The Network Layer

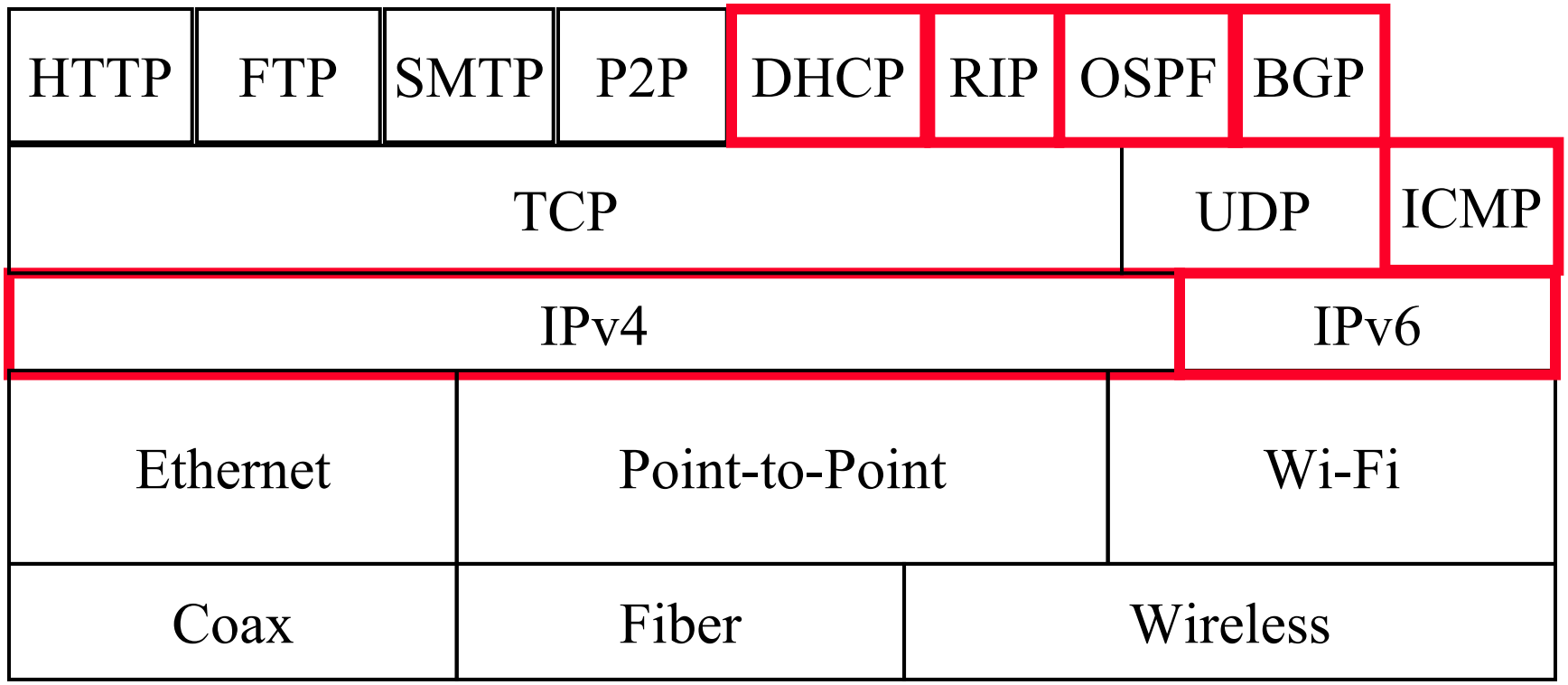
1. Network Layer Basics
2. Forwarding Protocols: IPv4, ICMP, DHCP, NAT, IPv6
3. Routing Algorithms: Link-State, Distance Vector
4. Routing Protocols: RIP, OSPF, BGP

Note: This class lecture is based on Chapter 4 of the textbook (Kurose and Ross) and the figures provided by the authors.

Network Layer Basics

1. Forwarding and Routing
2. Connection Oriented Networks: ATM Networks
3. Classes of Service
4. Router Components
5. Packet Queuing and Dropping

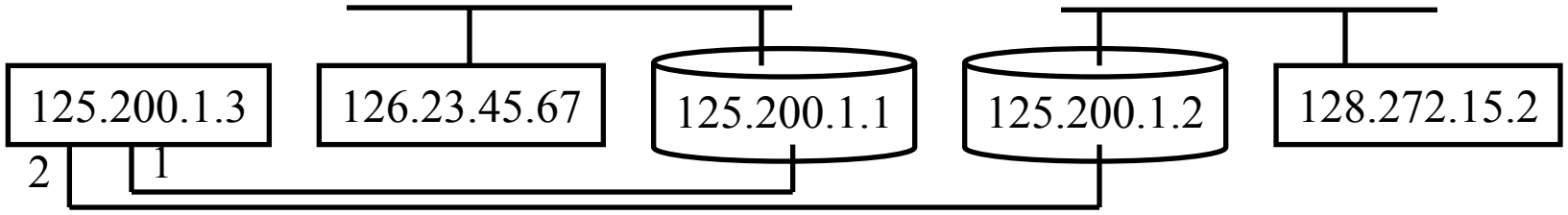
Network Layer Protocols



- Forwarding: IPv4 and IPv6
- Routing: RIP, OSPF, BGP, …
- Control and Management: ICMP, DHCP, …

Forwarding and Routing

- **Forwarding**: Input link to output link via Address prefix lookup.
- **Routing**: Making the Address lookup table
- **Longest Prefix Match**



Prefix	Next Router	Interface
126.23.45.67/32	125.200.1.1	1
128.272.15/24	125.200.1.2	2
128.272/16	125.200.1.1	1

Prefix Match	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

Suppose the packet's destination address is 11001000 00010111 00010110 10100001; because the 21-bit prefix of this address matches the first entry in the table, the router forwards the packet to link interface 0. If a prefix doesn't match any of the first three entries, then the router forwards the packet to interface 3.

The first 24 bits of the address 11001000 00010111 00011000 10101010 match the second entry in the table, and the first 21 bits of the address match the third entry in the table. When there are multiple matches, the router uses the **longest prefix matching rule**; that is, it finds the longest matching entry in the table and forwards the packet to the link interface associated with the longest prefix match.

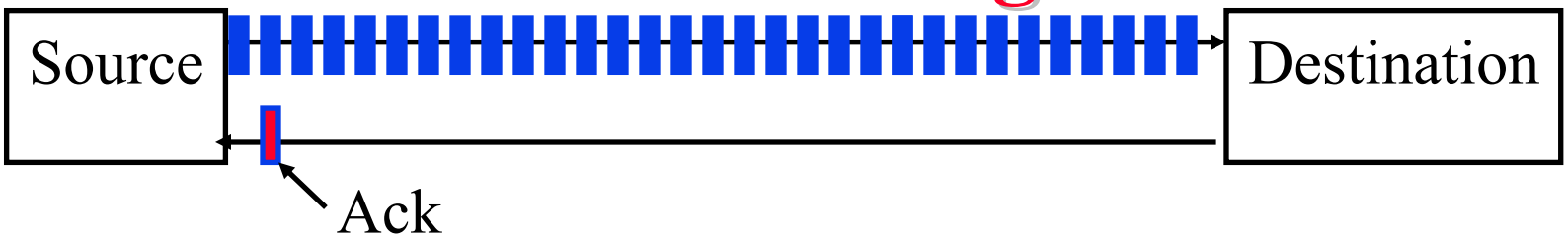
"Route Print" Command in Windows

MAC: netstat -rn

```
===========================================================================
Interface List
0x1 . . ....................... MS TCP Loopback interface
0x2 ...00 16 eb 05 af c0 .
 .... Intel(R) WiFi Link 5350 - Packet Scheduler Miniport
0x3 ...00 1f 16 15 7c 41 .
 Intel(R) 82567LM Gigabit Network Connection - Packet Scheduler Miniport
0x40005 ...00 05 9a 3c 78 00 .
 .. Cisco Systems VPN Adapter - Packet Scheduler Miniport
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
      169.254.0.0      255.255.0.0    192.168.0.106    192.168.0.106       20
      192.168.0.0    255.255.255.0    192.168.0.106    192.168.0.106       10
      192.168.0.0    255.255.255.0    192.168.0.108    192.168.0.108       10
    192.168.0.106  255.255.255.255        127.0.0.1        127.0.0.1       10
    192.168.0.108  255.255.255.255        127.0.0.1        127.0.0.1       10
    192.168.0.255  255.255.255.255    192.168.0.106    192.168.0.106       10
    192.168.0.255  255.255.255.255    192.168.0.108    192.168.0.108       10
        224.0.0.0        240.0.0.0    192.168.0.106    192.168.0.106       10
        224.0.0.0        240.0.0.0    192.168.0.108    192.168.0.108       10
  255.255.255.255  255.255.255.255    192.168.0.106    192.168.0.106       1
  255.255.255.255  255.255.255.255    192.168.0.106             40005       1
  255.255.255.255  255.255.255.255    192.168.0.108    192.168.0.108       1
Default Gateway:       192.168.0.1
===========================================================================
Persistent Routes:
  None
```

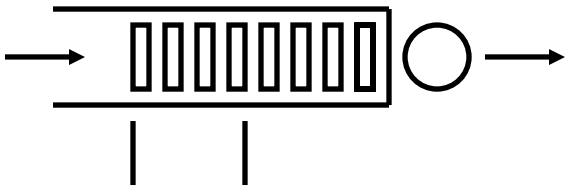
Note: 127.0.0.1 = Local Host, 224.x.y.z = Multicast on local LAN

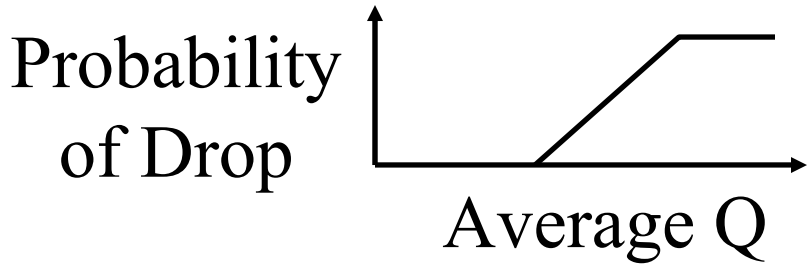
Ideal Buffering



- Flow Control Buffering = RTT*Transmission Rate
- Buffer = RTT*Transmission Rate/√(# of TCP flows)

Packet Dropping Policies



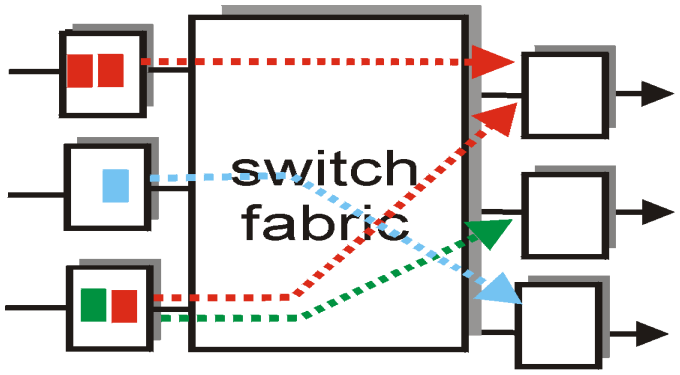


- **Drop-Tail**: Drop the arriving packet
- **Random Early Drop (RED):** Drop arriving packets even before the queue is full
  - Routers measure average queue and drop incoming packet with certain probability

