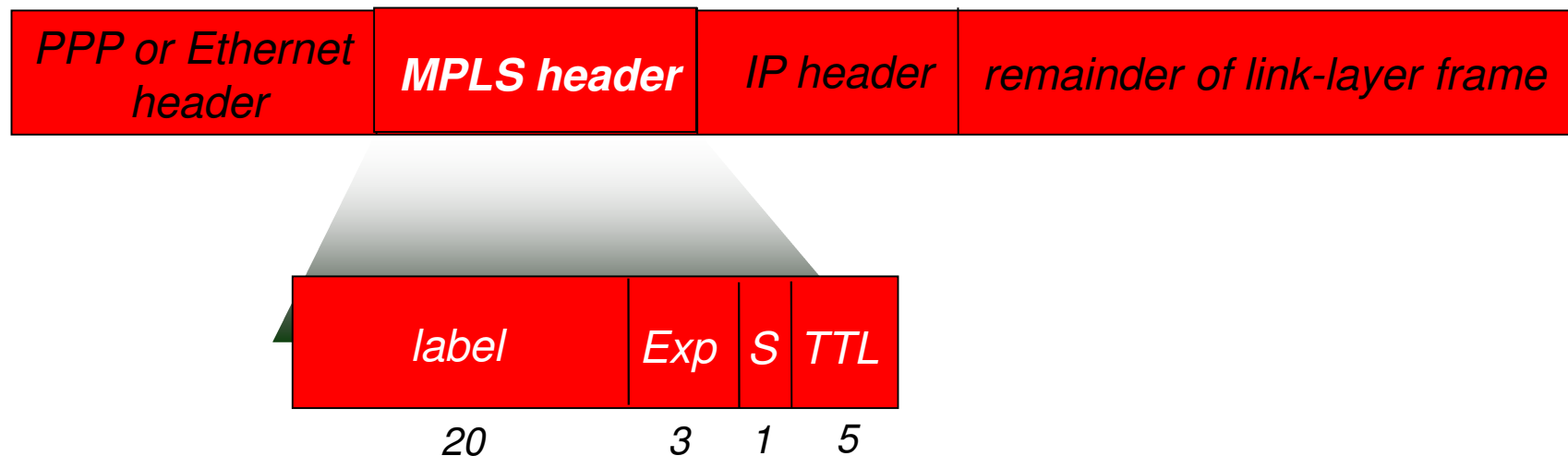
## Issues:

- *IP datagrams into ATM AAL5 PDUs*
- *from IP addresses to ATM addresses*
  - *just like IP addresses to 802.3 MAC addresses!*



