Announcement

- Final 3/18 (Th) 12:00-1:30pm, Rm 381
- Close Book
- One 8.5" by 11" sheet of paper permitted (single side)
- Cover network layer, data link layer and network security

- Extra office hour next Tu. 2-4pm, Rm 330

Outline

- Network Layer
  - Routing Principles
    - Link State Algorithm
    - Distance Vector Algorithm
  - Hierarchical Routing
  - The Internet (IP) Protocol
    - IPv4 addressing
    - Moving a datagram from source to destination
    - Datagram format
    - IP fragmentation
    - ICMP: Internet Control Message Protocol
    - NAT: Network Address Translation

Routing Algorithm classification

Global or decentralized information?

Global:

- all routers have complete topology, link cost info
- "link state" algorithms

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

Static or dynamic?

Static:

- routes change slowly over time

Dynamic:

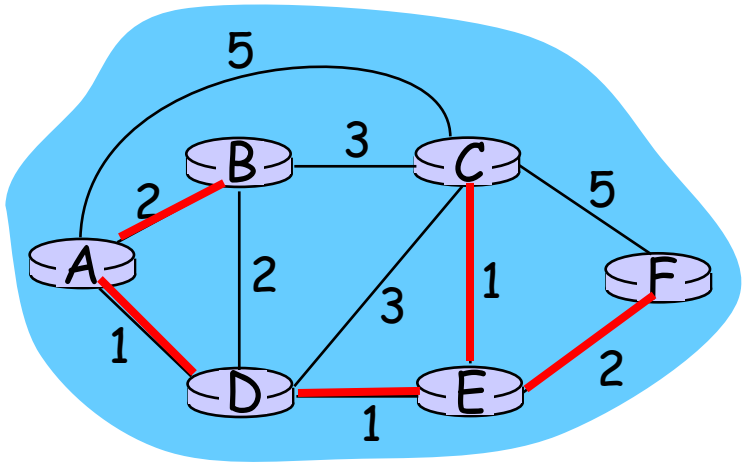
- routes change more quickly
  - periodic update
  - in response to link cost changes

Link-State: Dijsktra's Algorithm

```
1  Initialization:
2    N = {A}
3    for all nodes v
4      if v adjacent to A
5        then D(v) = c(A,v)
6        else D(v) = infinity
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 cost to v is either old cost to v or known
14     shortest path cost to w plus cost from w to v */
15 until all nodes in N
```

Dijkstra's algorithm: example

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	infinity	infinity
1	AD	2,A	4,D		2,D	infinity
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					



Distance Vector Routing Algorithm

iterative:

- continues until no nodes exchange info.
- *self-terminating*: no "signal" to stop

asynchronous:

- nodes need *not* exchange info/iterate in lock step!

distributed:

- each node communicates *only* with directly-attached neighbors

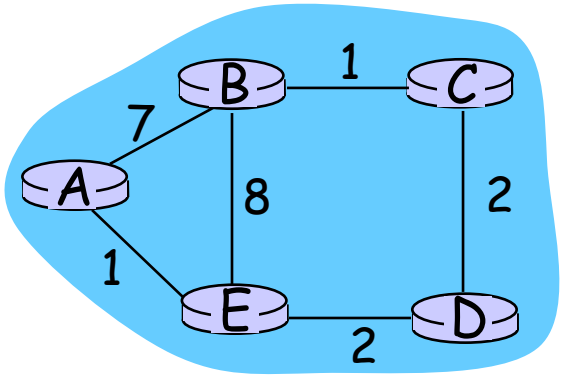
Distance Table data structure

- each node has its own
- row for each possible destination
- column for each directly-attached neighbor to node
- example: in node X, for dest. Y via neighbor Z:

$$D^X(Y,Z) = \text{distance } \textit{from } X \textit{ to } Y, \textit{ via } Z \text{ as next hop}$$

$$= c(X,Z) + \min_w \{D^Z(Y,w)\}$$

Distance Table: example



$D^E(C,D) = c(E,D) + \min_w\{D^D(C,w)\}$
$= 2+2 = 4$

$D^E(A,D) = c(E,D) + \min_w\{D^D(A,w)\}$
$= 2+3 = 5$ loop!

$D^E(A,B) = c(E,B) + \min_w\{D^B(A,w)\}$
$= 8+6 = 14$ loop!

$D^E()$ — cost to destination via

destination	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

Distance table gives routing table

cost to destination via

$D^E()$

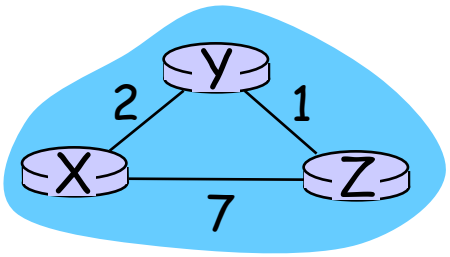
destination	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

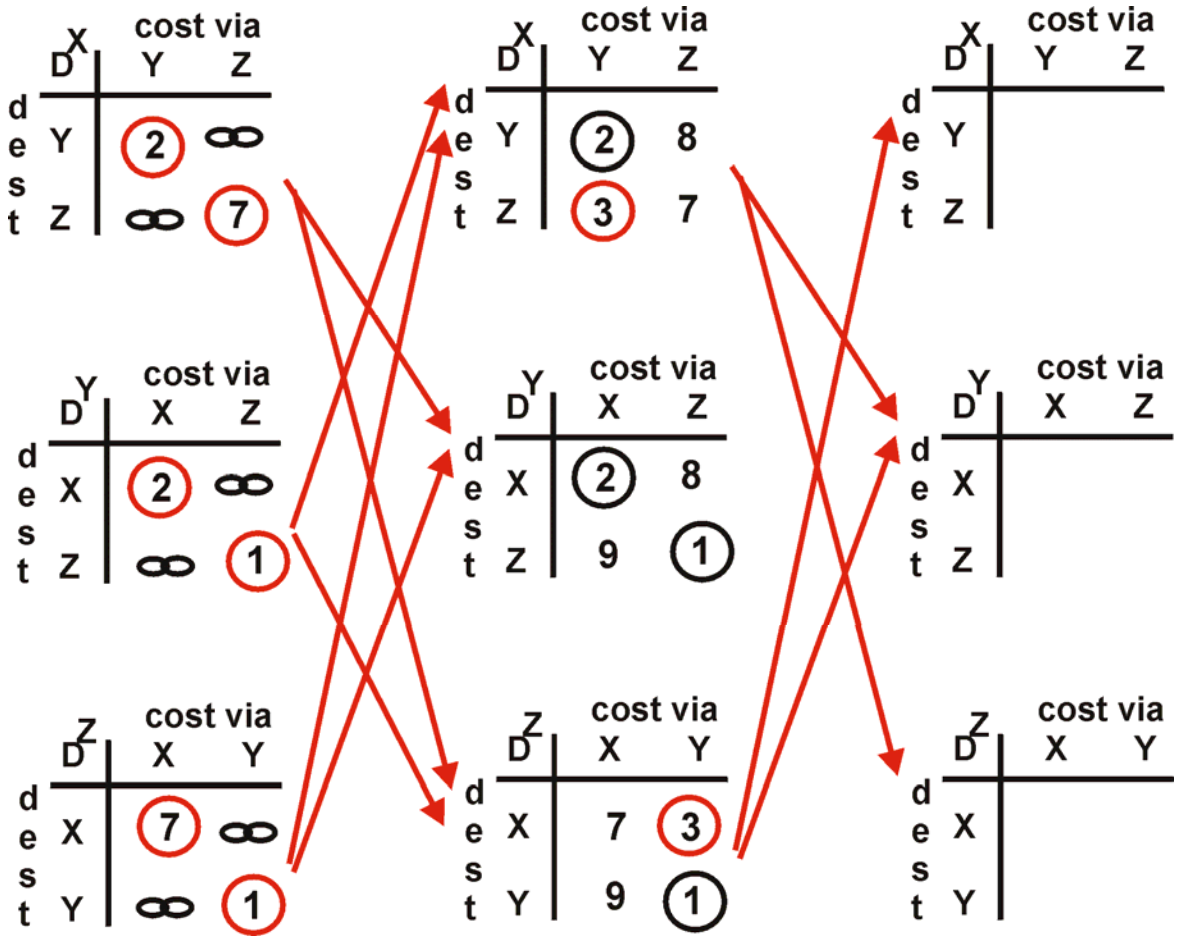
Outgoing link to use, cost

destination	
A	A,1
B	D,5
C	D,4
D	D,2

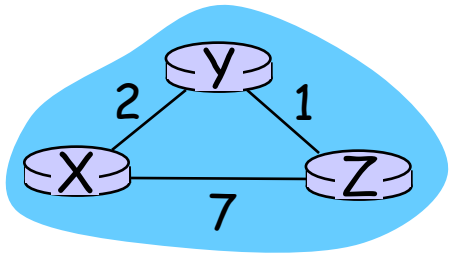
Distance table → Routing table

Distance Vector Algorithm: example





Distance Vector Algorithm: example



D^X

dest	via Y	via Z
Y	2	∞
Z	∞	7

D^Y

dest	via X	via Z
X	2	∞
Z	∞	1

D^Z

dest	via X	via Y
X	7	∞
Y	∞	1

D^X

dest	via Y	via Z
Y	2	8
Z	3	7

$$D^X(Y,Z) = c(X,Z) + \min_w\{D^Z(Y,w)\}$$
$$= 7+1 = 8$$

$$D^X(Z,Y) = c(X,Y) + \min_w\{D^Y(Z,w)\}$$
$$= 2+1 = 3$$

Comparison of LS and DV algorithms

Message complexity

- LS: with n nodes, E links, O(nE) msgs sent each
- DV: exchange between neighbors only
  - convergence time varies

Speed of Convergence

- LS: $O(n^2)$ algorithm requires O(nE) msgs
  - may have oscillations
- DV: convergence time varies
  - may be routing loops
  - count-to-infinity problem

Robustness: what happens if router malfunctions?