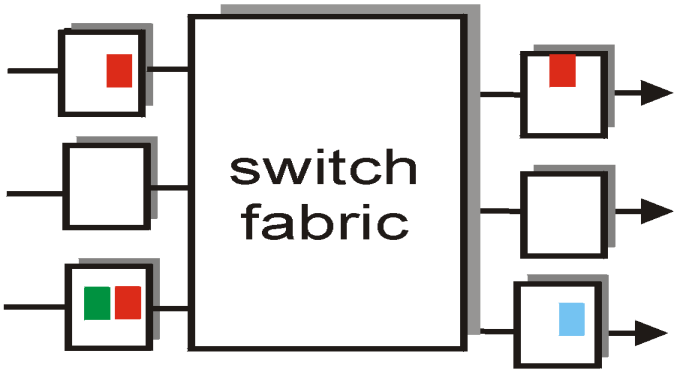
  ⇒ **Active Queue Management** (AQM)

Head-of-Line Blocking

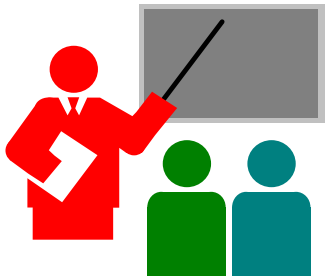
- Packet at the head of the queue is waiting
  ⇒ Other packets can not be forwarded even if they are going to other destination



output port contention
at time t - only one red
packet can be transferred



green packet
experiences HOL blocking



Network Layer Basics: Review

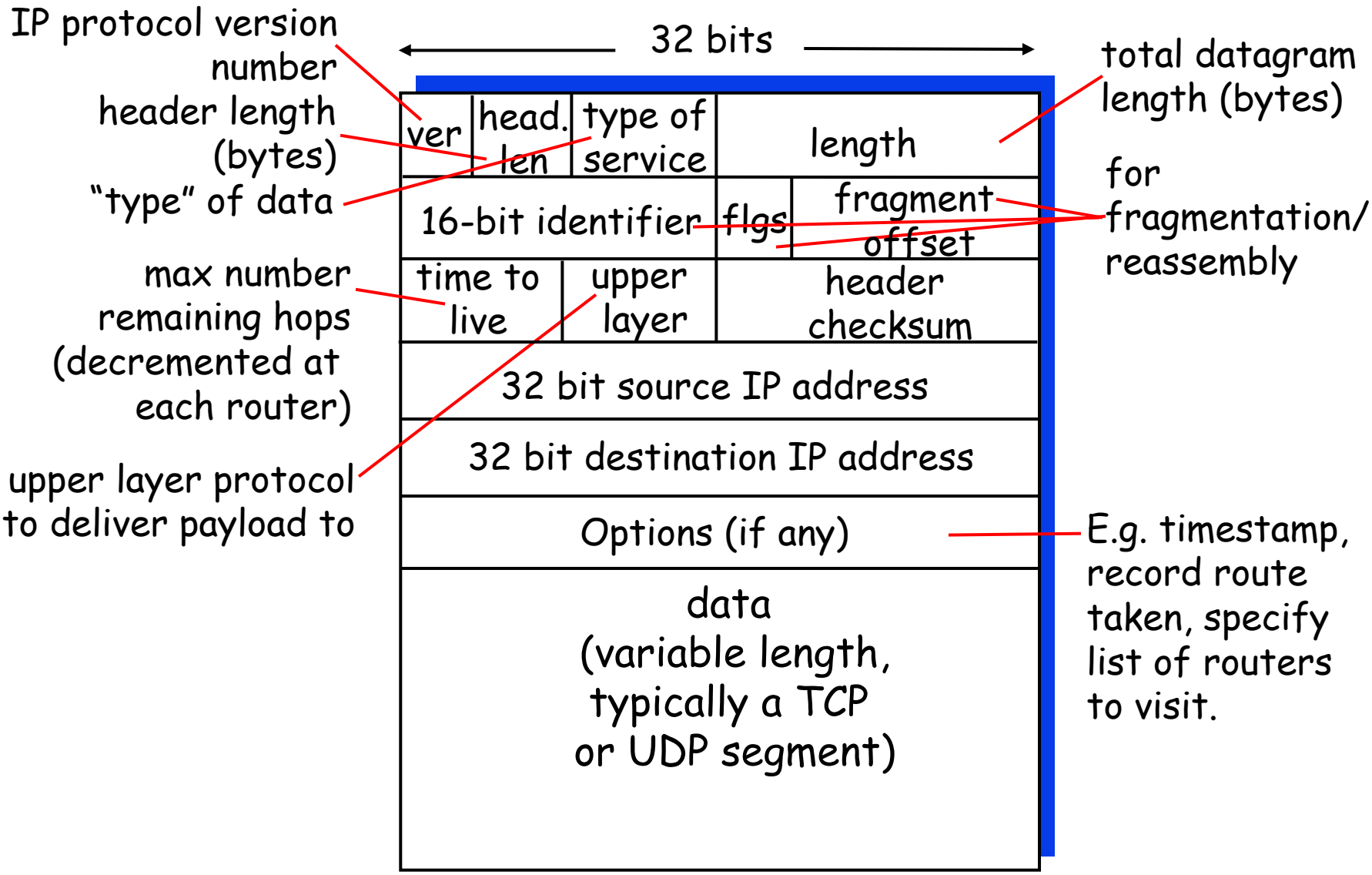
1. Forwarding uses routing table to find output port for datagrams using longest prefix match. Routing protocols make the table.

2. IP provides only best effort service (KISS).
3. Routers consist of input/output ports, switching fabric, and processors.
4. Datagrams may be dropped even if the queues are not full (Random early drop).
5. Queueing at input may result in head of line blocking.

Forwarding Protocols

1. IPv4 Datagram Format
2. IP Fragmentation and Reassembly
3. IP Addressing
4. Network Address Translation (NAT)
5. Universal Plug and Play
6. Dynamic Host Control Protocol (DHCP)
7. ICMP
8. IPv6

IP Datagram Format



IP Fragmentation Fields

- Data Unit Identifier (ID)
  - Sending host puts an identification number in each datagram
- Total length: Length of user data plus header in octets
- Data Offset - Position of fragment in original datagram
    - In multiples of 64 bits (8 octets)
- *More fragments* flag
    - Indicates that this is not the last fragment
- Datagrams can be fragmented/refragmented at any router
- Datagrams are reassembled only at the destination host

IP Fragmentation and Reassembly

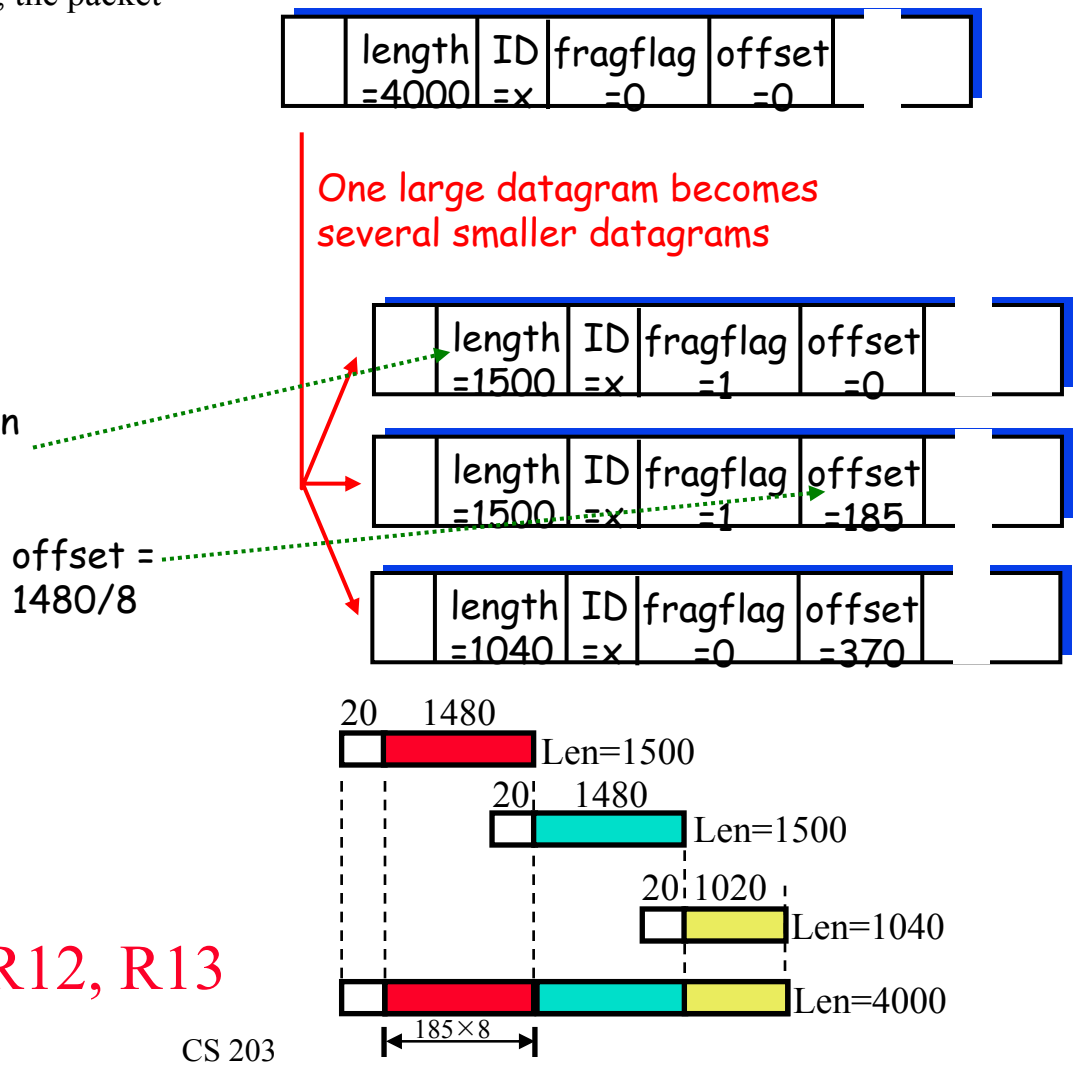
If an IP packet that is larger than the Maximum Transmission Unit (MTU) of an interface, the packet must either be fragmented.

Example

- 4000 byte datagram
- MTU = 1500 bytes

There is a 20 byte header in each packet. So the original packet contains 3,980 bytes of data. The fragments contain 1480, 1480, and 1020 bytes of data. 1480 + 1480 + 1020 = 3980

More-fragments flag (MF) =1; more fragments of this packet follow



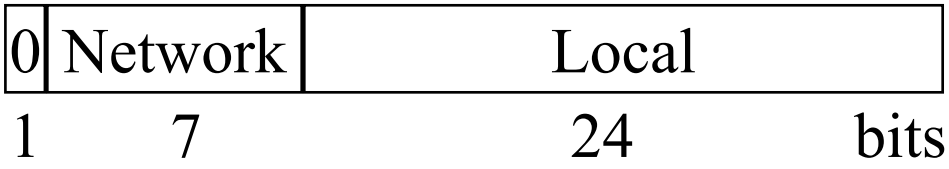
Ref: Section 4.4.1. Try R12, R13

Homework 4B

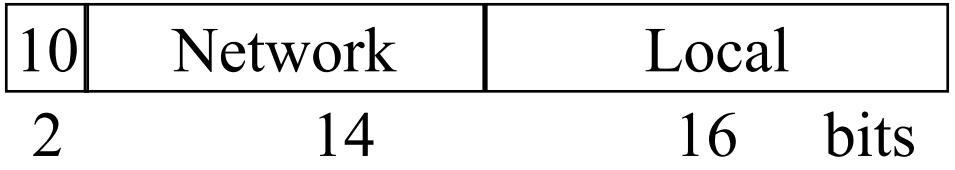
- Consider sending a 2400-byte datagram into a link that has an MTU of 720 bytes. Suppose the original datagram is stamped with the identification number 422. How many fragments are generated? What are the values in the various fields in the IP datagram(s) generated related to fragmentation?