# Multiprotocol label switching (MPLS)

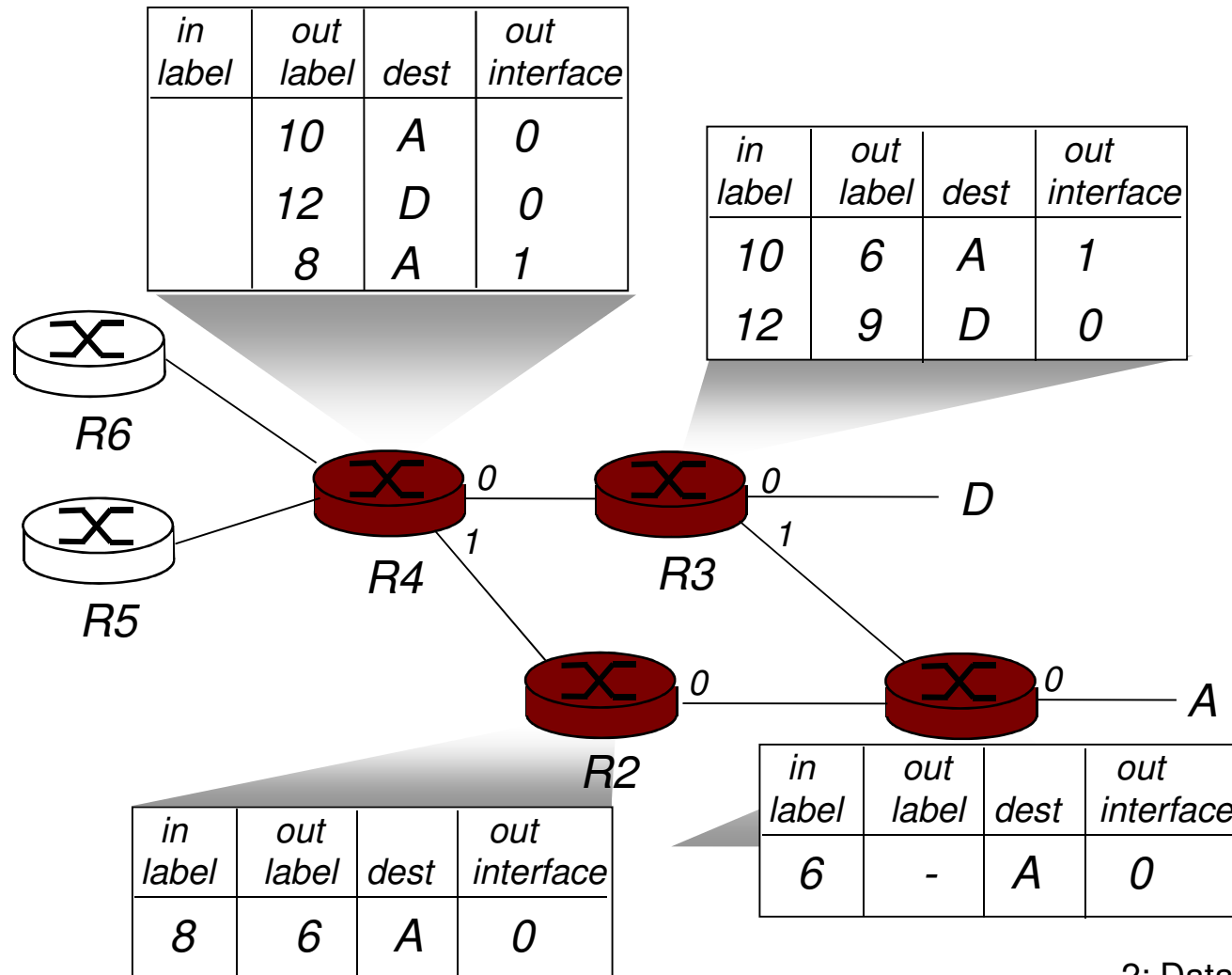
- *initial goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding*
  - *borrowing ideas from Virtual Circuit (VC) approach*
  - *but IP datagram still keeps IP address!*



# *MPLS capable routers*

- *a.k.a. label-switched router*
- *forwards packets to outgoing interface based only on label value (don't inspect IP address)*
  - *MPLS forwarding table distinct from IP forwarding tables*
- *signaling protocol needed to set up forwarding*
  - *RSVP-TE*
  - *forwarding possible along paths that IP alone would not allow (e.g., source-specific routing) !!*
  - *use MPLS for traffic engineering*
- *must co-exist with IP-only routers*

# MPLS forwarding tables



# Chapter 2: Summary

- *principles behind data link layer services:*
  - *error detection, correction*
  - *sharing a broadcast channel: multiple access*
  - *link layer addressing*
- *instantiation and implementation of various link layer technologies*
  - *switched LANS*
  - *PPP*
  - *Wireless links*
  - *virtualized networks as a link layer: ATM, MPLS*