LS:

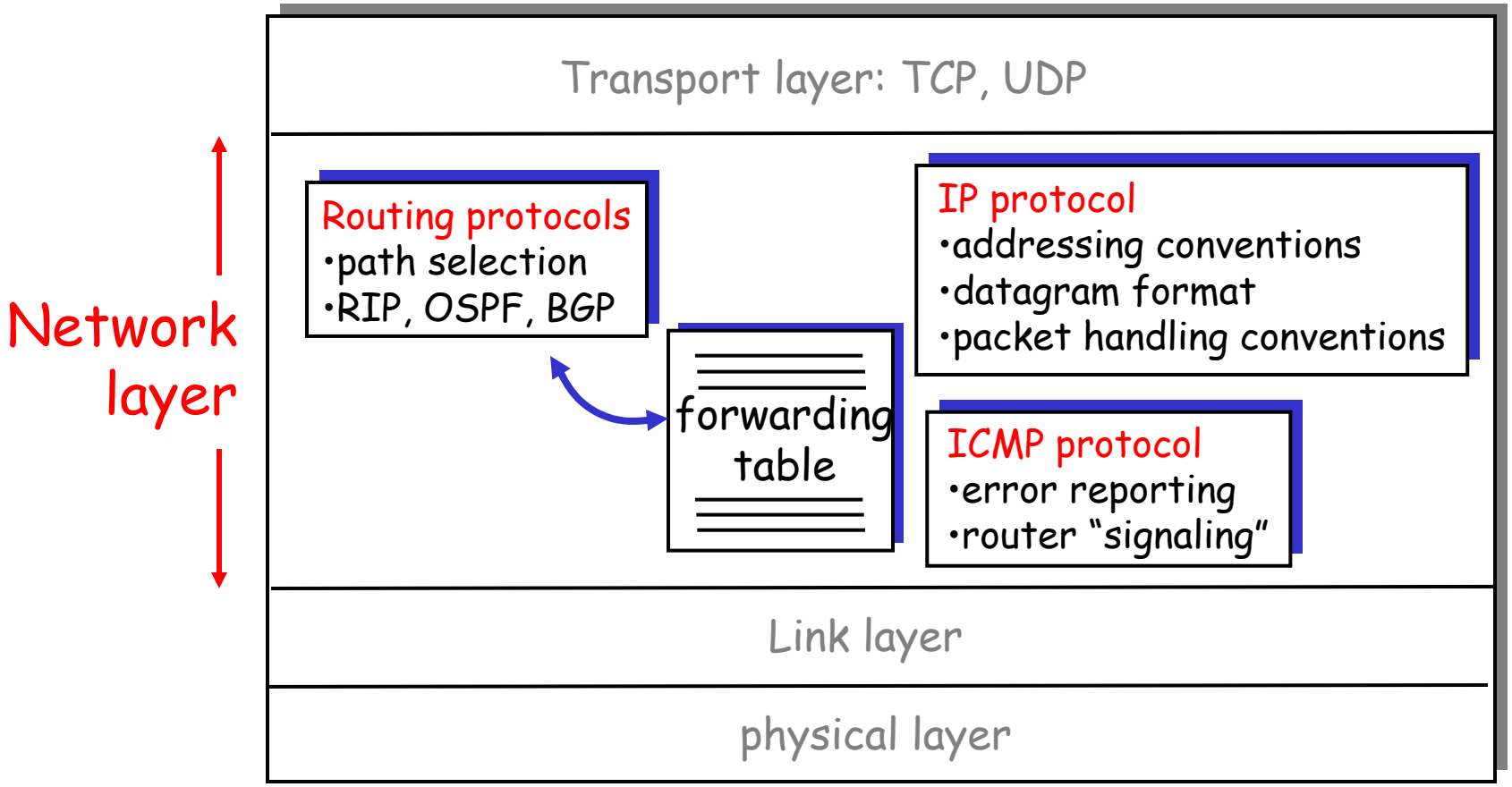
- node can advertise incorrect *link* cost
- each node computes only its *own* table

DV:

- DV node can advertise incorrect *path* cost
- each node's table used by others
  - error propagate thru network

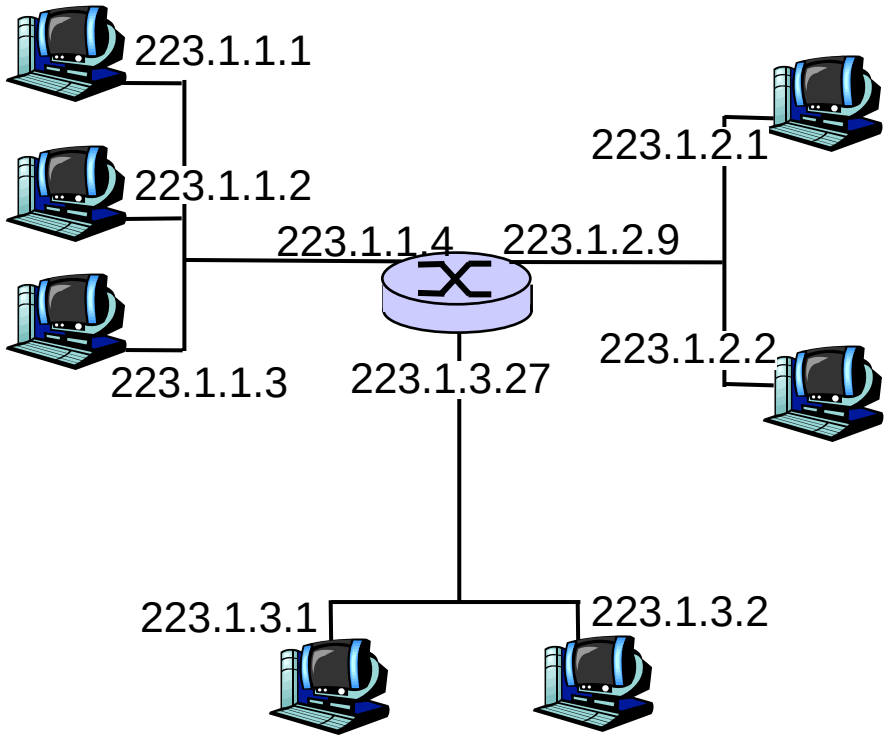
The Internet Network layer

Host, router network layer functions:



IP Addressing: introduction

- IP address: 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with each interface

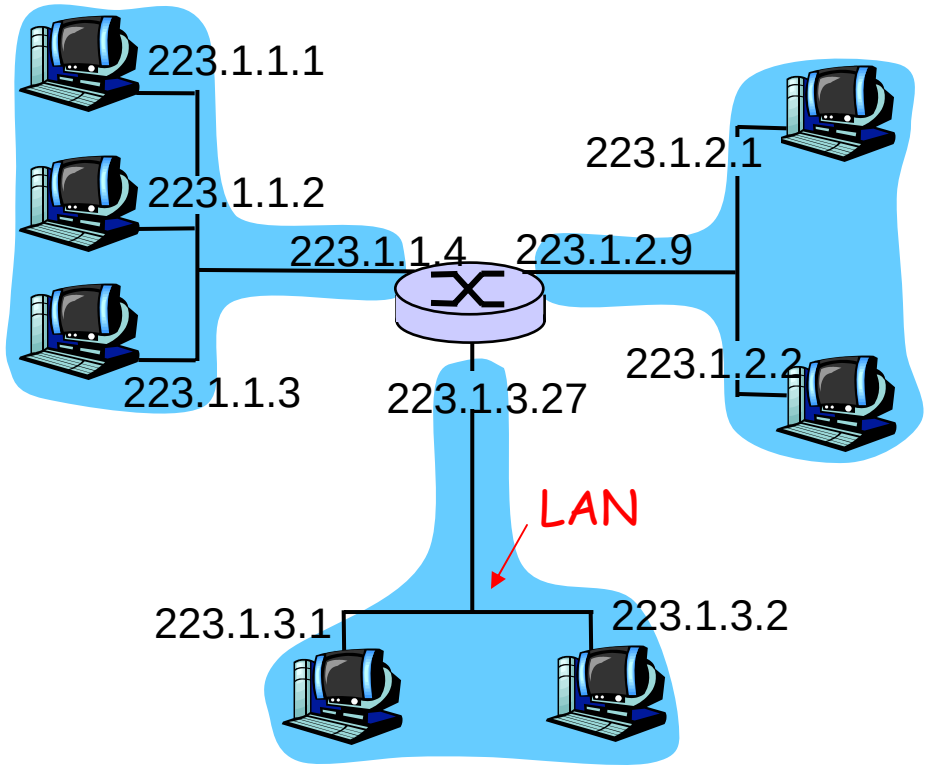


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

IP Addressing

- IP address:
  - network part (high order bits)
  - host part (low order bits)
- *What's a network ?* (from IP address perspective)
  - device interfaces with same network part of IP address
  - can physically reach each other without intervening router



network consisting of 3 IP networks (for IP addresses starting with 223, first 24 bits are network address)

Getting a datagram from source to dest.

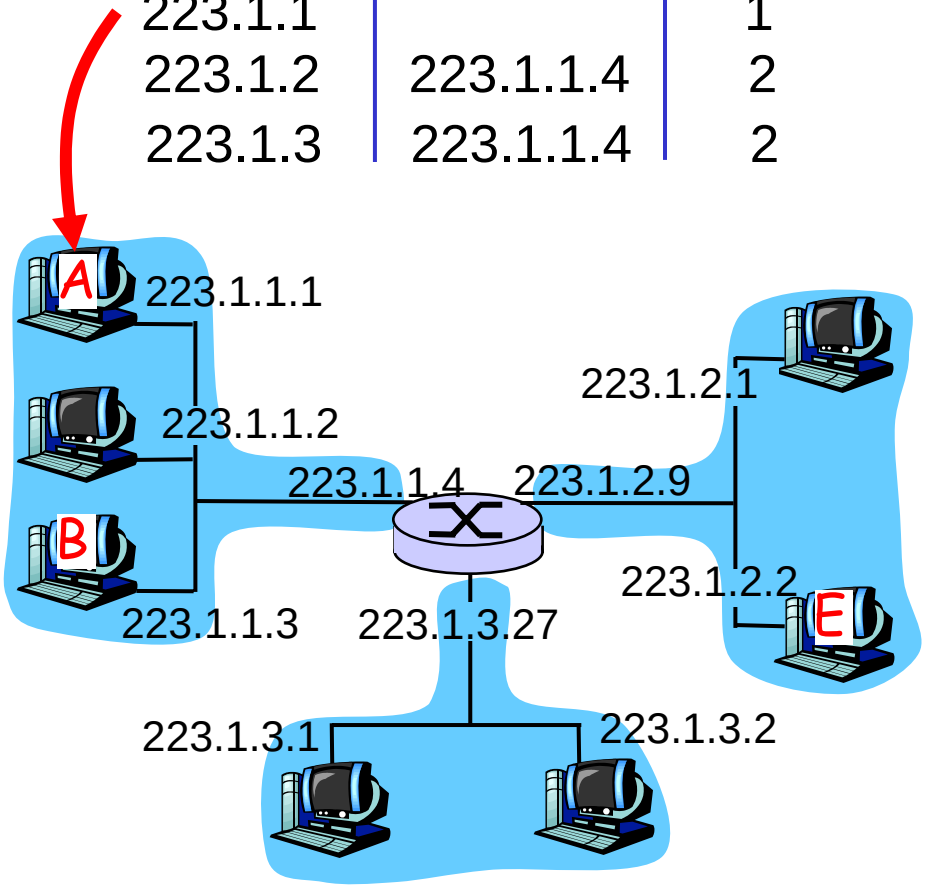
IP datagram:

misc fields	source IP addr	dest IP addr	data

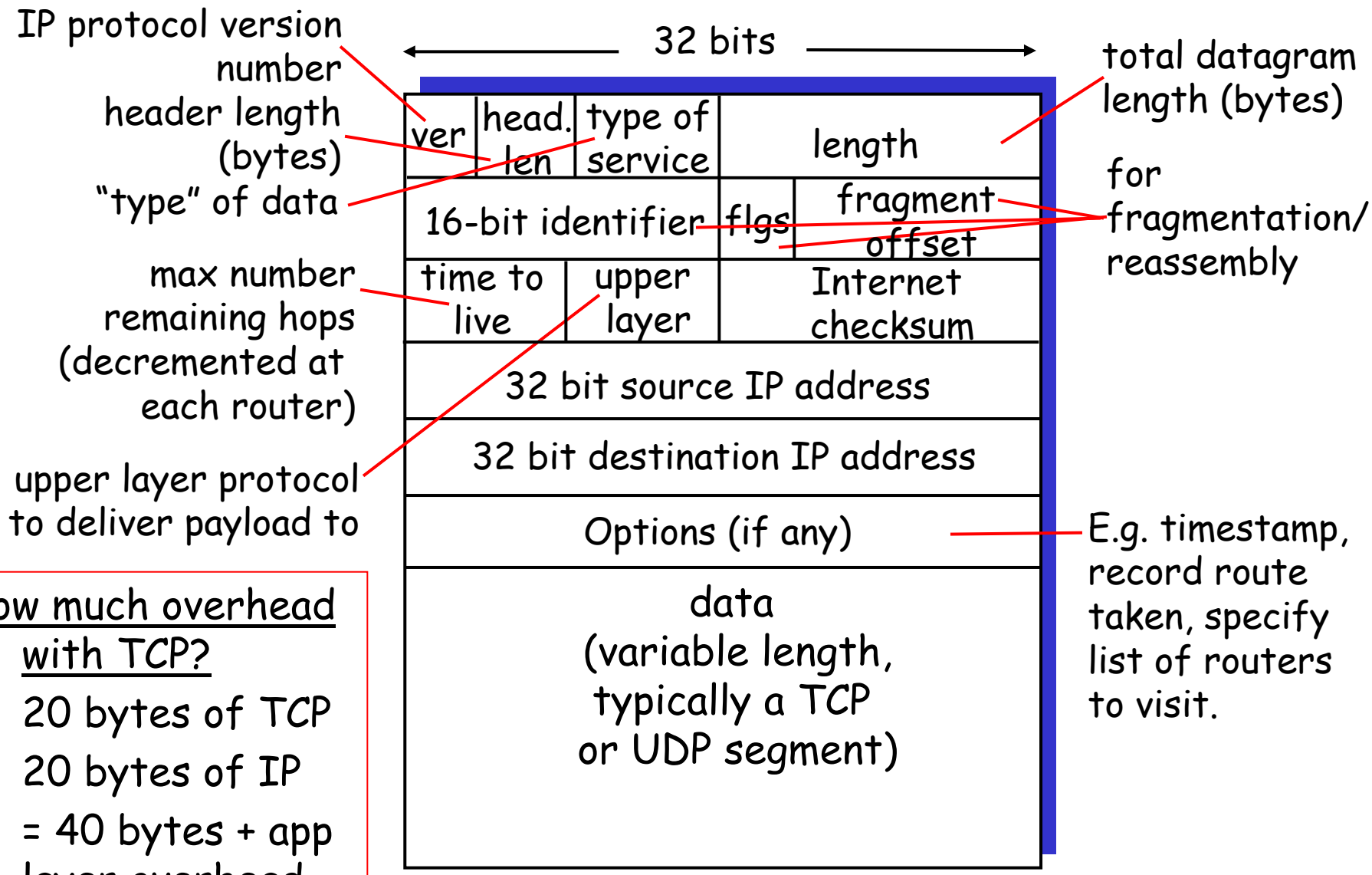
- datagram remains unchanged, as it travels source to destination
- addr fields of interest here

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



IP datagram format



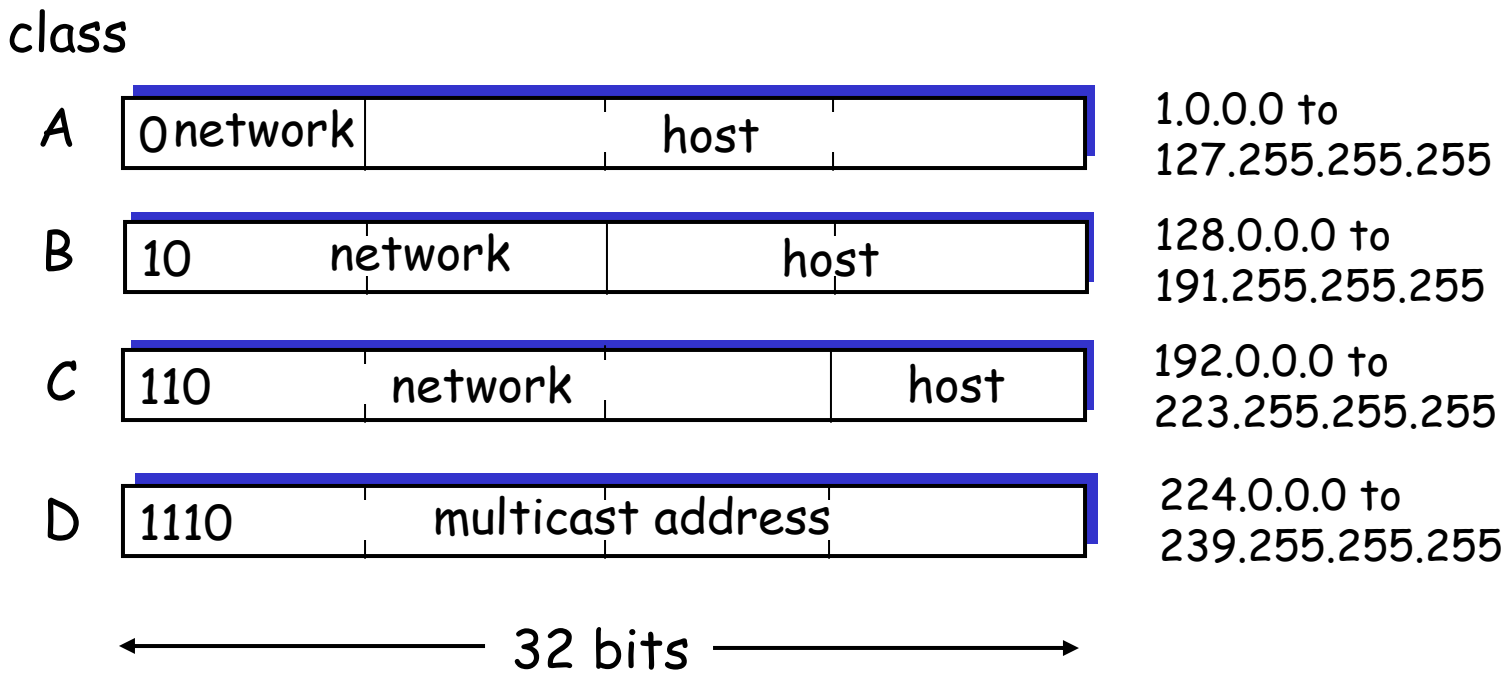
how much overhead with TCP?

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

IP Addresses

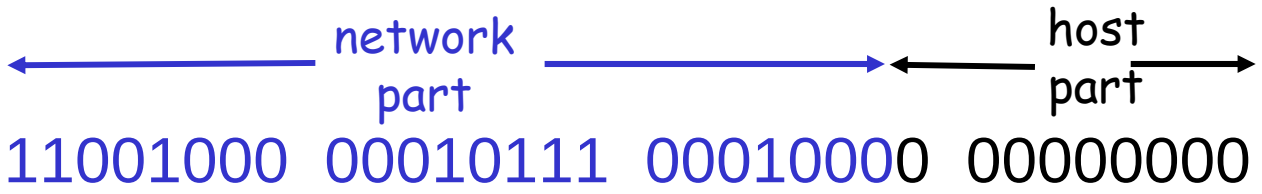
given notion of "network", let's re-examine IP addresses:

"class-full" addressing:



IP addressing: CIDR

- CIDR: Classless InterDomain Routing
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address
- "x" is often expressed as subnet mask



200.23.16.0/23

Subnet number	Subnet mask
200.23.16.0	255.255.254.0

Overview

- Routing in the Internet
  - Intra-AS routing: RIP and OSPF
  - Inter-AS routing: BGP
- Multicast Routing

Some slides are in courtesy of J. Kurose and K. Ross

Routing in the Internet

- The Global Internet consists of Autonomous Systems (AS) interconnected with each other:
  - **Stub AS**: small corporation: one connection to other AS's
  - **Multihomed AS**: large corporation (no transit): multiple connections to other AS's
  - **Transit AS**: provider, hooking many AS's together

- Two-level routing:
  - **Intra-AS**: administrator responsible for choice of routing algorithm within network
  - **Inter-AS**: unique standard for inter-AS routing: BGP

OSPF (Open Shortest Path First)