IP Address Classes

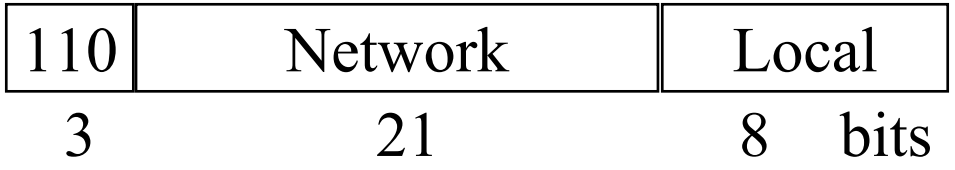
- Class A:



- Class B:

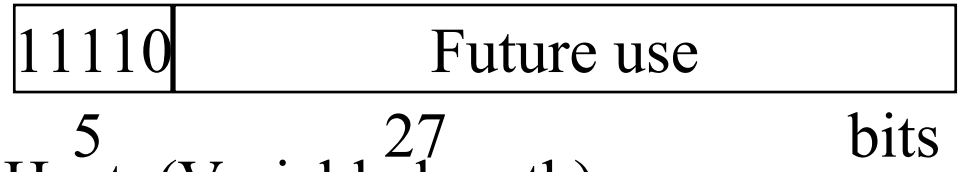


- Class C:

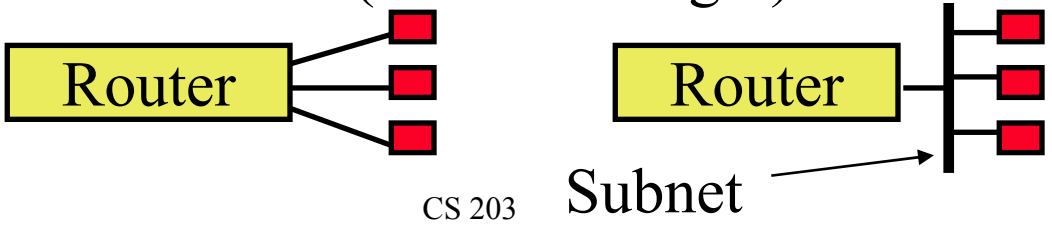


- Class D:

- Class E:



- Local = Subnet + Host (Variable length)



IP address classes

Class	1st Octet Decimal Range	1st Octet High Order Bits	Network/Host ID (N=Network, H=Host)	Default Subnet Mask	Number of Networks	Hosts per Network (Usable Addresses)
A	1 – 126*	0	N.H.H.H	255.0.0.0	126 ($2^7 - 2$)	16,777,214 ($2^{24} - 2$)
B	128 – 191	10	N.N.H.H	255.255.0.0	16,382 ($2^{14} - 2$)	65,534 ($2^{16} - 2$)
C	192 – 223	110	N.N.N.H	255.255.255.0	2,097,150 ($2^{21} - 2$)	254 ($2^8 - 2$)
D	224 – 239	1110	Reserved for Multicasting			
E	240 – 254	1111	Experimental; used for research			

Note: Class A addresses 127.0.0.0 to 127.255.255.255 cannot be used and is reserved for loopback and diagnostic functions.

Private IP Addresses

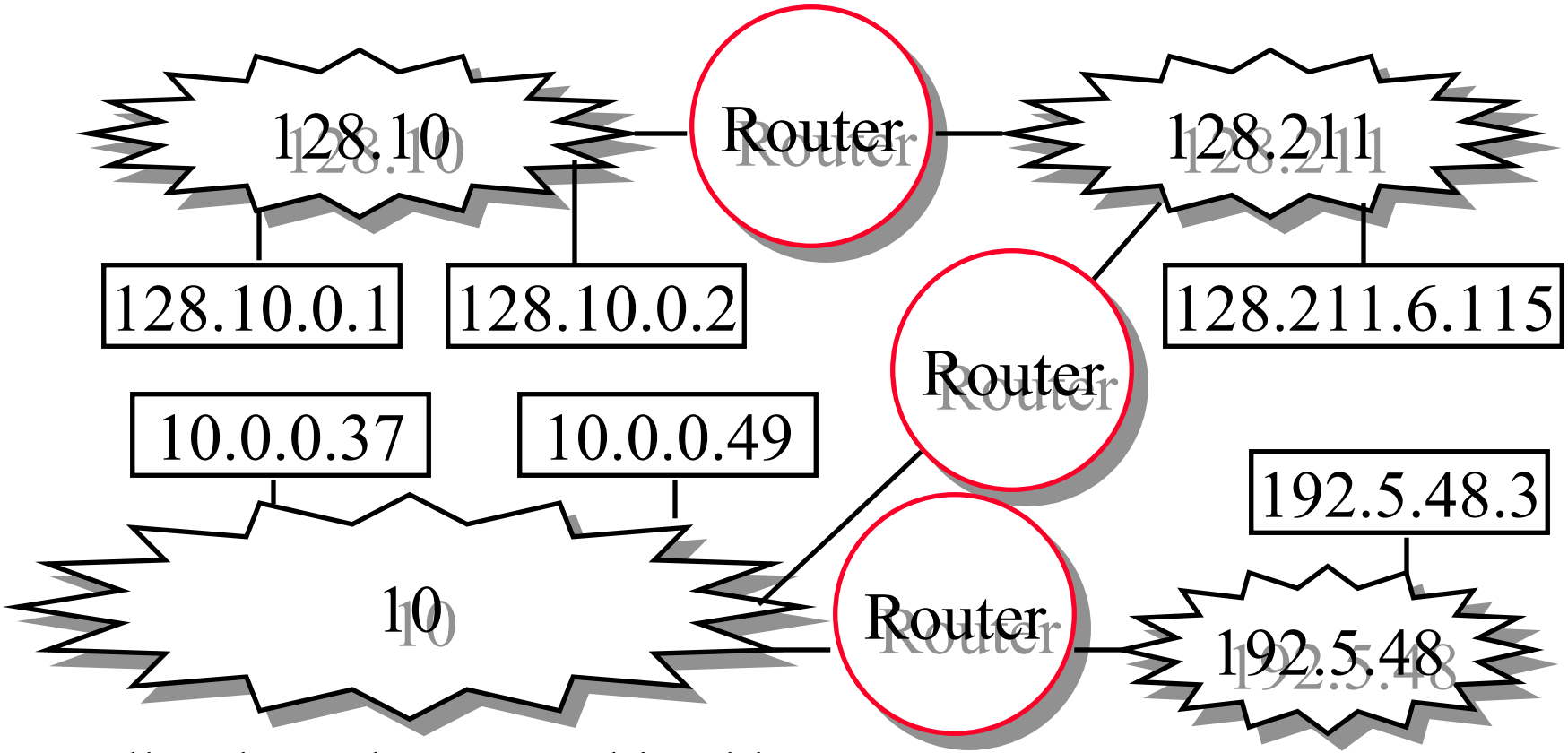
Class	Private Networks	Subnet Mask	Address Range
A	10.0.0.0	255.0.0.0	10.0.0.0 - 10.255.255.255
B	172.16.0.0 - 172.31.0.0	255.240.0.0	172.16.0.0 - 172.31.255.255
C	192.168.0.0	255.255.0.0	192.168.0.0 - 192.168.255.255

A netmask (sometime called prefix length) is a 32-bit number that masks an IP address, and divides the IP address into network address and host address. In a netmask, two bits are always automatically assigned. For example, in 255.255.225.0, “0” is the assigned network address. In 255.255.255.255, “255” is the assigned broadcast address. The 0 and 255 are always assigned and cannot be used.

Total #of networks = 2^(netmask length - # of used segments) – 2
For example, if we used a netmask length of 24, having a netmask of 255.255.255.0 with 3 used segments, 2,097,150 total number of networks can be possible.

Total of number hosts=2^(# of zeroes) - 2
For example, with a netmask length of 24, as shown in the above example, there are 8 zeroes. Therefore, 254 total number of hosts can be possible; 2 is subtracted from this number to account for the broadcast and network addresses.

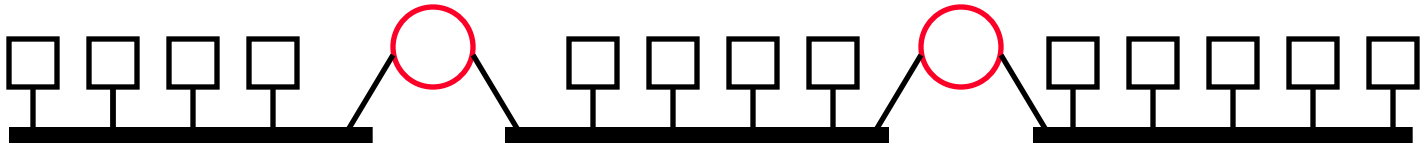
IP Addressing



- All IP hosts have a 32-bit address.128.10.0.1 =1000 0000 0000 1010 0000 0000 0000 0001
- All hosts on a network have the same network prefix

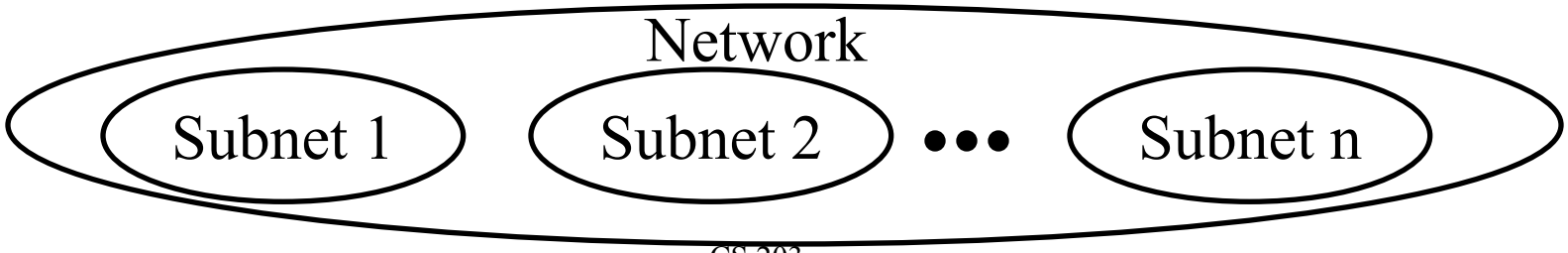
The dividing a network into two or more networks is called subnetting

Subnetting



- All hosts on a subnetwork have the same prefix. Position of the prefix is indicated by a "subnet mask"
- Example: First 23 bits = subnet

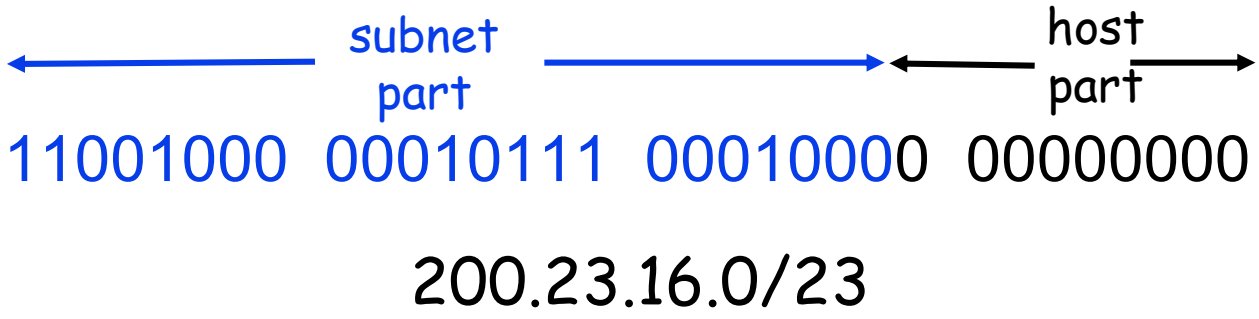
  Address: 10010100 10101000 00010000 11110001
  Mask: 11111111 11111111 11111110 00000000
  .AND. 10010100 10101000 00010000 00000000



IP addressing: CIDR

CIDR (Classless Inter-Domain Routing, sometimes called supernetting) is a way to allow more flexible allocation of Internet Protocol (IP) addresses than was possible with the original system of IP address classes.