# *Appendix*



# 802.11 frame: addressing



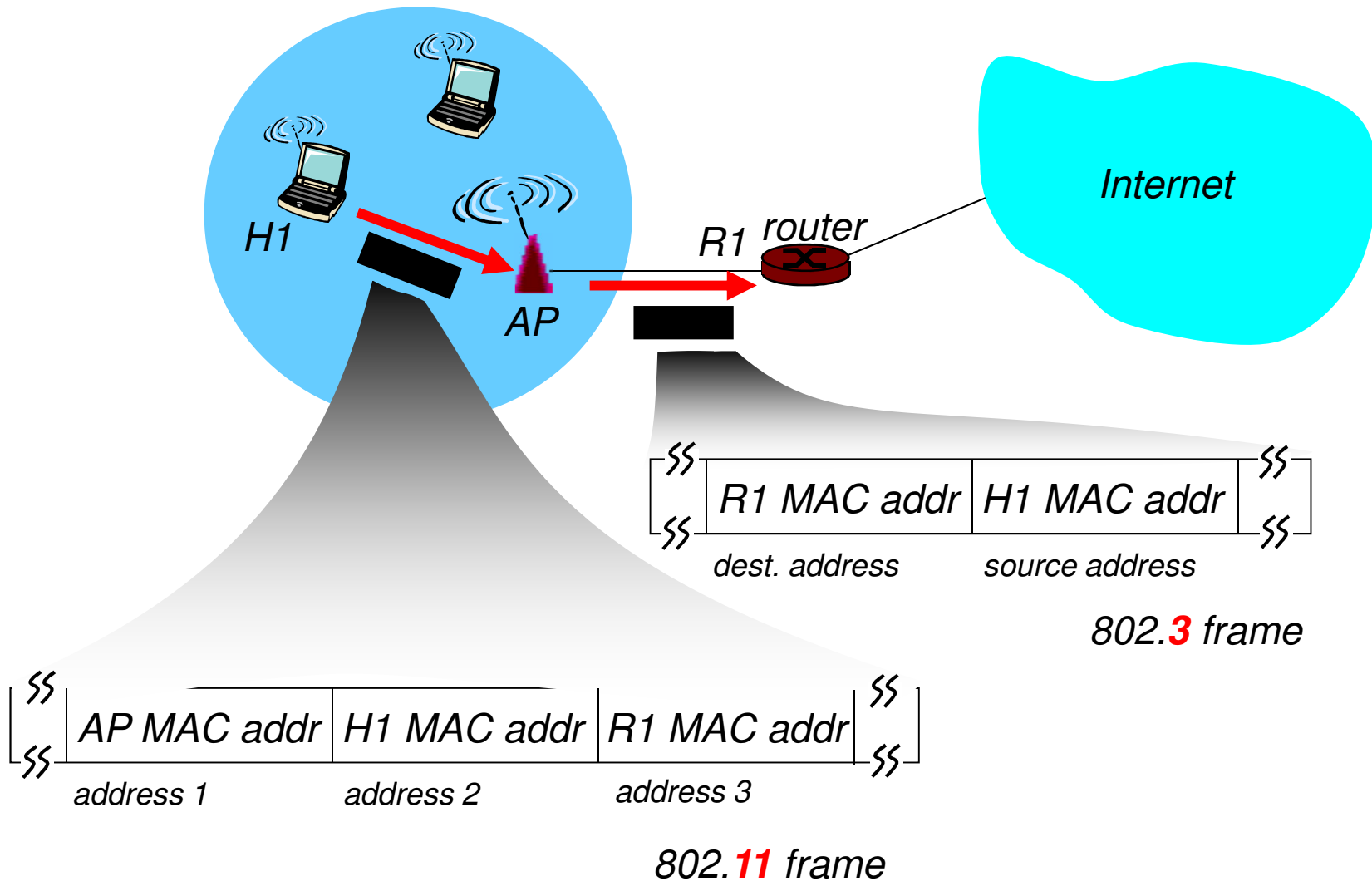
*Address 1: MAC address of wireless host or AP to receive this frame*