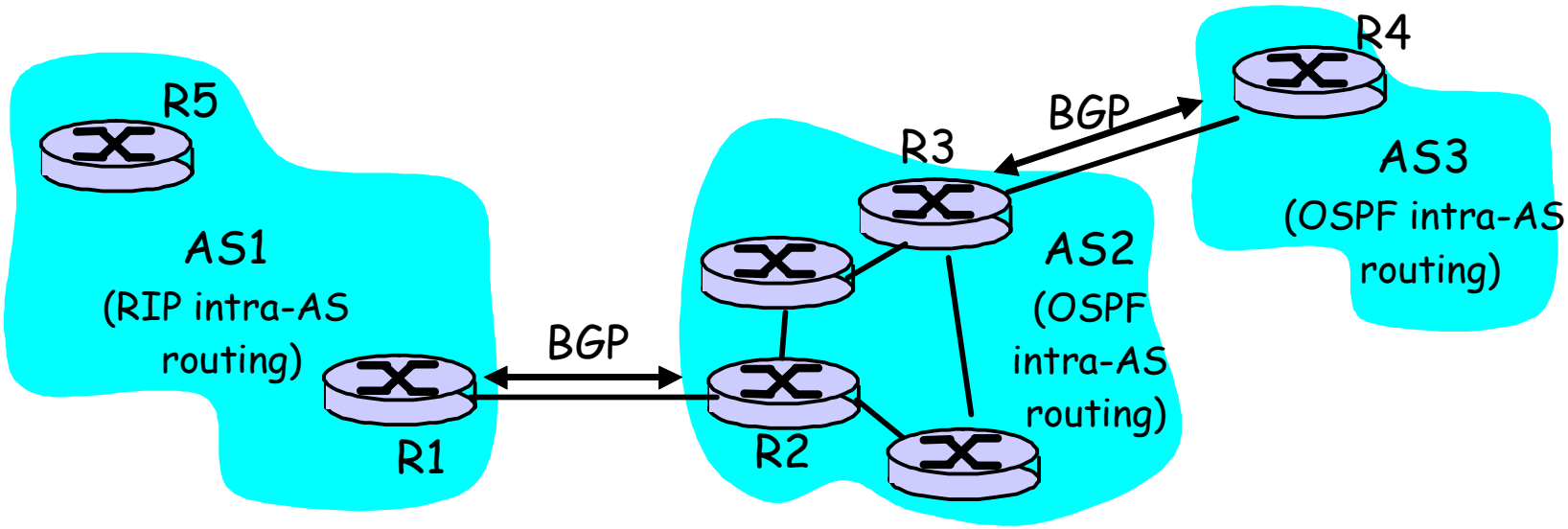
- Uses Link State algorithm
  - LS packet dissemination
  - Topology map at each node
  - Route computation using Dijkstra's algorithm

- OSPF advertisement carries one entry per neighbor router
- Advertisements disseminated to entire AS (via flooding)
- Broadcast link states
  - Whenever there is a link state change
  - Periodically (at least every 30 minutes)

Hierarchical OSPF

- Two-level hierarchy: local area, backbone.
  - Link-state advertisements only in area
  - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.
- **Area border routers:** "summarize" distances to nets in own area, advertise to other Area Border routers.
- **Backbone routers:** run OSPF routing limited to backbone.
- **Boundary routers:** connect to other AS's.

Inter-AS routing in the Internet: BGP



Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): *the* de facto standard
- **Path Vector** protocol:
  - similar to Distance Vector protocol
  - each Border Gateway broadcast to neighbors (peers) *entire path* (i.e., sequence of AS's) to destination
    - Enforce the policy specified
    - Detect loops
  - BGP routes to networks (ASs), not individual hosts
  - E.g., Gateway X may send its path to dest. Z:

Path (X,Z) = X,Y1,Y2,Y3,…,Z

Why different Intra- and Inter-AS routing ?

Policy:

- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

Topology Confidentiality:

- Intra-AS: complete topology info
- Inter-AS: only AS-level path, neighboring info

Performance:

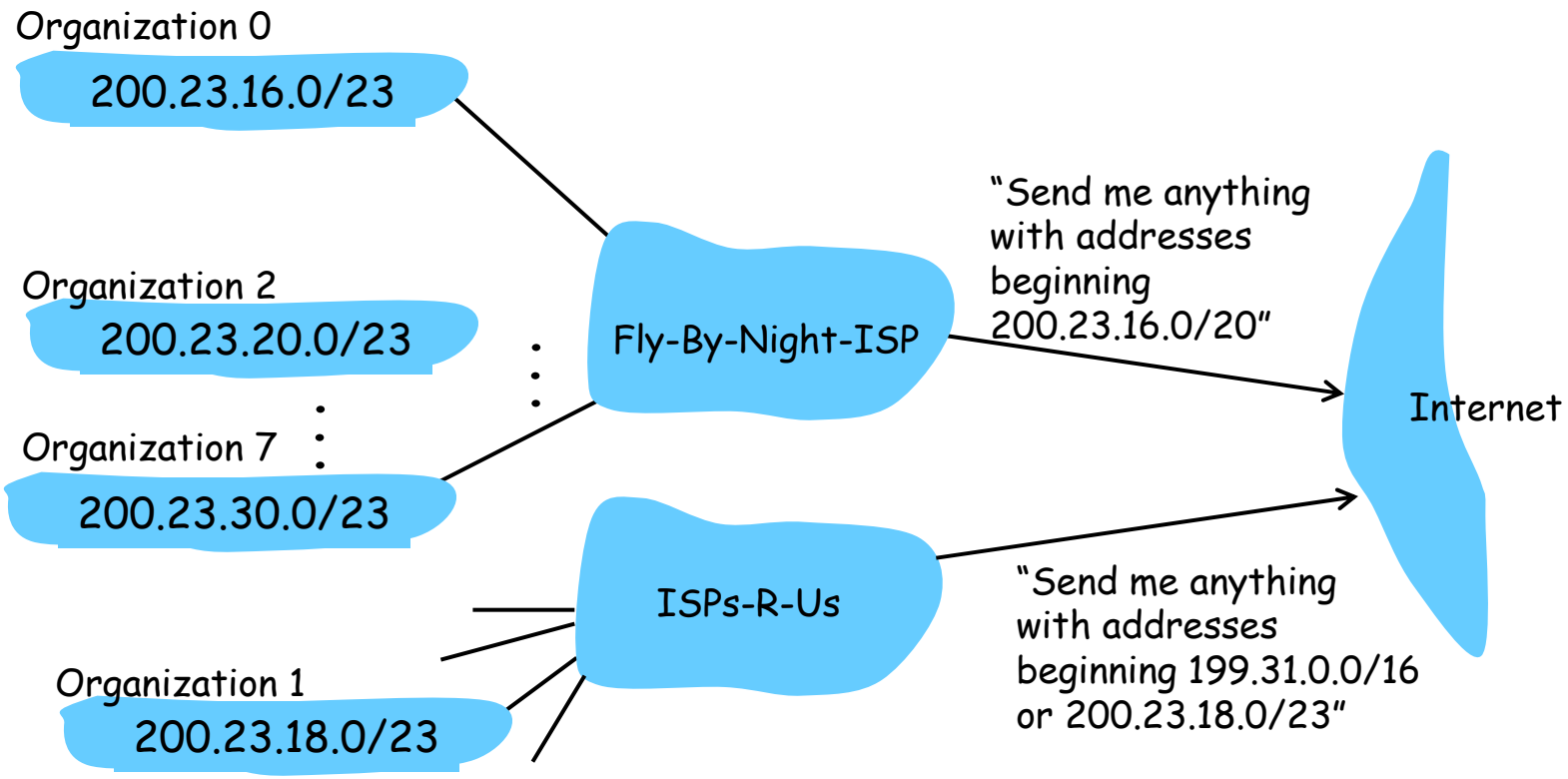
- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance

Scale:

- Inter-As require more scalability than interra-AS

Hierarchical addressing: more specific routes

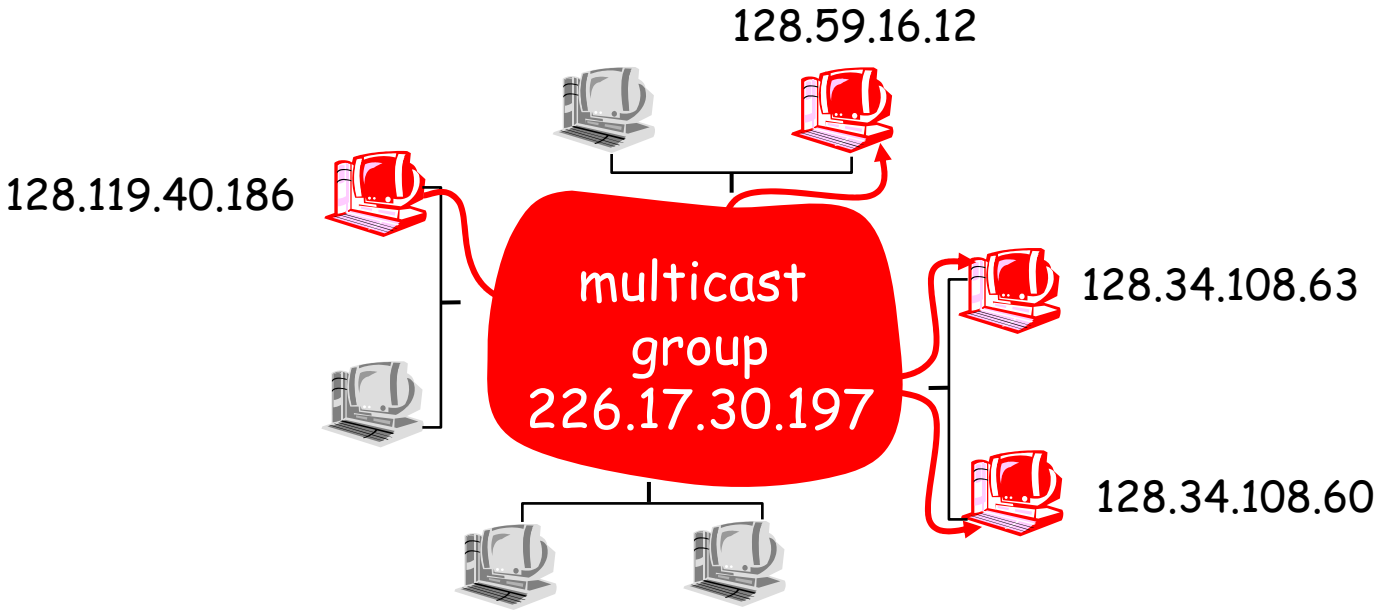
ISPs-R-Us has a more specific route to Organization 1
Longest prefix match routing!