- CIDR: Classless InterDomain Routing
  - Subnet portion of address of arbitrary length
  - Address format: a.b.c.d/x, where x is # bits in subnet portion of address
  - All 1's in the host part is used for subnet broadcast



Homework 4C

- Consider a router that interconnects 3 subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix 223.1.17/24. Also suppose that Subnet 1 is required to support up to 63 interfaces, Subnet 2 is to support up to 96 interfaces, and Subnet 3 is to support up to 16 interfaces. Provide three network address prefixes (of the form a.b.c.d/x) that satisfy these constraints. **Use adjacent allocations**. For each subnet, also list the subnet mask to be used in the hosts.

Forwarding an IP Datagram

- Delivers **datagram**s to destination network (subnet)
- Routers maintain a "routing table" of "next hops"
- Next Hop field does not appear in the datagram

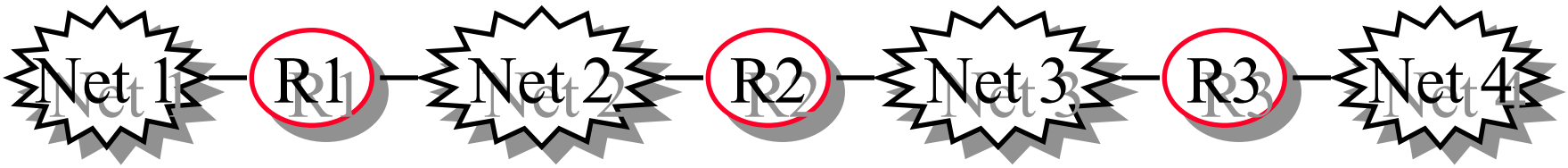


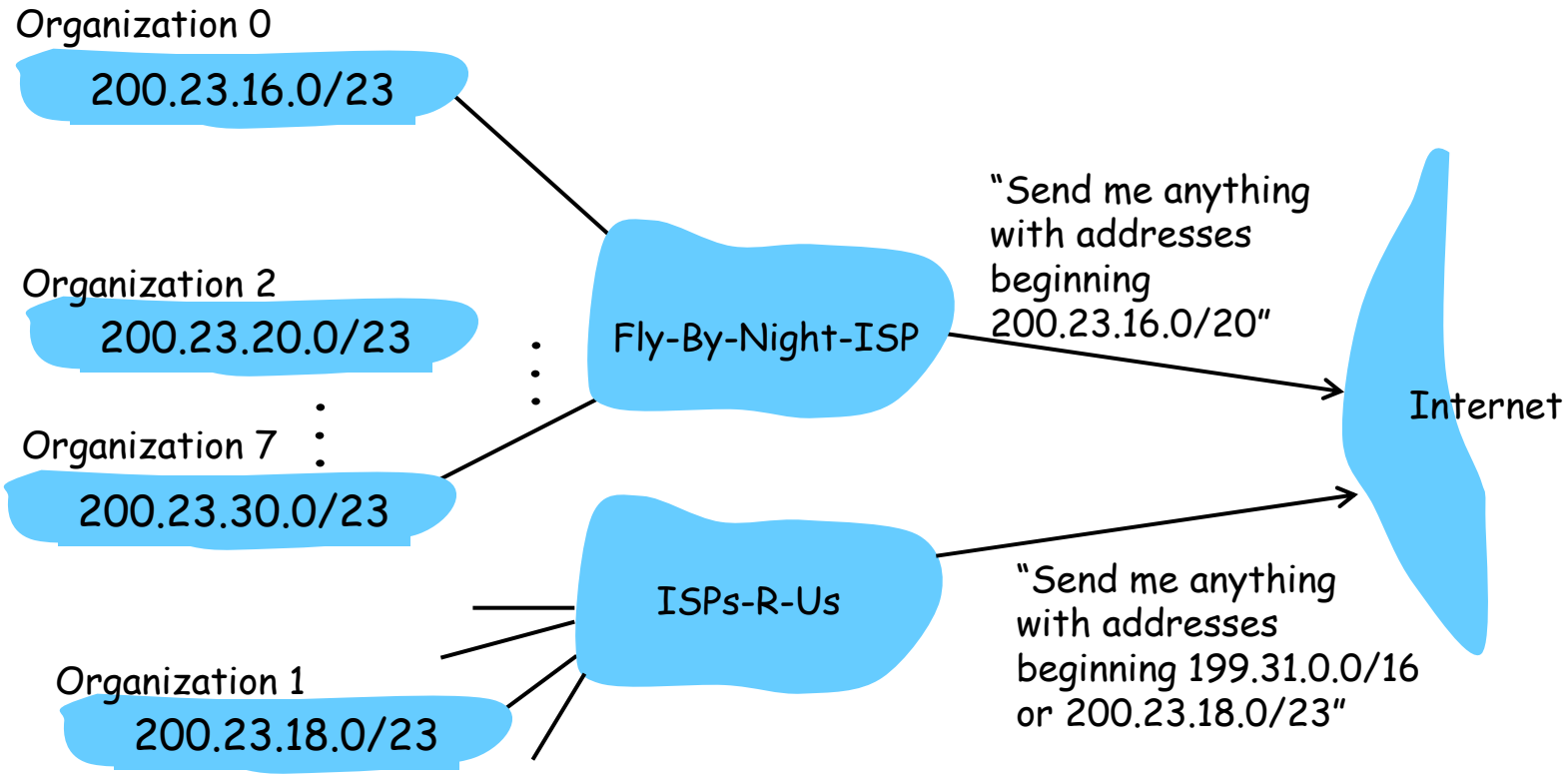
Table at R2:

Destination	Next Hop
Net 1	Forward to R1
Net 2	Deliver Direct
Net 3	Deliver Direct
Net 4	Forward to R3

Route Aggregation

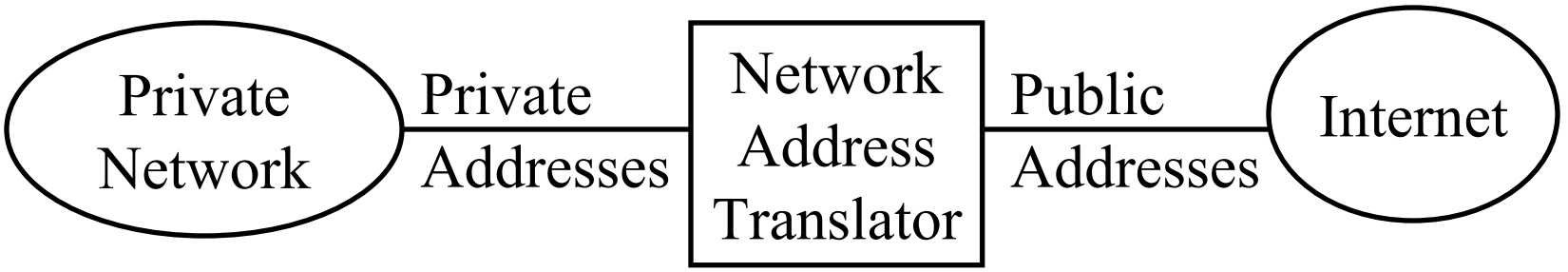
Route aggregation/summarization, is a method of minimizing the number of routing tables in an IP network. It works by consolidating selected multiple routes into a single route advertisement, in contrast to flat routing in which every routing table contains a unique entry for each route.

- Can combine two or more prefixes into a shorter prefix
- ISPs-R-Us has a more specific route to organization 1

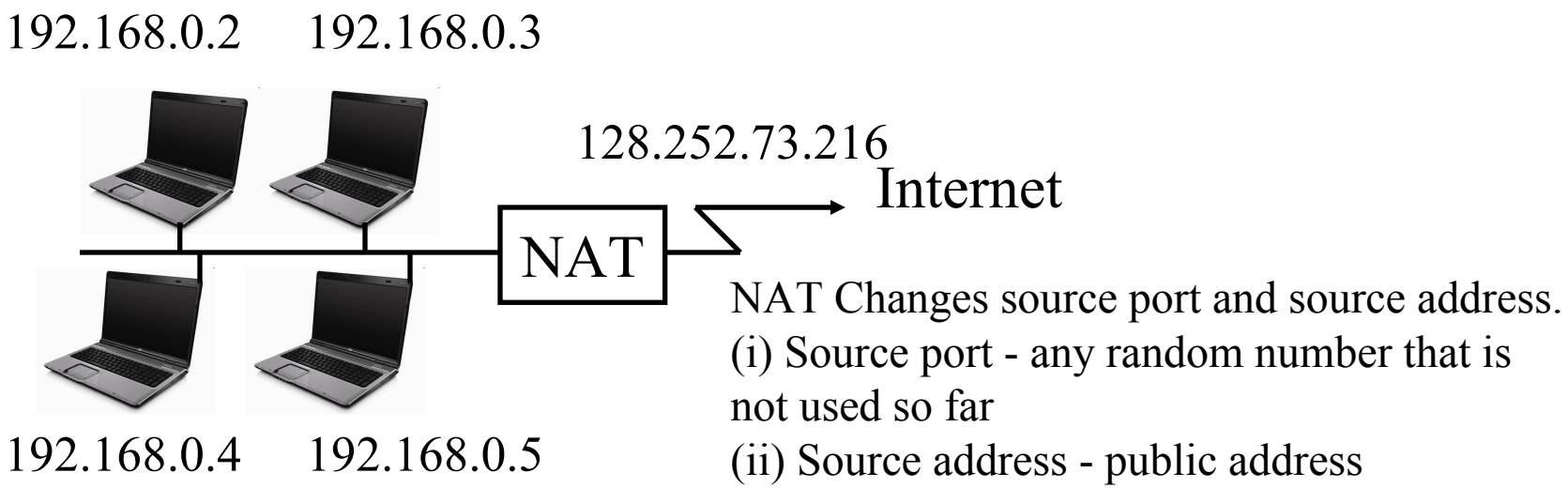


Private Addresses

- Any organization can use these inside their network Can't go on the internet. [RFC 1918]
- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)