*Address 2: MAC address of wireless host or AP transmitting this frame*

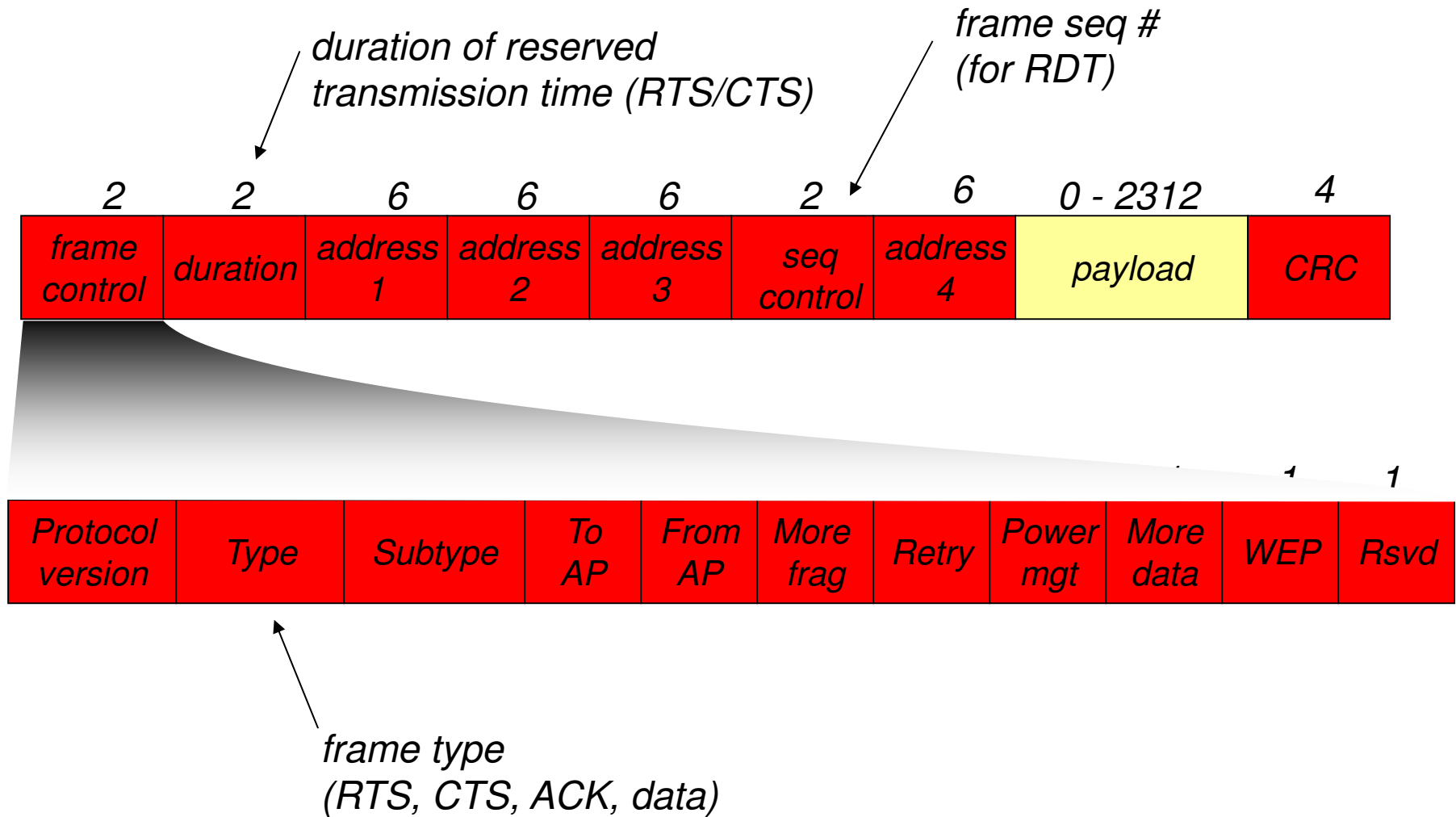
*Address 3: MAC address of router interface to which AP is attached*