Review

- Multicast Routing
  - Three options
    - IP-layer multicast
    - Unicast
    - App-layer multicast
  - source-based tree: one tree per source
    - shortest path trees
    - reverse path forwarding
  - group-shared tree: group uses one tree
    - minimal spanning (Steiner)
    - center-based trees

Internet Multicast Service Model

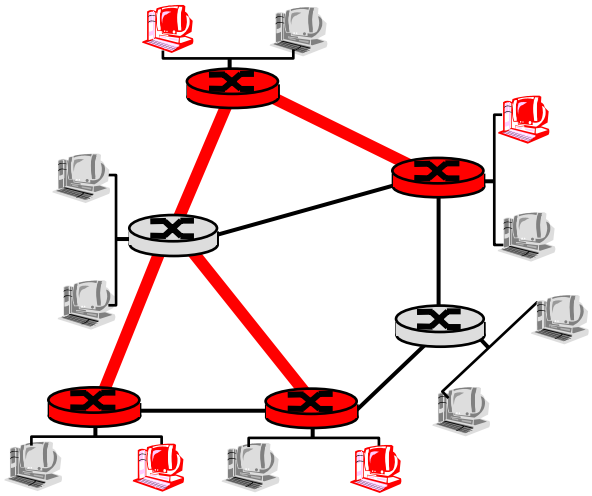


multicast group concept: use of indirection

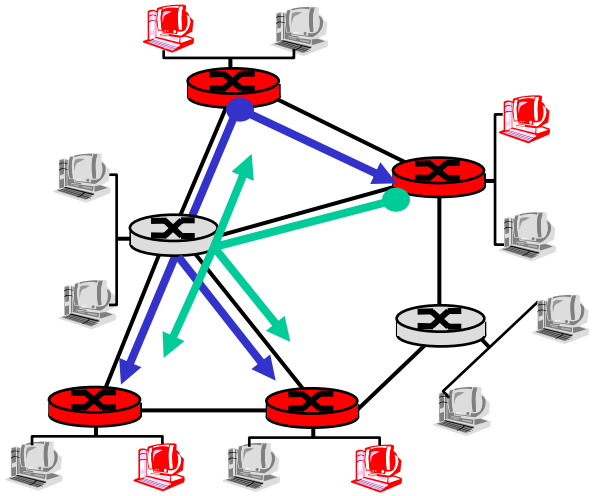
- hosts addresses IP datagram to multicast group
- routers forward multicast datagrams to hosts that have "joined" that multicast group

Multicast Routing: Problem Statement

- ***Goal:*** find a tree (or trees) connecting routers having local mcast group members
  - *tree:* not all paths between routers used
  - *source-based:* different tree from each sender to rcvrs
  - *shared-tree:* same tree used by all group members



Shared tree



Source-based trees

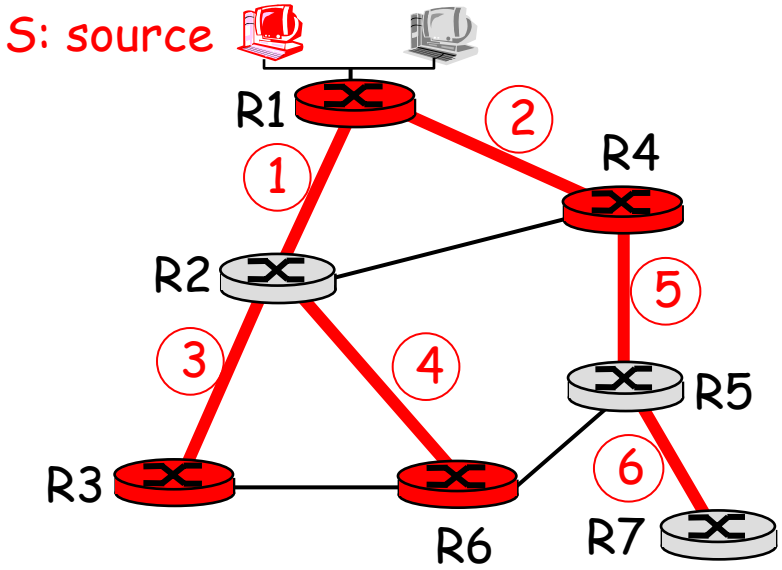
Approaches for building mcast trees

Approaches:

- source-based tree: one tree per source
  - shortest path trees
  - reverse path forwarding
- group-shared tree: group uses one tree
  - minimal spanning (Steiner)
  - center-based trees

Shortest Path Tree

- mcast forwarding tree: tree of shortest path routes from source to all receivers
  - Dijkstra's algorithm



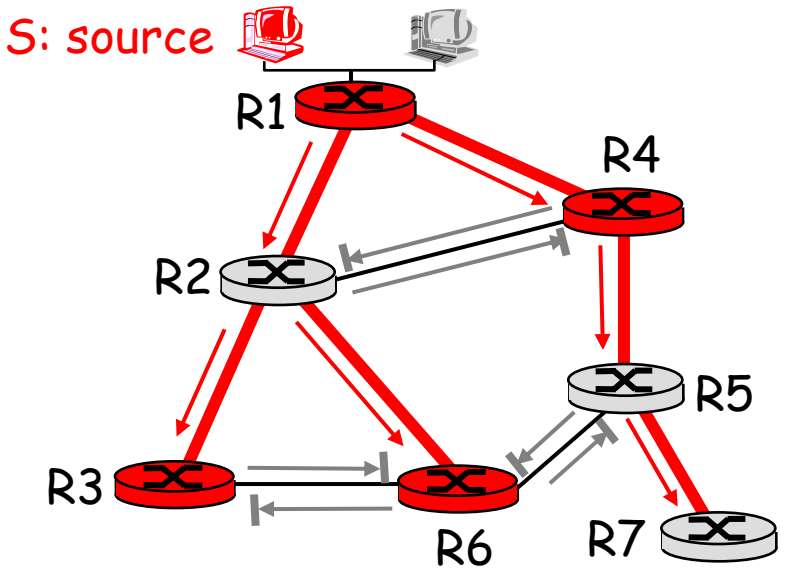
LEGEND

router with attached group member

router with no attached group member

i — link used for forwarding, i indicates order link added by algorithm

Reverse Path Forwarding: example



LEGEND

router with attached group member

router with no attached group member

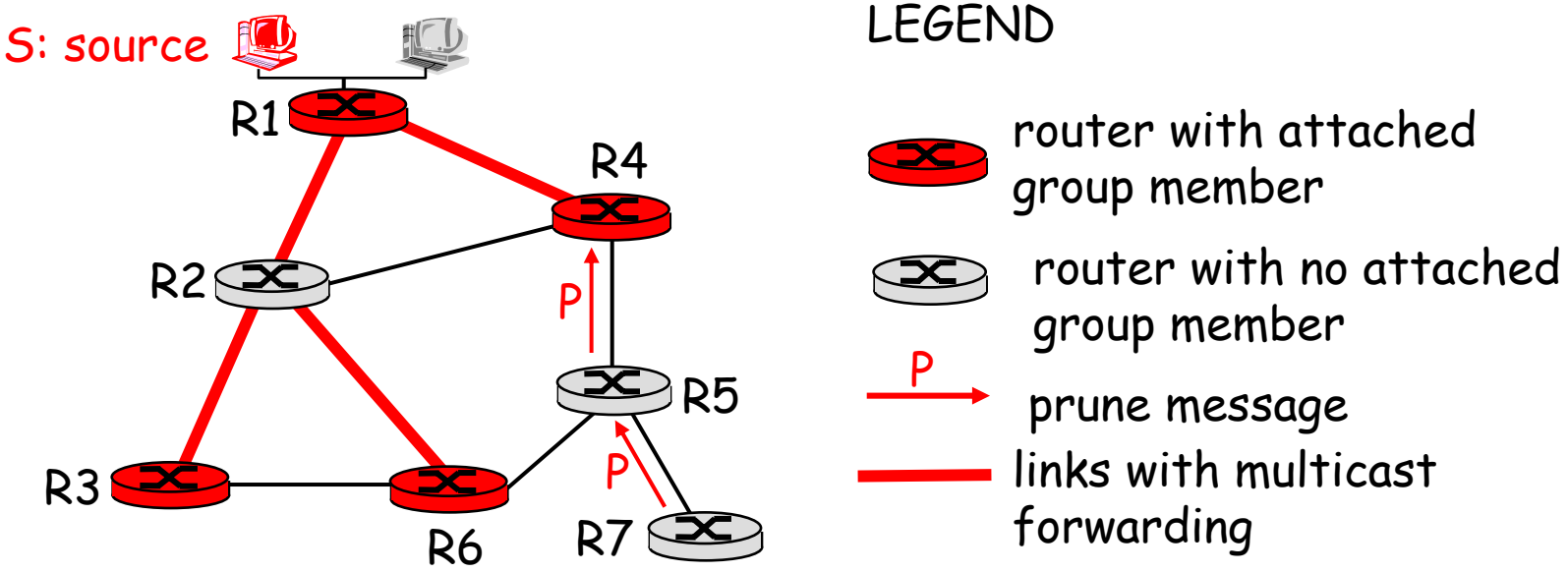
datagram will be forwarded

datagram will not be forwarded

- result is a source-specific *reverse* SPT
  - may be a bad choice with asymmetric links

Reverse Path Forwarding: pruning

- forwarding tree contains subtrees with no mcast group members
  - no need to forward datagrams down subtree
  - "prune" msgs sent upstream by router with no downstream group members

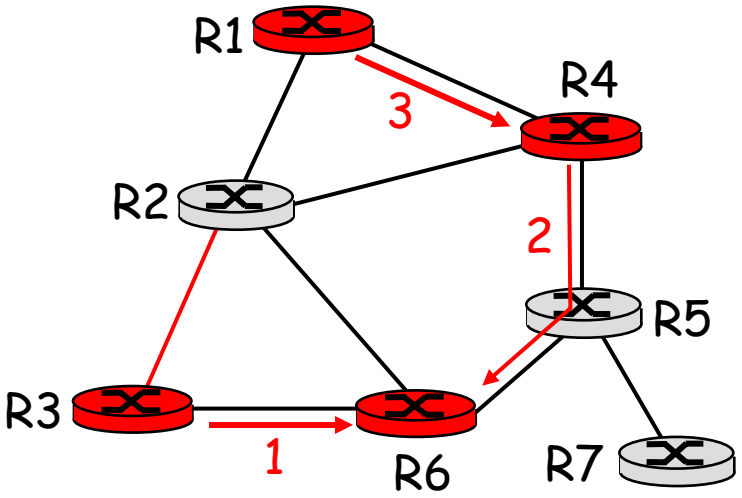


Shared-Tree: Steiner Tree

- Steiner Tree: minimum cost tree connecting all routers with attached group members
- problem is NP-complete
- excellent heuristics exists
- not used in practice:
  - computational complexity
  - information about entire network needed
  - monolithic: rerun whenever a router needs to join/leave

Center-based trees: an example

Suppose R6 chosen as center:



LEGEND

router with attached group member

router with no attached group member

1 path order in which join messages generated

Data Link Layer

- Datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Error Detection and Correction
  - Parity
  - Internet checksum
  - CRC
- Framing and Link Access
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - 'physical addresses' used in frame headers to identify source, dest
    - different from IP address!