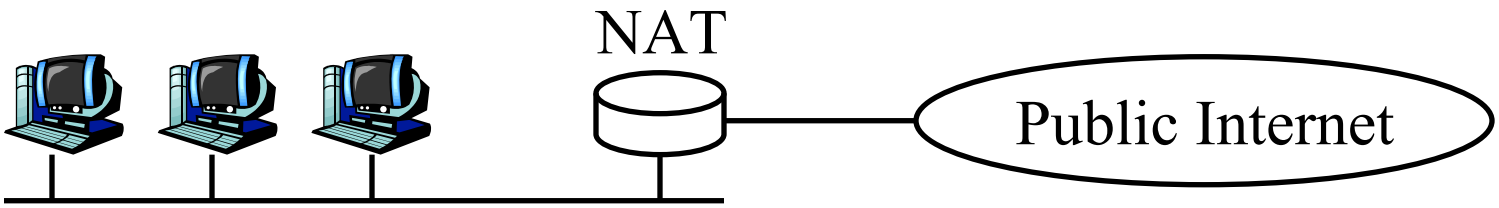
Network Address Translation (NAT)



- Private IP addresses 192.168.x.x
- Can be used by anyone inside their networks
- Cannot be used on the public Internet
- NAT overwrites source addresses on all outgoing packets and overwrites destination addresses on all incoming packets
- Only outgoing connections are possible

Universal Plug and Play

- NAT needs to be manually programmed to forward external requests
- UPnP allows hosts to request port forwarding
- Both hosts and NAT should be UPnP aware
- Host requests forwarding all port xx messages to it
- NAT returns the public address and the port #.
- Host can then announce the address and port # outside
- Outside hosts can then reach the internal host (server)

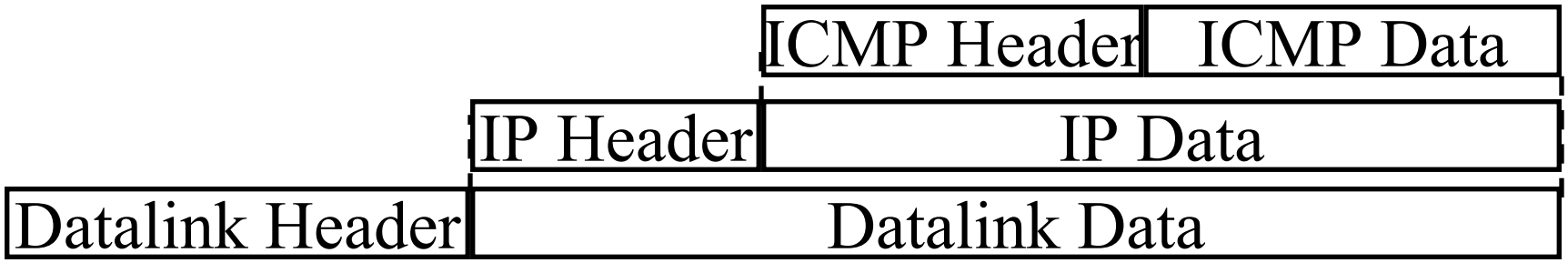


DHCP

- Dynamic Host Control Protocol
- Allows hosts to get an IP address automatically from a server
- Do not need to program each host manually
- Each allocation has a limited “lease” time
- Can reuse a limited number of addresses
- Hosts broadcast “Is there a DHCP Server Here?” Sent to 255.255.255.255
- DHCP servers respond

ICMP

- Internet Control Message Protocol
- Required companion to IP. Provides feedback from the network.
- ICMP: Used by IP to send error and control messages
- ICMP uses IP to send its messages (Not UDP)
- ICMP does not report errors on ICMP messages.
- ICMP reports error only on the first fragment



IPv6

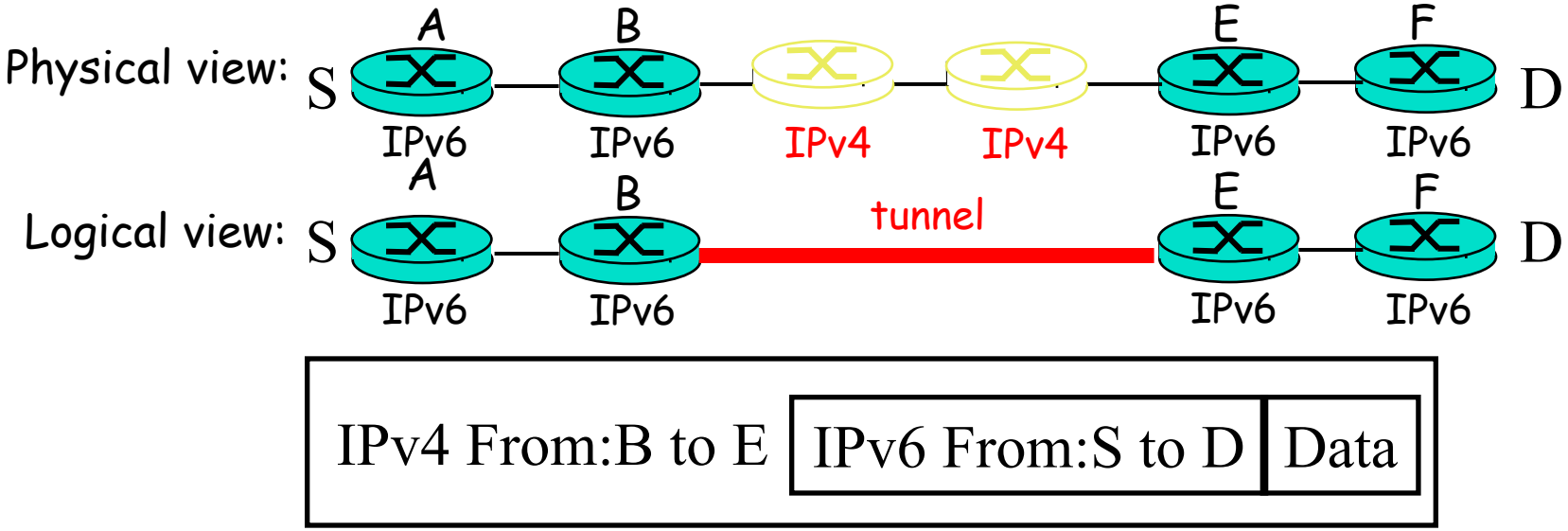
- Shortage of IPv4 addresses ⇒ Need larger addresses
- IPv6 was designed with 128-bit addresses
- $2^{128} = 3.4 \times 10^{38}$ addresses
  ⇒ 665×10^{21} addresses per sq. m of earth surface
- If assigned at the rate of $10^6/\mu s$, it would take 20 years
- **Dot-Decimal**: 127.23.45.88
- **Colon-Hex:** FEDC:0000:0000:0000:3243:0000:0000:ABCD
  - Can skip leading zeros of each word
  - Can skip one sequence of zero words, e.g.,
    FEDC::3243:0000:0000:ABCD
    ::3243:0000:0000:ABCD
  - Can leave the last 32 bits in dot-decimal, e.g., ::127.23.45.88
  - Can specify a prefix by /length, e.g., 2345:BA23:0007::/50

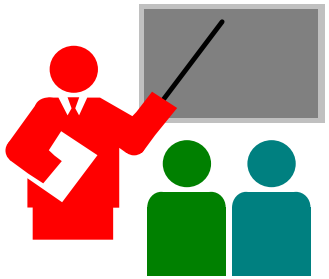
IPv6 vs. IPv4

- 1995 vs. 1975
- IPv6 only twice the size of IPv4 header
- Only version number has same position and meaning as in IPv4
- Removed: header length, type of service, identification, flags, fragment offset, header checksum $\Rightarrow$ No fragmentation
- Datagram length replaced by payload length
- Protocol type replaced by next header
- Time to live replaced by hop limit
- Added: Priority and flow label
- All fixed size fields.
- No optional fields. Replaced by extension headers.
- 8-bit hop limit = 255 hops max (Limits looping)
- Next Header = 6 (TCP), 17 (UDP)

IPv4 to IPv6 Transition

- **Dual Stack**: Each IPv6 router also implements IPv4
  IPv6 is used only if source host, destination host, and all routers on the path are IPv6 aware.
- **Tunneling**:The last IPv6 router puts the entire IPv6 datagram in a new IPv4 datagram addressed to the next IPv6 router
  = **Encapsulation**





Forwarding Protocols: Review

1. IPv4 uses 32 bit addresses consisting of subnet + host
2. Private addresses can be reused
   $\Rightarrow$ Helped solve the address shortage to a great extent 3.
   ICMP is the IP control protocol to convey IP error messages
4. DHCP is used to automatically allocate addresses to hosts 5.
   IPv6 uses 128 bit addresses. Requires dual stack or tunneling to coexist with IPv4.

Routing Algorithms

1. Graph abstraction
2. Distance Vector vs. Link State
3. Dijkstra's Algorithm
4. Bellman-Ford Algorithm

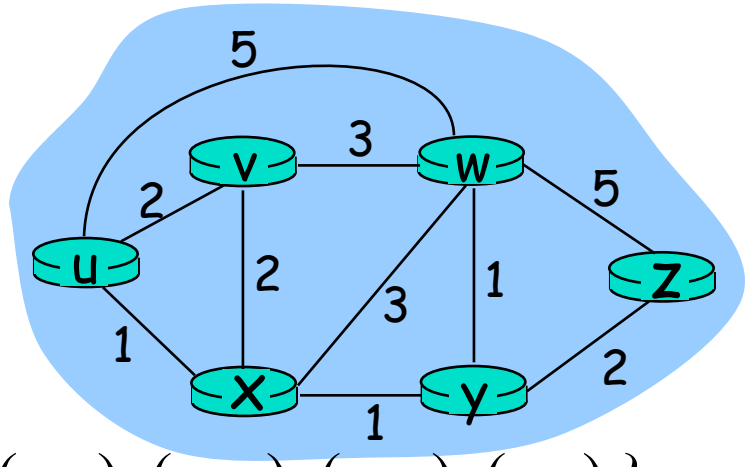
Rooting or Routing

- *Rooting* is what fans do at football games, what pigs do for truffles under oak trees in the Vaucluse, and what nursery workers intent on propagation do to cuttings from plants.
- *Routing* is how one creates a beveled edge on a table top or sends a corps of infantrymen into full scale, disorganized retreat

Routeing or Routing

- Routeing: British
- Routing: American
- Since Oxford English Dictionary is much heavier than any other dictionary of American English, British English generally prevails in the documents produced by ISO and CCITT; wherefore, most of the international standards for routing standards use the routeing spelling.

Graph abstraction

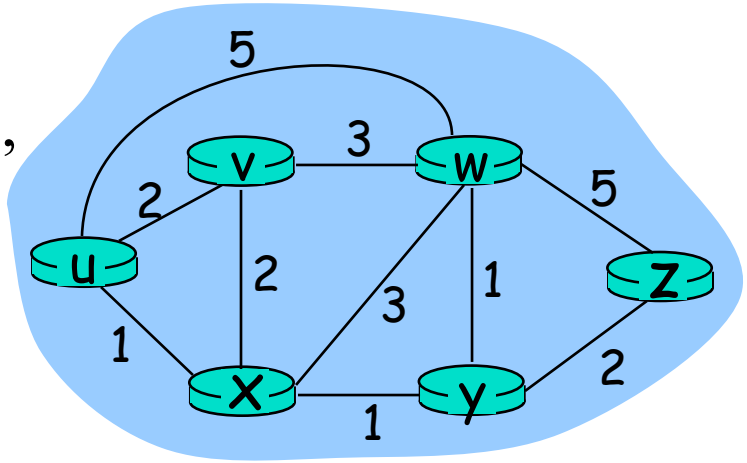


- Graph: G = (N,E)
- N = Set of routers
  = { u, v, w, x, y, z }
- E = Set of links
  ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }
- Each link has a cost, e.g., c(w,z) = 5
- Cost of path $(x_1, x_2, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$
- Routing Algorithms find the least cost path
- We limit to "Undirected" graphs, i.e., cost is same in both directions

Distance Vector vs Link State

Distance Vector:

- Vector of distances to all nodes, e.g.,
  u: {u:0, v:2, w:5, x:1, y:2, z:4}
- Sent to neighbors, e.g.,
  u will send to v, w, x
- Large vectors to small # of nodes
  Tell about the world to neighbors
- Older method. Used in RIP.