*Address 4: used only in ad hoc mode*

# 802.11 frame: addressing

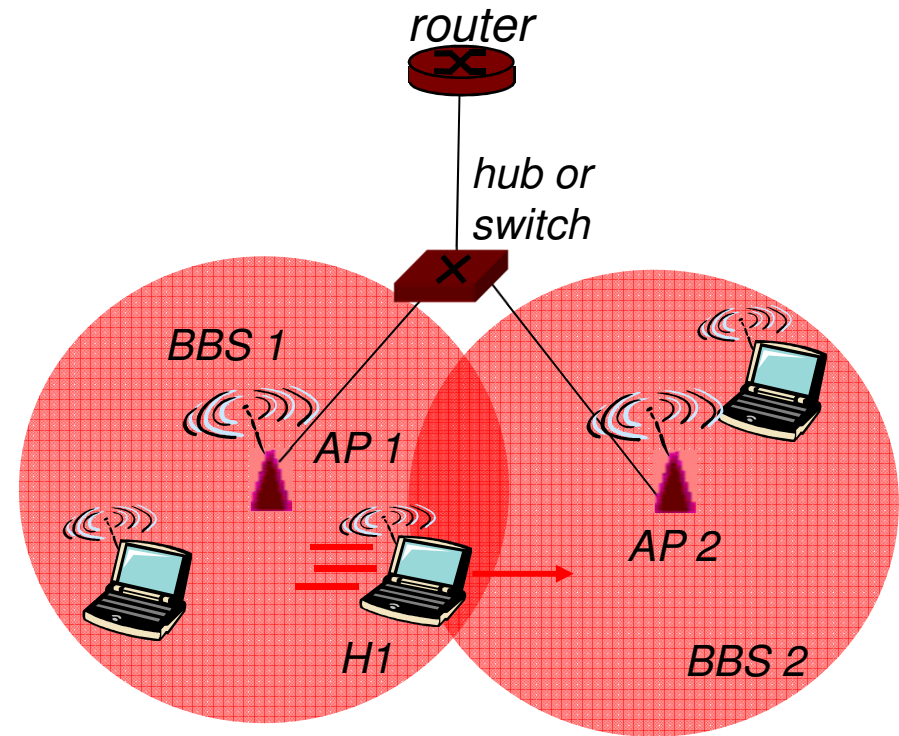


# 802.11 frame: more



# 802.11: mobility within same subnet

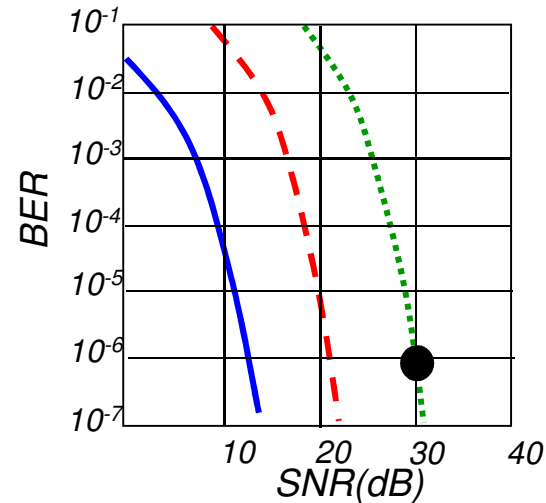
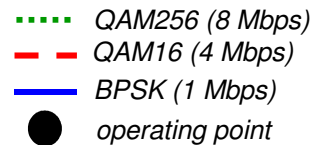
- H1 remains in same IP subnet: IP address can remain same
- switch: which AP is associated with H1?
  - self-learning (Ch. 5): switch will see frame from H1 and “remember” which switch port can be used to reach H1



# 802.11: advanced capabilities

## Rate Adaptation

- base station, mobile dynamically change transmission rate (physical layer modulation technique) as mobile moves, SNR varies



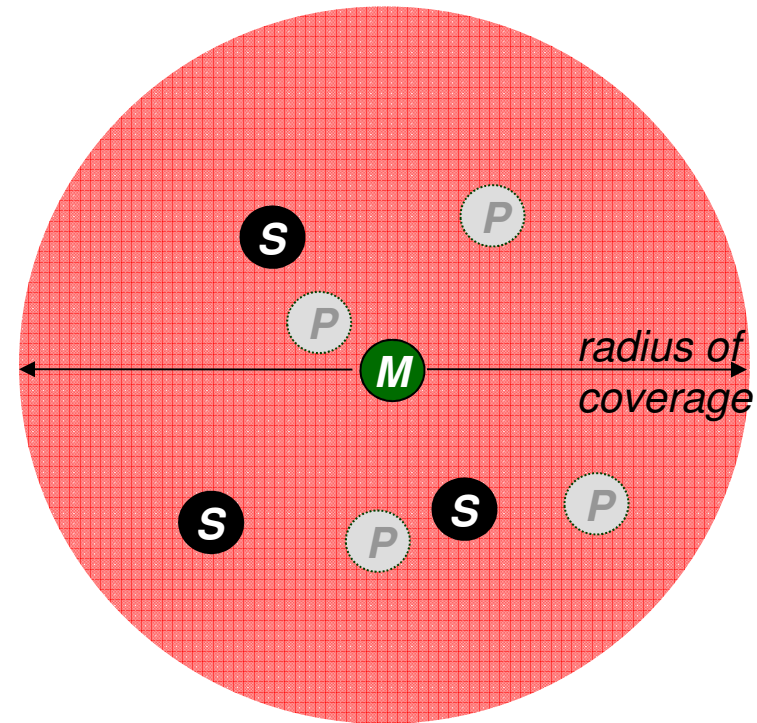
1. SNR decreases, BER increase as node moves away from base station
2. When BER becomes too high, switch to lower transmission rate but with lower BER

# 802.11: advanced capabilities

- Power Management
- node-to-AP: “I am going to sleep until next beacon frame”
  - AP knows not to transmit frames to this node
  - node wakes up before next beacon frame
- beacon frame: contains list of mobiles with AP-to-mobile frames waiting to be sent
  - node will stay awake if AP-to-mobile frames to be sent; otherwise sleep again until next beacon frame

# 802.15: personal area network

- less than 10 m diameter
- replacement for cables (mouse, keyboard, headphones)
- ad hoc: no infrastructure
- master/slaves:
  - slaves request permission to send (to master)
  - master grants requests
- 802.15: evolved from Bluetooth specification
  - 2.4-2.5 GHz radio band
  - up to 721 kbps



**M** Master device

**S** Slave device

**P** Parked device (inactive)

# 802.16: WiMAX

- like 802.11 & cellular:  
base station model
  - transmissions to/from base station by hosts with omnidirectional antenna
  - base station-to-base station backhaul with point-to-point antenna
- unlike 802.11:
  - range ~ 6 miles (“city rather than coffee shop”)
  - ~14 Mbps