Data Link Layer

- Datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Error Detection and Correction
  - Parity
  - Internet checksum
  - CRC
- Framing and Link Access
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - MAC addresses used in frame headers to identify source, dest
    - different from IP address!

MAC Protocols: a taxonomy

Three broad classes:

- Channel Partitioning
  - divide channel into smaller "pieces" (time slots, frequency, code) – TDMA, FDMA, CDMA
  - allocate piece to node for exclusive use
- Random access (MAC protocol) specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
  - Two options
    - slotted ALOHA
    - CSMA, CSMA/CD

Slotted ALOHA

Assumptions

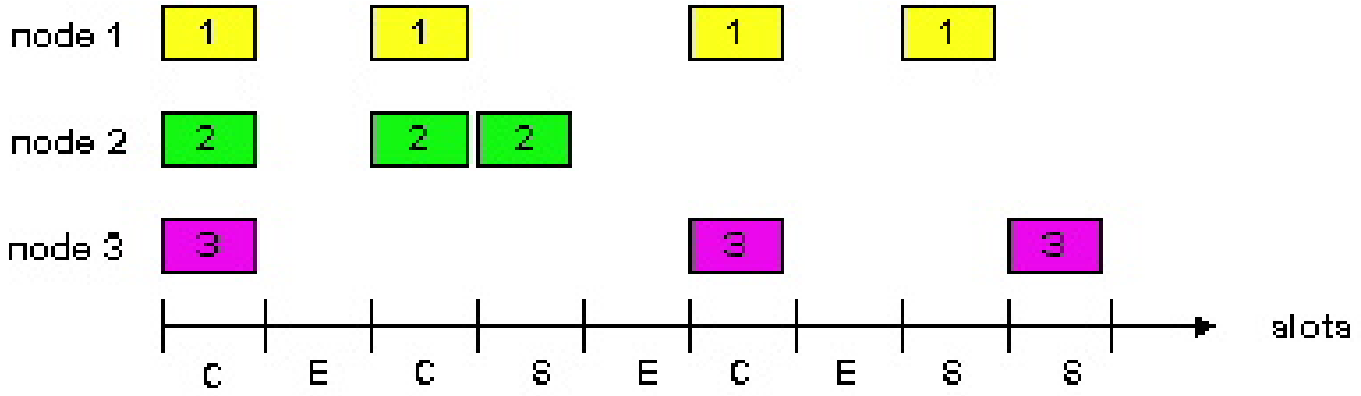
- all frames same size
- time is divided into equal size slots, time to transmit 1 frame
- nodes start to transmit frames only at beginning of slots
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation

- when node obtains fresh frame, it transmits in next slot
- no collision, node can send new frame in next slot
- if collision, node retransmits frame in each subsequent slot with prob. p until success

At best: channel used for useful transmissions 37% of time!

Slotted ALOHA



Pros

- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons

- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet

CSMA (Carrier Sense Multiple Access)

CSMA: listen before transmit:

- If channel sensed idle: transmit entire frame
- If channel sensed busy, defer transmission

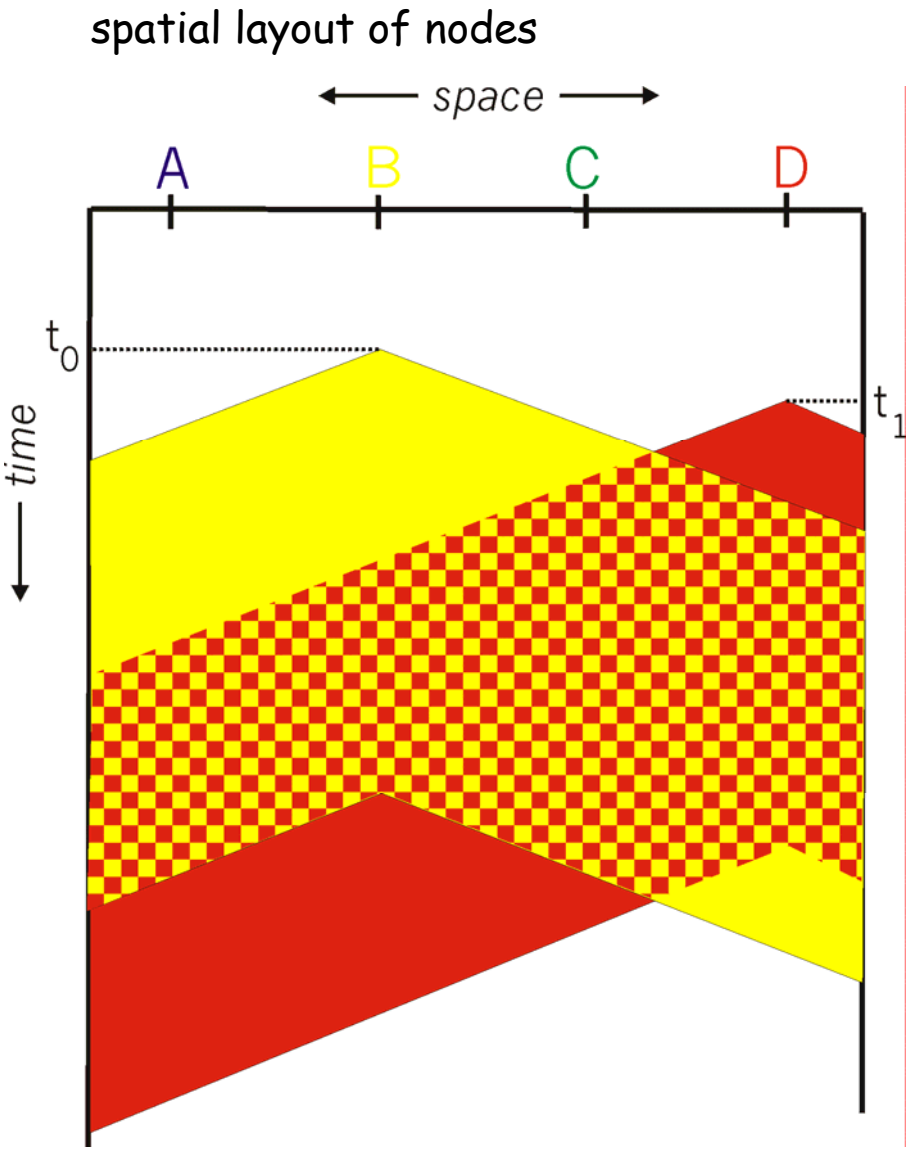
- Human analogy: don't interrupt others!

CSMA collisions

collisions *can* still occur:
propagation delay means
two nodes may not hear
each other's transmission

collision:
entire packet transmission
time wasted

note:
role of distance & propagation
delay in determining collision
probability



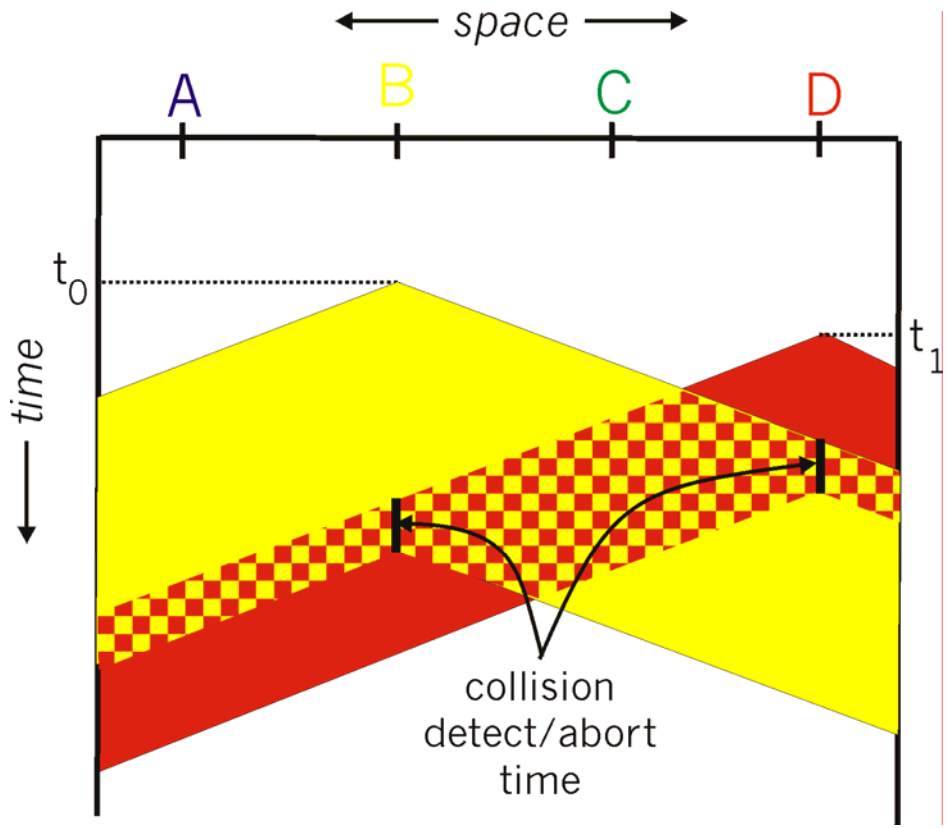
CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting