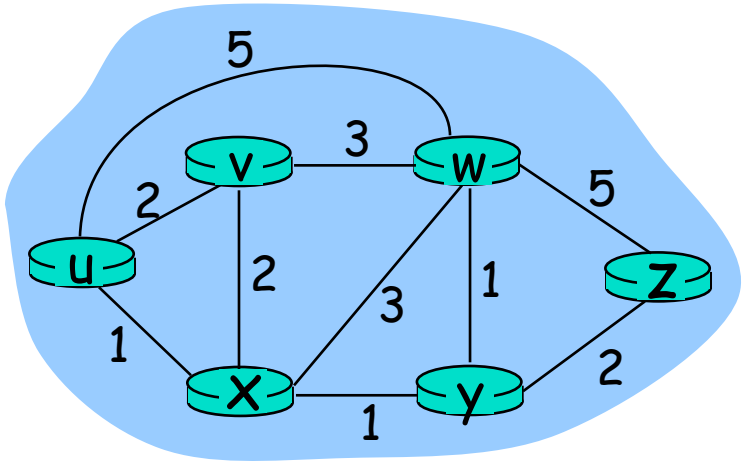
Link State:

- Vector of link cost to neighbors, e.g, u: {v:2, w:5, x:1}
- Sent to all nodes, e.g., u will send to v, w, x, y, z
- Small vectors to large # of nodes
  Tell about the neighbors to the world
- Newer method. Used in OSPF.

Dijkstra's Algorithm

- Goal: Find the least cost paths from a given node to all other nodes in the network
- Notation:
  c(i,j) = Link cost from i to j if i and j are connected
  D(k) = Total path cost from s to k
  N' = Set of nodes so far for which the least cost path is known
- Method:
  - Initialize: N'={u}, D(v) = c(u,v) for all neighbors of u
  - Repeat until N includes all nodes:
    - Find node $w \notin$ N', whose D(w) is minimum
    - Add w to N'
    - Update D(v) for each neighbor of w that is not in N'
      D(v) = min[D(v), D(w) + c(w,v)] for all $v \notin$ N'

Dijkstra's Algorithm: Example

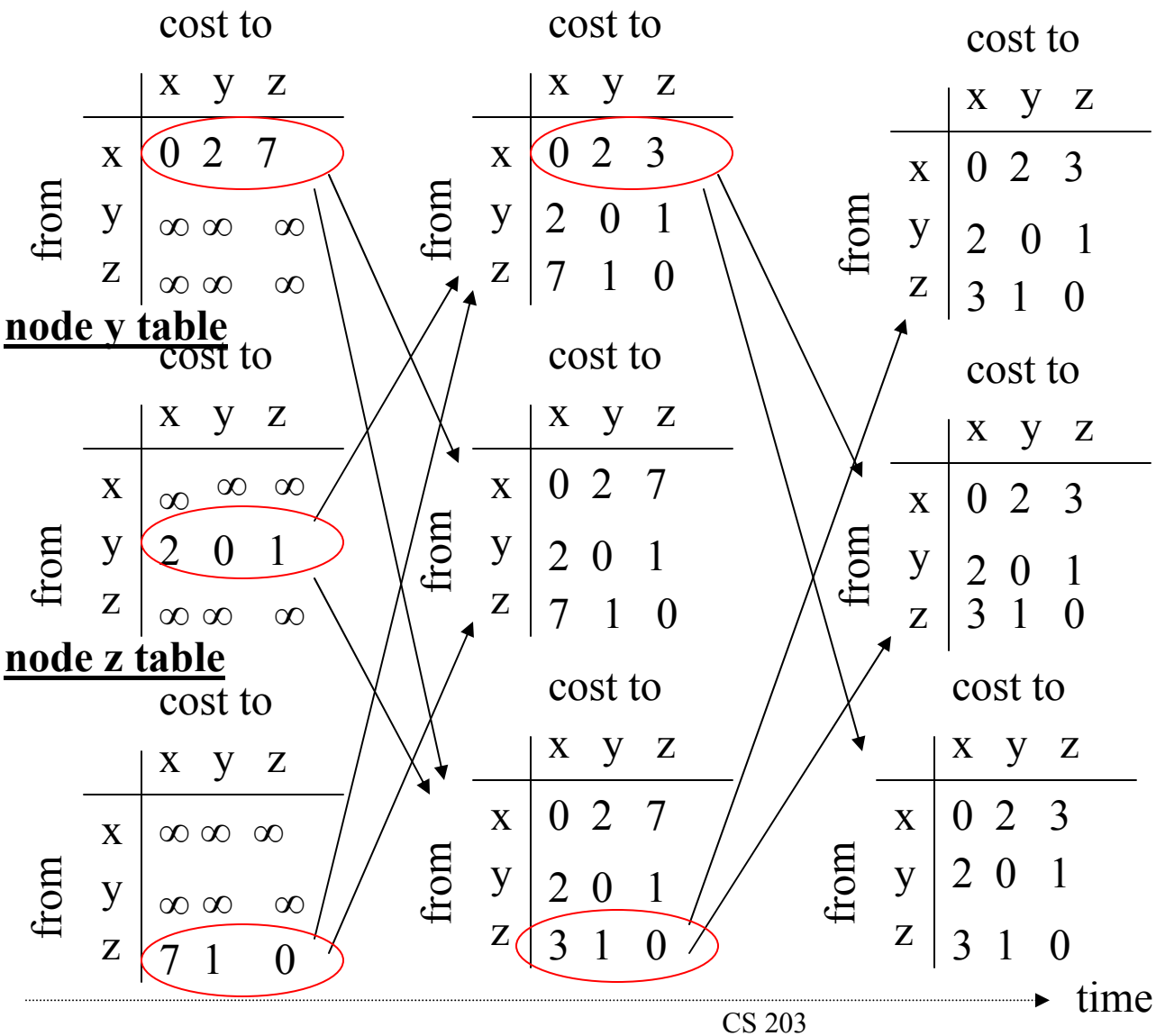


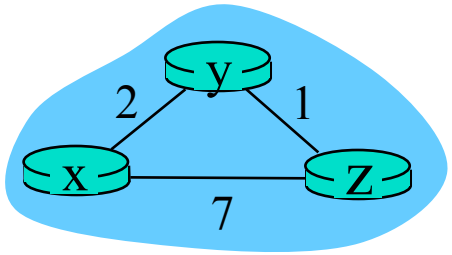
	N'	D(v)	Path	D(w)	Path	D(x)	Path	D(y)	Path	D(z)	Path
0	{u}	2	u-v	5	u-w	1	u-x	∞	-	∞	-
1	{u, x}	2	u-v	4	u-x-w			2	u-x-y	∞	-
2	{u, x, y}	2	u-v	3	u-x-y-w					4	u-x-y-z
3	{u, x, y, v}			3	u-x-y-w					4	u-x-y-z
4	{u, x, y, v, w}									4	u-x-y-z
5	{u, x, y, v, w, z}										

Bellman-Ford Algorithm

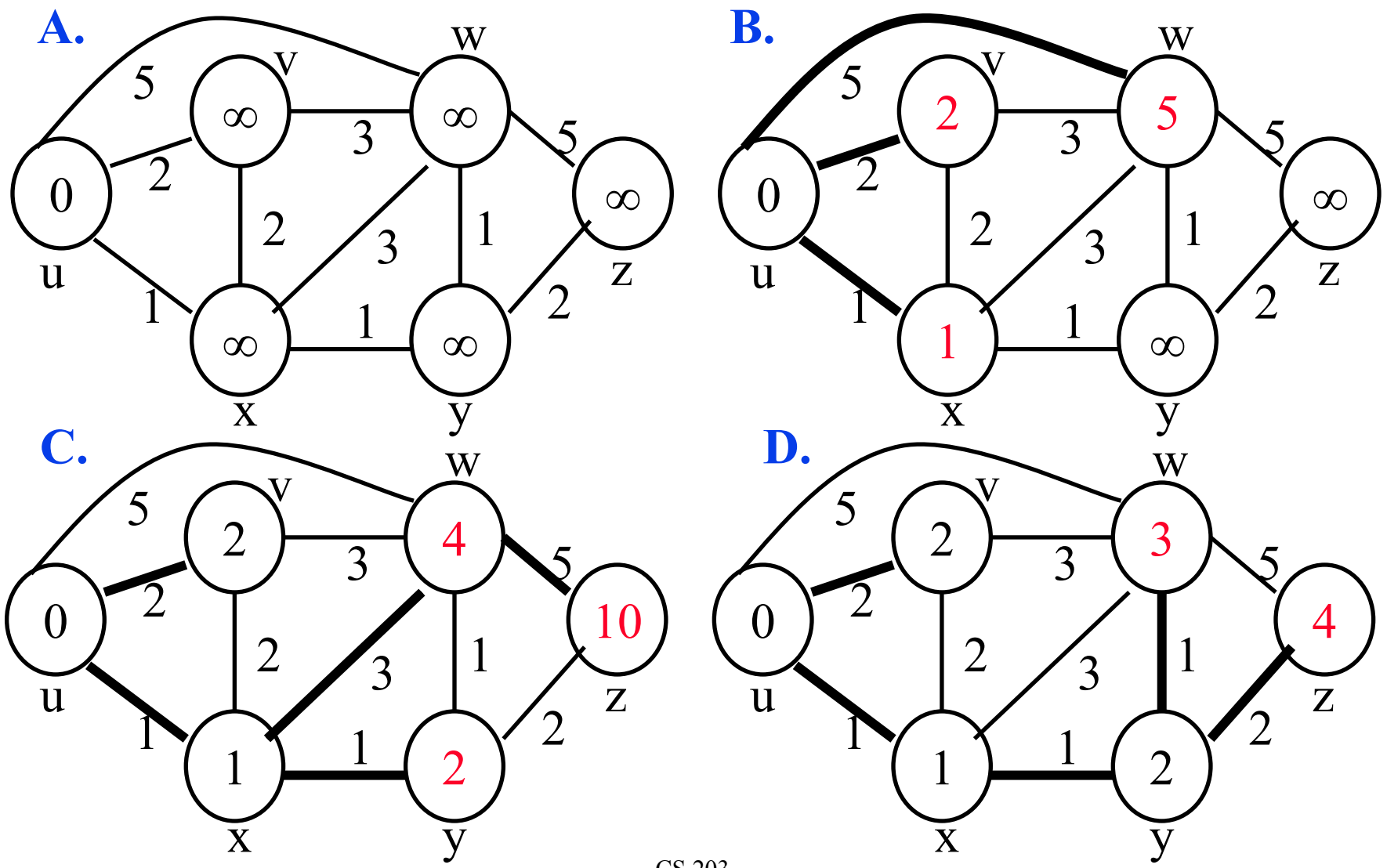
- Notation:
  u = Source node
  c(i,j) = link cost from i to j
  h = Number of hops being considered
  $D_u(n)$ = Cost of h-hop path from u to n
- Method:
  1. Initialize: $D_u(n) = \infty$ for all $n \neq u$; $D_u(u) = 0$
  2. For each node: $D_u(n) = \min_j [D_u(j) + c(j,n)]$
  3. If any costs change, repeat step 2

Bellman Ford Example 1

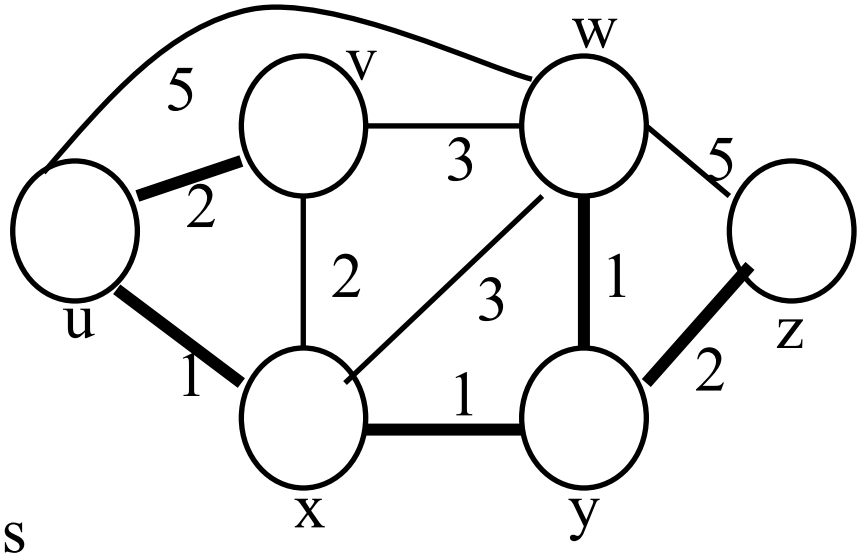




Bellman-Ford Example 2



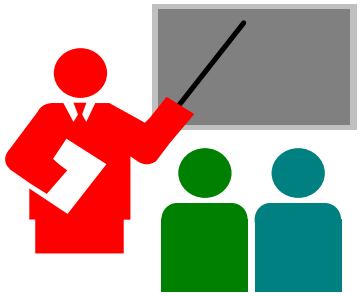
Bellman-Ford: Tabular Method



If cost changes

⇒ Recompute the costs to all neighbors

h	D(v)	Path	D(w)	Path	D(x)	Path	D(y)	Path	D(z)	Path
0	∞	-	∞	-	∞	-	∞	-	∞	-
1	2	u-v	5	u-w	1	u-x	∞	-	∞	-
2	2	u-v	4	u-x-w	1	u-x	2	u-x-y	10	u-w-z
3	2	u-v	3	u-x-y-w	1	u-x	2	u-x-y	4	u-x-y-z
4	2	u-v	3	u-x-y-w	1	u-x	2	u-x-y	4	u-x-y-z

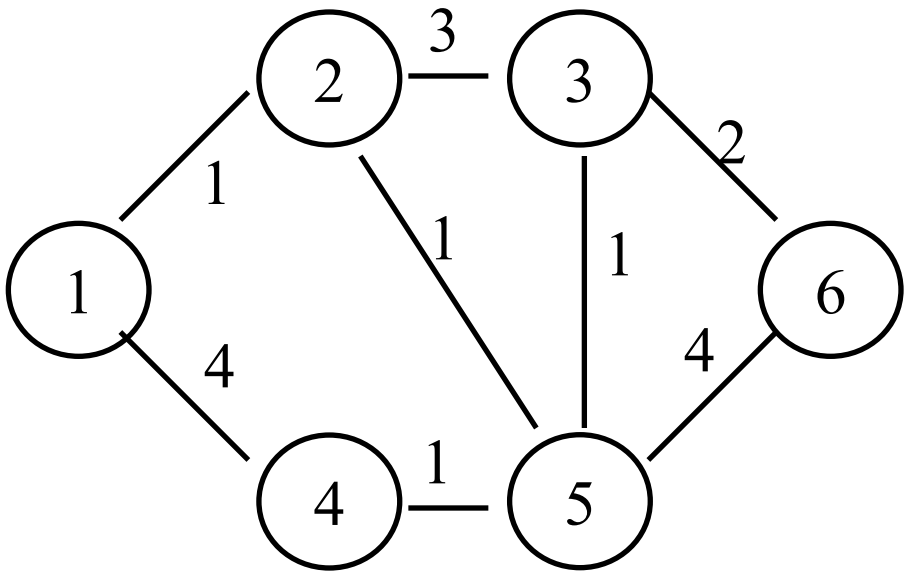


Routing Algorithms: Summary

1. Distance Vectors: Distance to all nodes in the network sent to neighbors
2. Link State: Cost of link to neighbors sent to entire network
3. Dijkstra's algorithm is used to compute shortest path using link state
4. Bellman Ford's algorithm is used to compute shortest paths using distance vectors

Homework 4E

Prepare the routing calculation ***table*** for node 1 in the following network using (a) Dijkstra's algorithm (b) Bellman Ford Algorithm.



Routing Protocols

1. Autonomous Systems (AS)
2. Routing Information Protocol (RIP)
   - Counting to Infinity Problem
3. Open Shortest Path First (OSPF)
   - OSPF Areas
4. Border Gateway Protocol (BGP)

Autonomous Systems

- An internet connected by homogeneous routers under the administrative control of a single entity

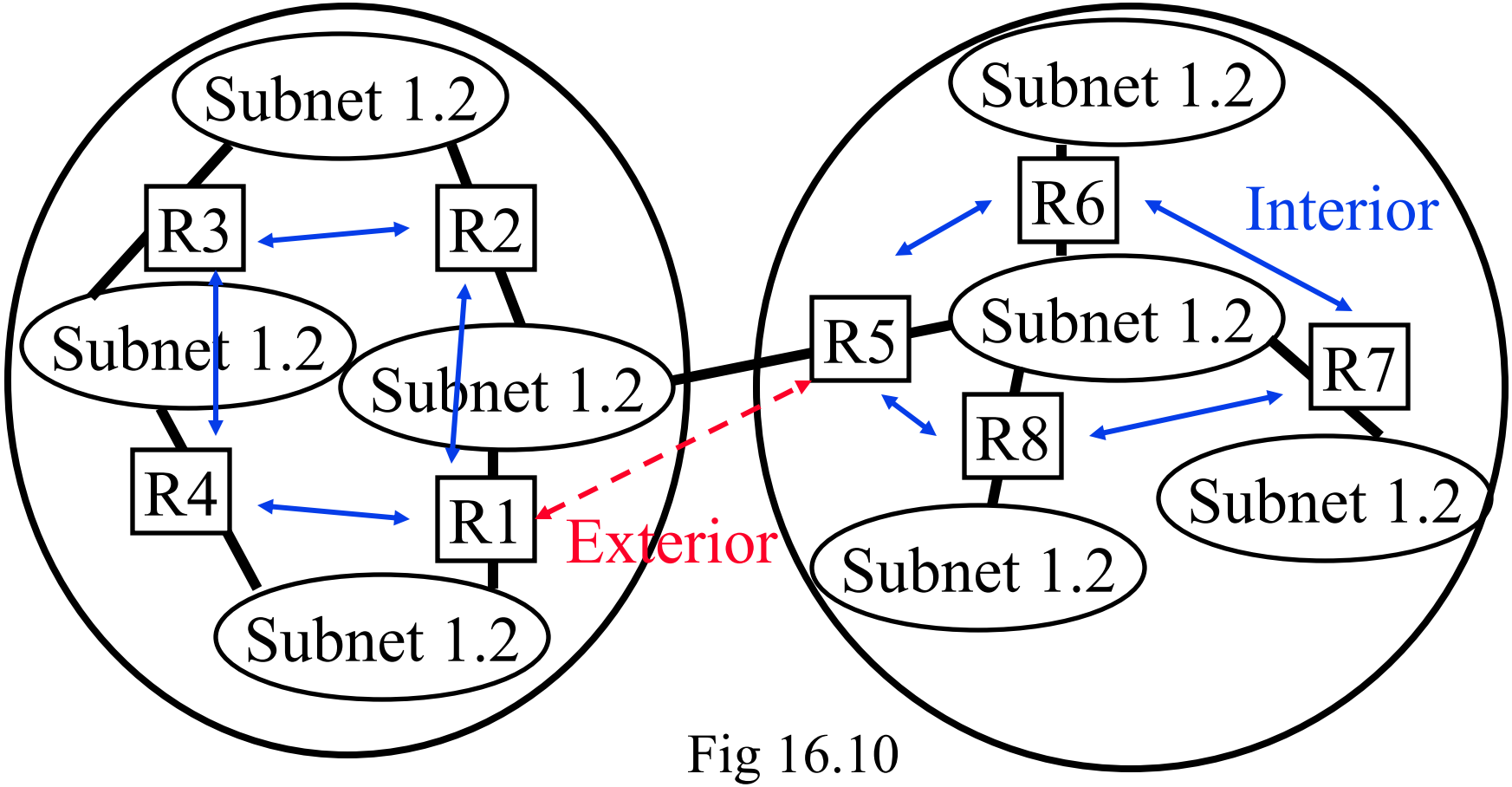


Fig 16.10

Routing Protocols

- Interior Router Protocol (IRP): Used for passing routing information among routers internal to an autonomous system. Also known as IGP.
  - Examples: RIP, OSPF
- Exterior Router Protocol (ERP): Used for passing routing information among routers between autonomous systems. Also known as EGP.
  - Examples: EGP, BGP, IDRP

    Note: EGP is a class as well as an instance in that class.