CSMA/CD collision detection



Ethernet uses CSMA/CD

- No slots
- adapter doesn't transmit if it senses that some other adapter is transmitting, that is, carrier sense
- transmitting adapter aborts when it senses that another adapter is transmitting, that is, collision detection
- Before attempting a retransmission, adapter waits a random time, that is, random access

Ethernet CSMA/CD algorithm

1. Adaptor gets datagram from and creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame !
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters **exponential backoff**: after the *m*th collision, adapter chooses a K at random from $\{0,1,2,\ldots,2^m-1\}$. Adapter waits K*512 bit times and returns to Step 2

Ethernet's CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 48 bits;

Bit time: 0.1 microsec for 10 Mbps Ethernet ;
for K=1023, wait time is about 50 msec

Exponential Backoff:

- *Goal*: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from {0,1}; delay is K x 512 bit transmission times
- after second collision: choose K from {0,1,2,3}…
- after ten collisions, choose K from {0,1,2,3,4,…,1023}

LAN Addresses and ARP

32-bit IP address:

- *network-layer* address
- used to get datagram to destination IP network (recall IP network definition)

LAN (or MAC or physical or Ethernet) address:

- used to get datagram from one interface to another physically-connected interface (same network)
- 48 bit MAC address (for most LANs) burned in the adapter ROM

LAN Address (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:

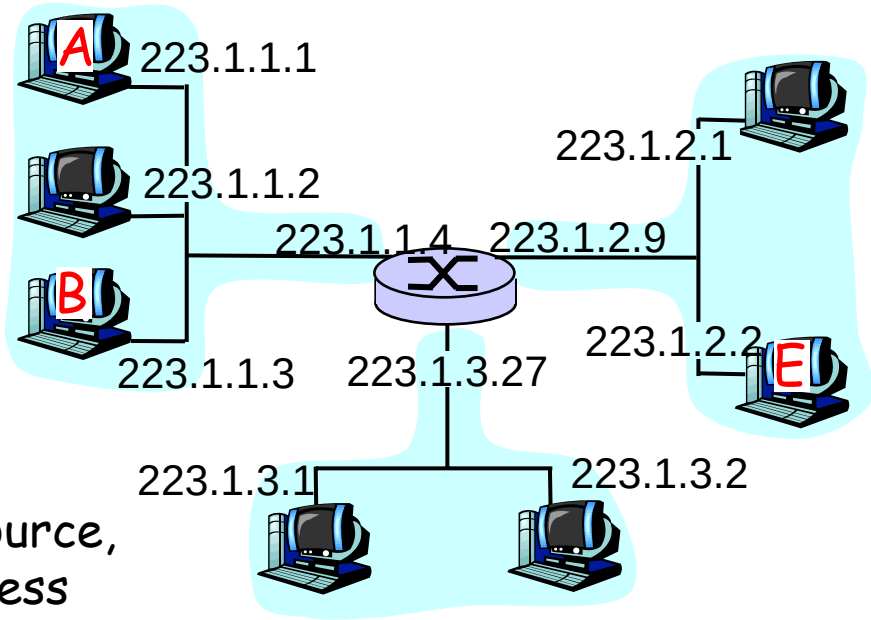
  (a) MAC address: like Social Security Number

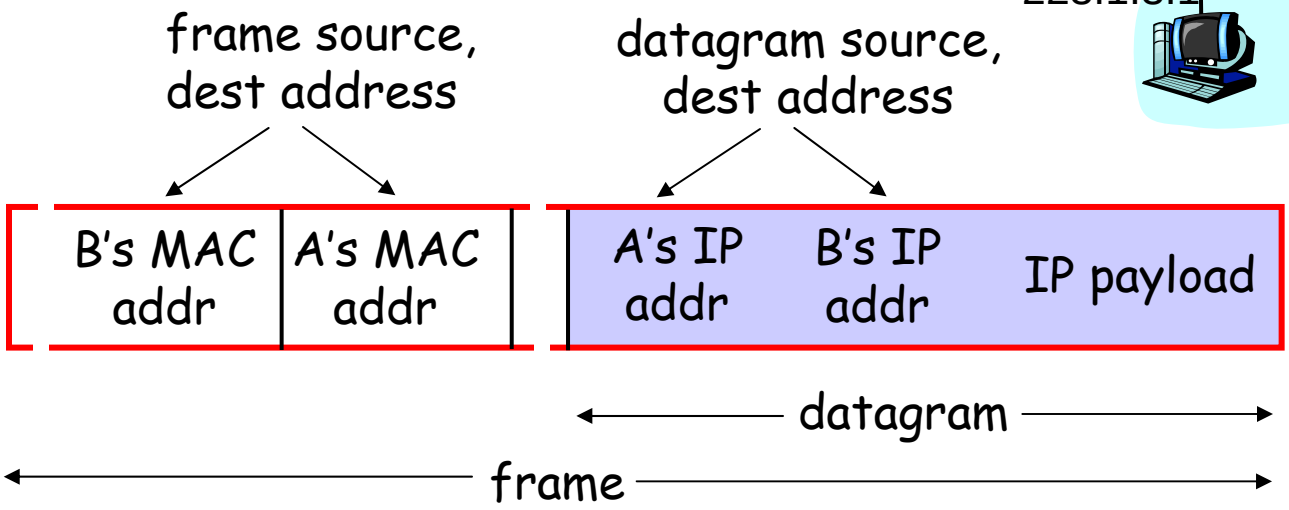
  (b) IP address: like postal address
- MAC flat address => portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - depends on IP network to which node is attached

Recall earlier routing discussion

Starting at A, given IP datagram addressed to B:

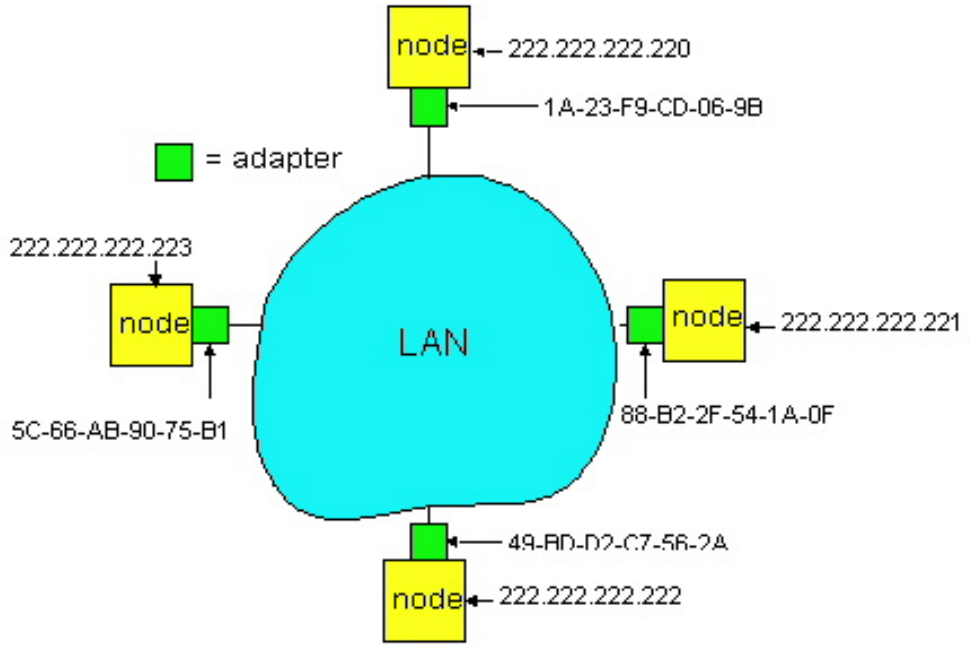
- look up net. address of B, find B on same net. as A
- link layer send datagram to B inside link-layer frame





ARP: Address Resolution Protocol

Question: how to determine MAC address of B knowing B's IP address?



- Each IP node (Host, Router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

  < IP address; MAC address; TTL>

  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

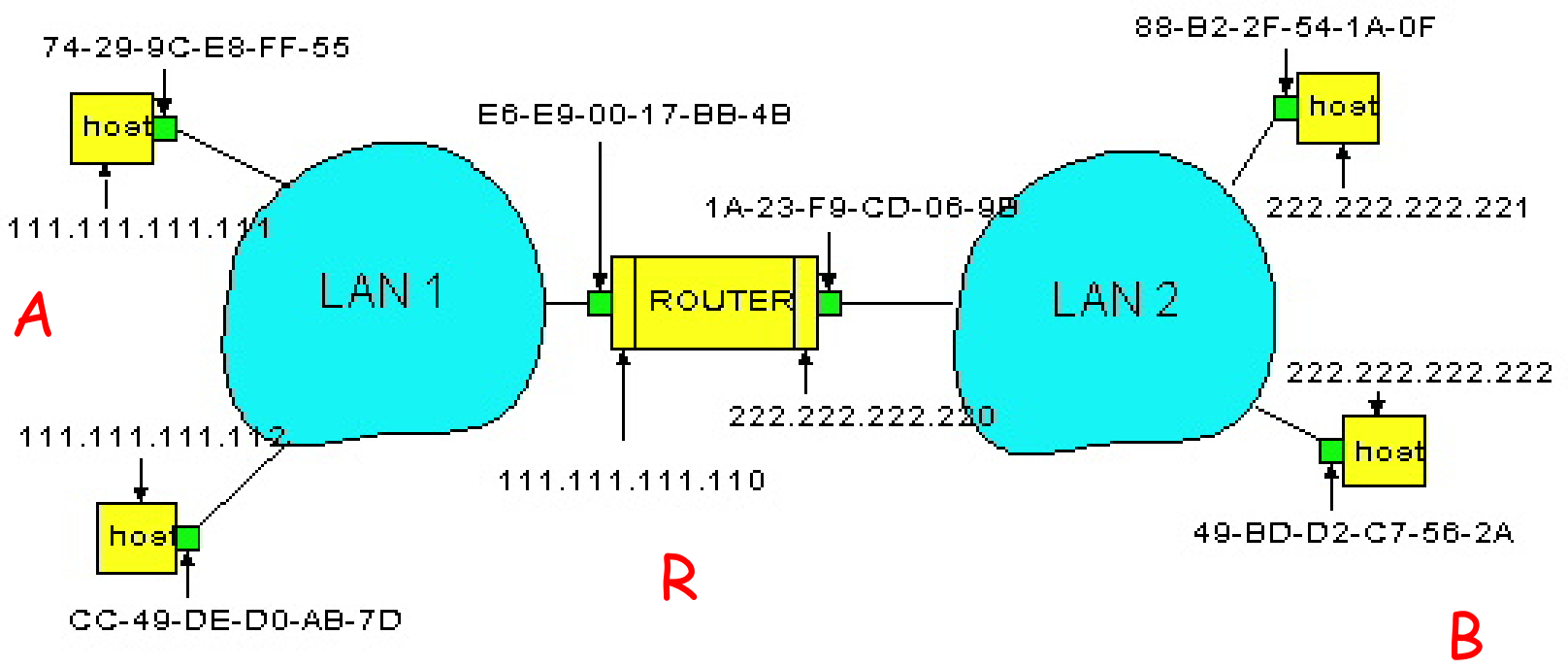
ARP protocol

- A wants to send datagram to B, and A knows B's IP address.
- Suppose B's MAC address is not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables without intervention from net administrator