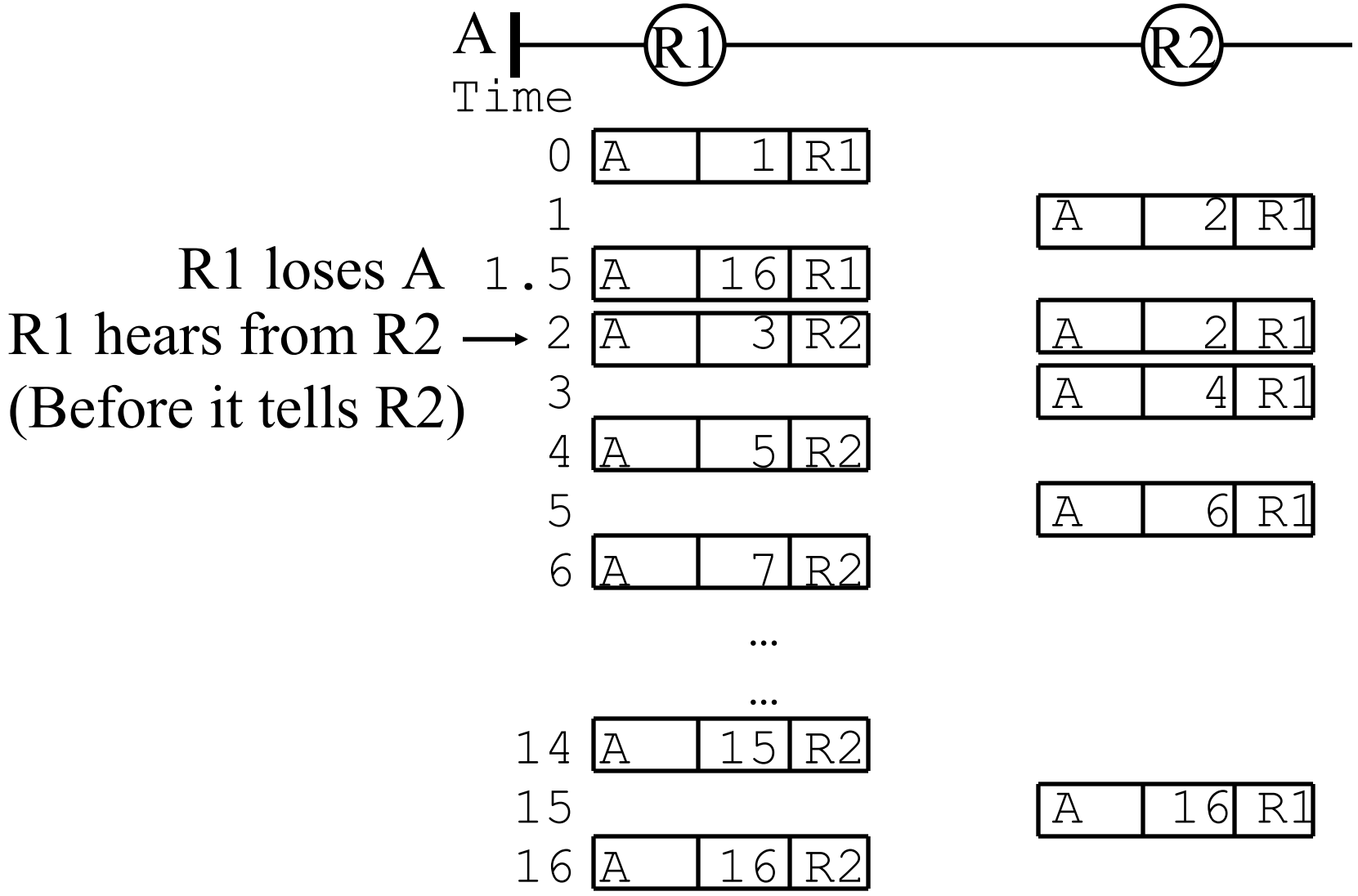
Routing Information Protocol

- RIP uses distance vector $\Rightarrow$ A vector of distances to all nodes is sent to neighbors
- Each router computes new distances:
  - Replace entries with new lower hop counts
  - Insert new entries
  - Replace entries that have the same next hop but higher cost
  - Each entry is aged.
    Remove entries that have aged out
- Send out updates every 30 seconds.

Shortcomings of RIP

- Maximum network diameter = 15 hops
- Cost is measured in hops
  Only shortest routes. May not be the fastest route.
- Entire tables are broadcast every 30 seconds.
  Bandwidth intensive.
- Uses UDP with 576-byte datagrams.
  Need multiple datagrams.
  300-entry table needs 12 datagrams.
- An error in one routing table is propagated to all routers
- Slow convergence

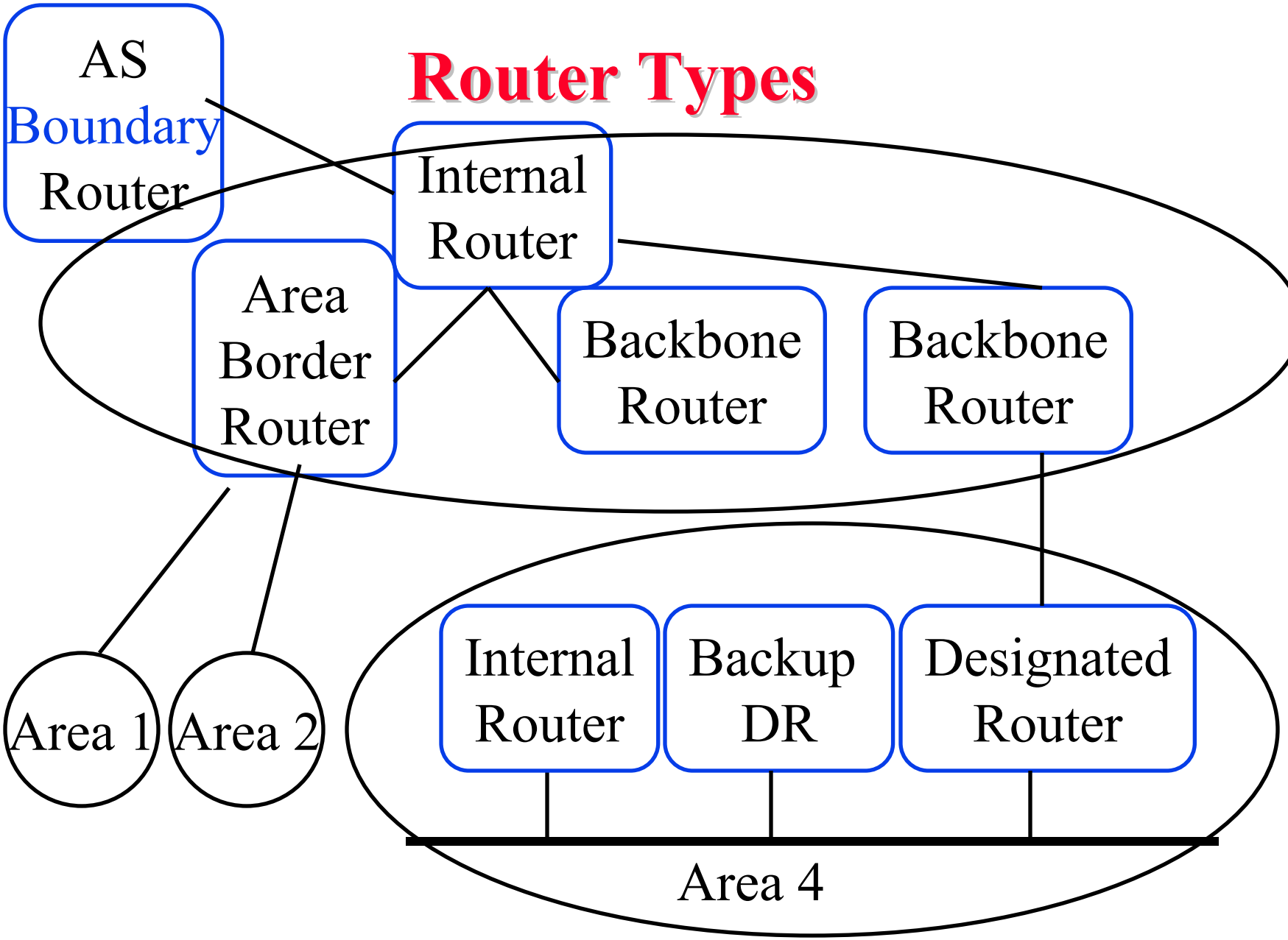
Counting to Infinity Problem



Open Shortest Path First (OSPF)

- Uses true metrics (not just hop count)
- Uses subnet masks
- Allows load balancing across equal-cost paths
- Supports type of service (ToS)
- Allows external routes (routes learnt from other autonomous systems)
- Authenticates route exchanges
- Quick convergence
- Direct support for multicast
- Link state routing $\Rightarrow$ Each router broadcasts its connectivity with neighbors to entire network

Router Types

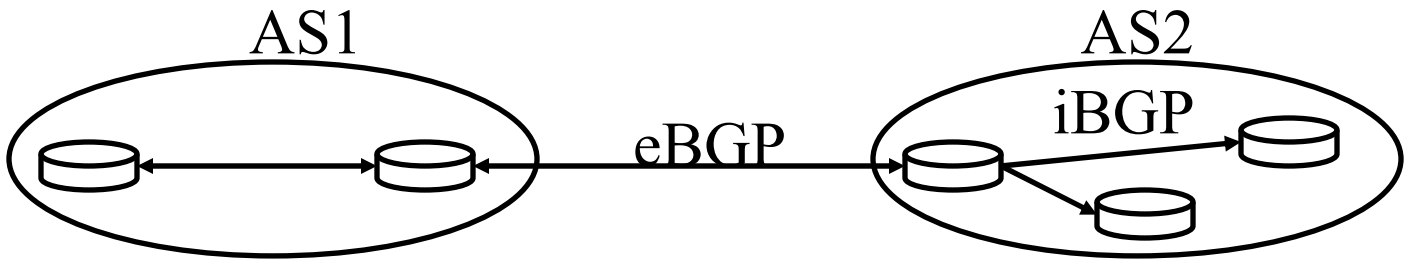


Router Types (Cont)

- **Internal Router (IR)**: All interfaces belong to the same area
- **Area Border Router (ABR):** Interfaces to multiple areas
- **Backbone Router (BR):** Interfaces to the backbone
- **Autonomous System Boundary Router (ASBR):** Exchanges routing info with other autonomous systems
- **Designated Router (DR):** Generates link-state info about the subnet
- **Backup Designated Router (BDR):** Becomes DR if DR fails.

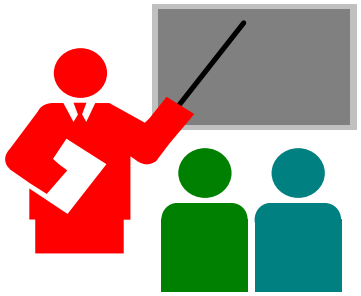
Border Gateway Protocol

- Inter-autonomous system protocol [RFC 1267]
- Used since 1989 but not extensively until recently
- Runs on TCP (segmentation, reliable transmission)
- Advertises all transit ASs on the path to a destination address
- A router may receive multiple paths to a destination $\Rightarrow$ Can choose the best path
- iBGP used to forward paths inside the AS.
  eBGP used to exchange paths between ASs.



Intra- vs. Inter-AS Routing

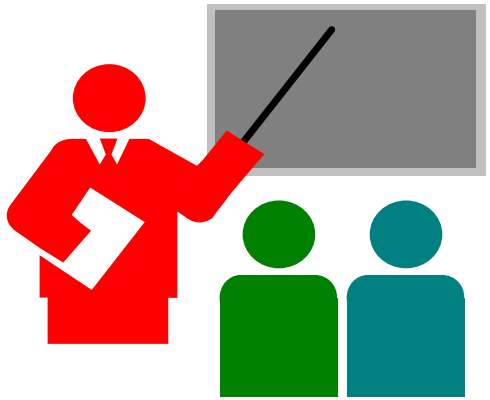
- **Policy:**
  - Inter-AS: admin wants control over how its traffic routed, who routes through its net.
  - Intra-AS: single admin, so no policy decisions needed
- **Scale**:
  - Hierarchical routing saves table size, reduced update traffic
- **Performance**:
  - Intra-AS: can focus on performance
  - Inter-AS: policy may dominate over performance



Routing Protocols: Summary

1. RIP uses distance-vector routing
2. RIP v2 fixes the slow convergence problem
3. OSPF uses link-state routing and divides the autonomous systems into multiple areas.
   Area border router, AS boundary router, designated router
4. BGP is an inter-AS protocol $\Rightarrow$ Policy driven

Network Layer: Summary



1. IP is a forwarding protocol. IPv6 uses 128 bit addressing.
2. Dijkstra's algorithm allows path computation using link state
3. Bellman Ford's algorithm allows path computation using distance vectors.
4. RIP is a distance vector IGP while OSPF is a link state IGP.
5. BGP is an EGP and uses path vectors