Routing to another LAN

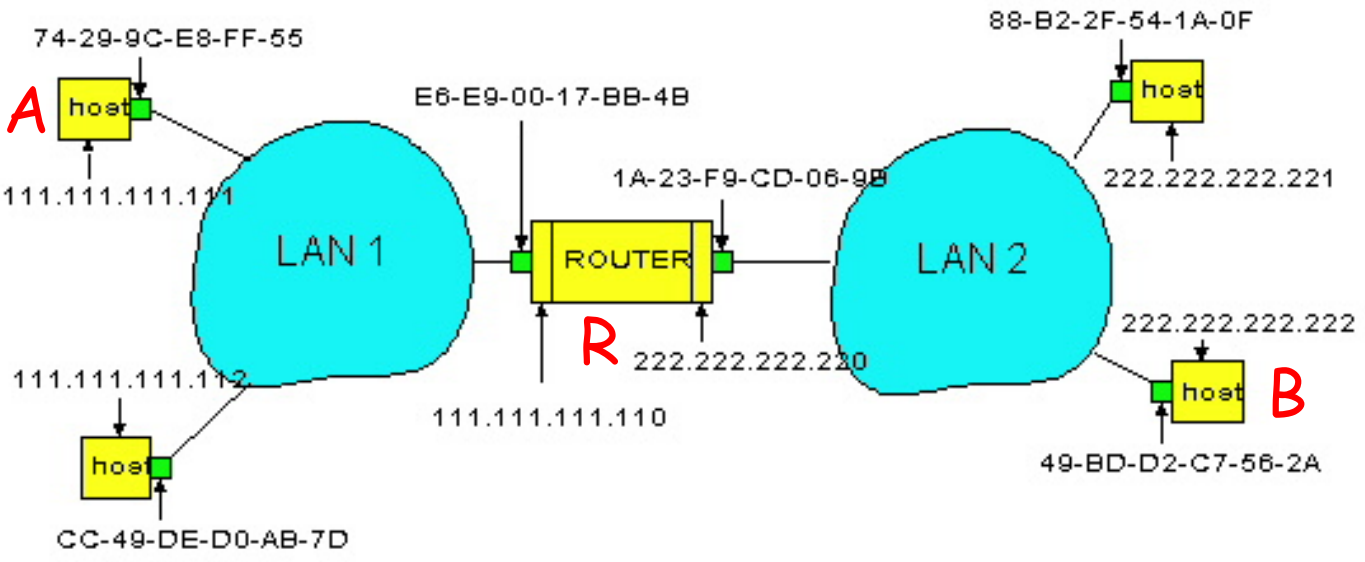
walkthrough: send datagram from A to B via R

assume A know's B IP address



- Two ARP tables in router R, one for each IP network (LAN)

- A creates datagram with source A, destination B
- A uses ARP to get R's MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
- A's data link layer sends frame
- R's data link layer receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's physical layer address
- R creates frame containing A-to-B IP datagram sends to B

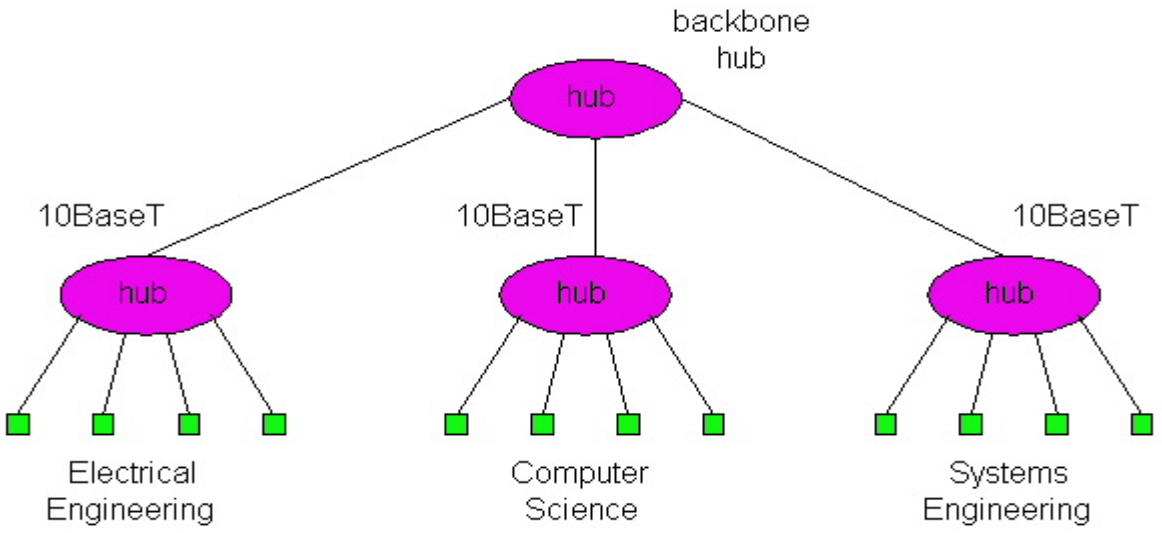


Interconnecting LAN segments

- Hubs
- Bridges
- Switches
  - Remark: switches are essentially multi-port bridges.
  - What we say about bridges also holds for switches!

Interconnecting with hubs

- Backbone hub interconnects LAN segments
- Physical layer devices
- Extends max distance between nodes
- But individual segment collision domains become one large collision domian
  - if a node in CS and a node EE transmit at same time: collision
- Can't interconnect 10BaseT & 100BaseT



Bridges

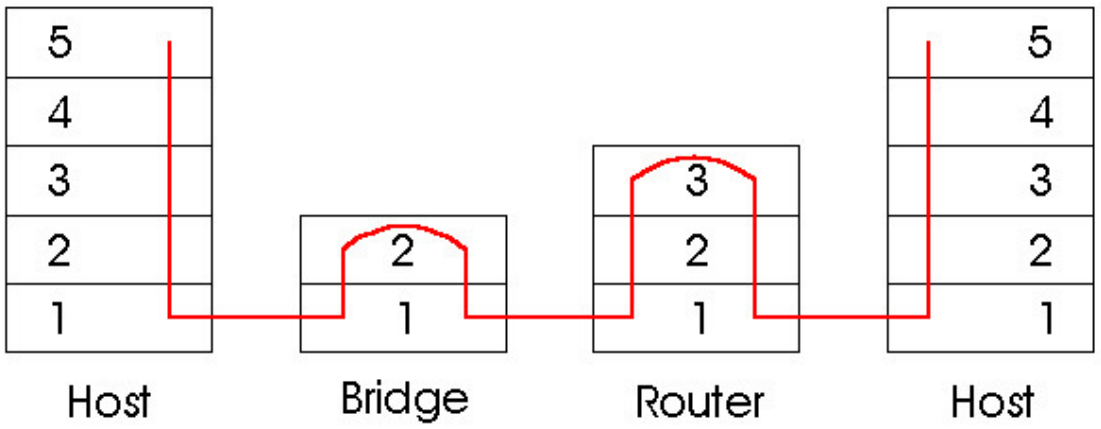
- Link layer device
  - stores and forwards Ethernet frames
  - examines frame header and selectively forwards frame based on MAC dest address
  - when frame is to be forwarded on segment, uses CSMA/CD to access segment
- transparent
  - hosts are unaware of presence of bridges
- plug-and-play, self-learning
  - bridges do not need to be configured

Self learning

- A bridge has a bridge table
- entry in bridge table:
  - (Node LAN Address, Bridge Interface, Time Stamp)
  - stale entries in table dropped (TTL can be 60 min)
- bridges *learn* which hosts can be reached through which interfaces
  - when frame received, bridge "learns" location of sender: incoming LAN segment
  - records sender/location pair in bridge table

Bridges vs. Routers

- both store-and-forward devices
  - routers: network layer devices (examine network layer headers)
  - bridges are link layer devices
- routers maintain routing tables, implement routing algorithms
- bridges maintain bridge tables, implement filtering, learning and spanning tree algorithms



Routers vs. Bridges

Bridges + and -

+ Bridge operation is simpler requiring less packet processing
+ Bridge tables are self learning
- All traffic confined to spanning tree, even when alternative bandwidth is available
- Bridges do not offer protection from broadcast storms

Routers vs. Bridges

Routers + and -

+ arbitrary topologies can be supported, cycling is limited by TTL counters (and good routing protocols)
+ provide protection against broadcast storms
- require IP address configuration (not plug and play)
- require higher packet processing

- bridges do well in small (few hundred hosts) while routers used in large networks (thousands of hosts)

- What about Hubs vs. Bridges?

Summary comparison

	hubs	bridges	routers	switches
traffic isolation	no	yes	yes	yes
plug & play	yes	yes	no	yes
optimal routing	no	no	yes	no

Network Security

- What is network security?
- Principles of cryptography
  - Symmetric Key
  - Public Key
- Authentication
  - Protocol evolution
- Access control: firewalls
- Attacks and counter measures
  - Packet sniffing
  - IP spoofing
  - DoS attacks