

The Network Layer

Raj Jain

Washington University in Saint Louis

Saint Louis, MO 63130

Jain@wustl.edu

Audio/Video recordings of this lecture are available on-line at:

<http://www.cse.wustl.edu/~jain/cse473-11/>



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

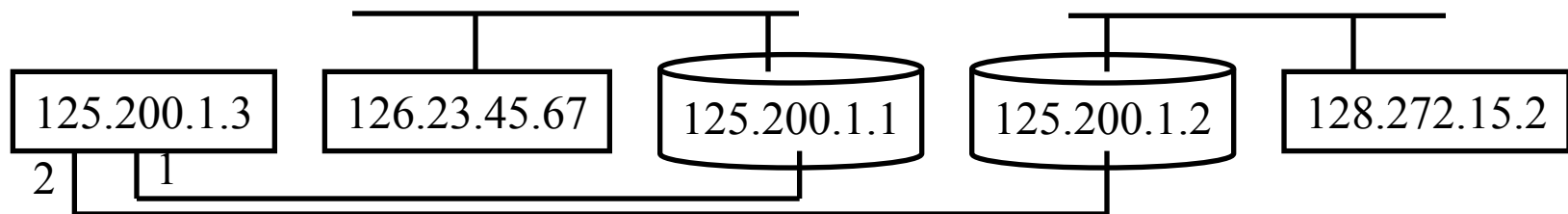
Network Layer Protocols

HTTP	FTP	SMTP	P2P	DHCP	RIP	OSPF	BGP
TCP						UDP	ICMP
IPv4						IPv6	
Ethernet	Point-to-Point				Wi-Fi		
Coax	Fiber		Wireless				

- ❑ 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

Optional Homework: R3

“Route Print” Command in Windows

MAC: netstat -rn

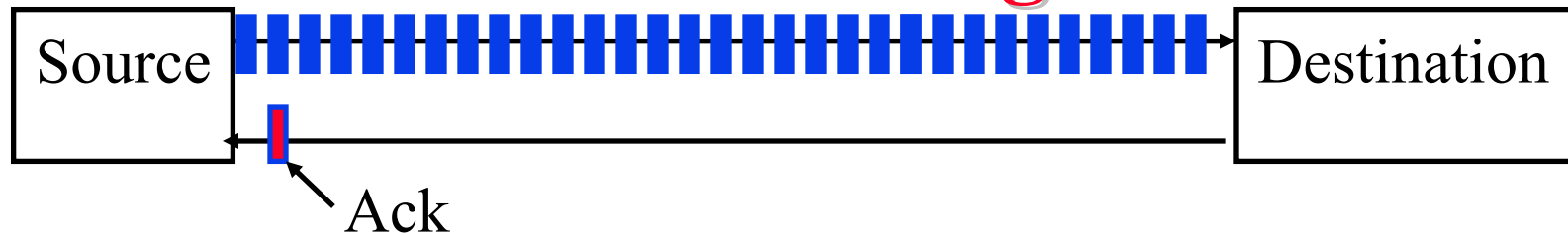
```
=====
Interface List
0x1 ..... MS TCP Loopback interface
0x2 ...00 16 eb 05 af c0 ..... Intel(R) WiFi Link 5350 - Packet Scheduler Miniport
0x3 ...00 1f 16 15 7c 41 ..... Intel(R) 82567LM Gigabit Network Connection - Packet Scheduler Miniport
0x40005 ...00 05 9a 3c 78 00 ..... Cisco Systems VPN Adapter - Packet Scheduler Miniport
=====
Active Routes:
Network Destination        Netmask          Gateway           Interface         Metric
0.0.0.0                    0.0.0.0          192.168.0.1       192.168.0.108     10
0.0.0.0                    0.0.0.0          192.168.0.1       192.168.0.106     10
127.0.0.0                  255.0.0.0        127.0.0.1         127.0.0.1         1
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

Homework 4A

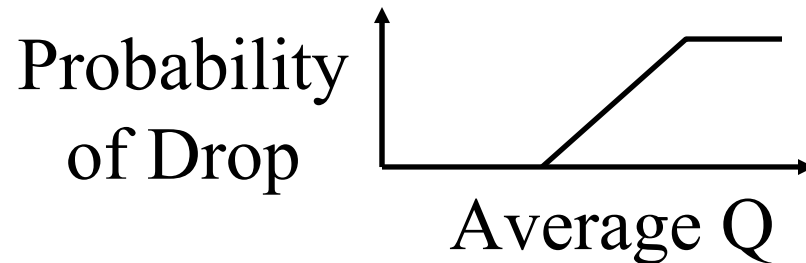
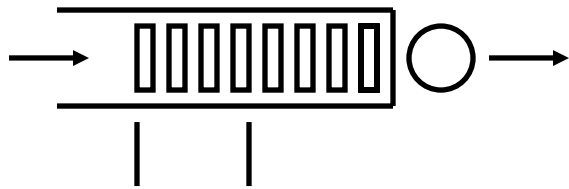
- ❑ Use “Route Help” (or man route) to learn the route command
- ❑ Ping www.google.com to find its address
- ❑ Make sure that you are connected via wired Ethernet to Internet and that wireless link is also available
- ❑ Disable wireless. Run ipconfig (ifconfig on Mac) to find your available links.
- ❑ Print route table
- ❑ Trace route to www.google.com using tracert
- ❑ Enable Wireless. Run ipconfig to ensure that both wired and wireless links are available.
- ❑ Print Route table.
- ❑ Modify routing table so that preferred path for www.google.com (just one address) is via wireless interface
- ❑ Print the new routing table
- ❑ Trace route to www.google.com and verify
- ❑ Note the command you used to modify the routing table and submit (one line only).

Ideal Buffering



- ❑ Flow Control Buffering = $RTT * \text{Transmission Rate}$
- ❑ Buffer = $RTT * \text{Transmission Rate} / \sqrt{(\# \text{ of TCP flows})}$

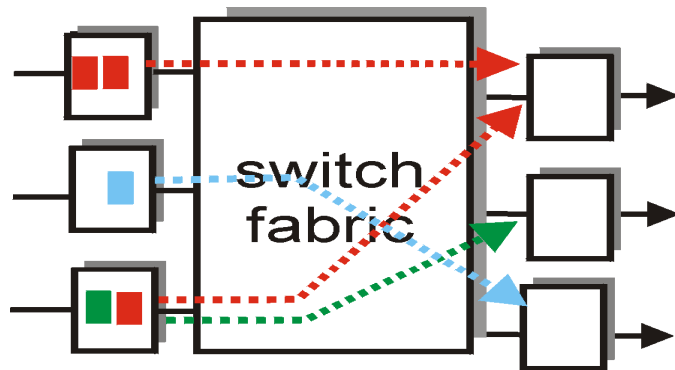
Packet Dropping Policies



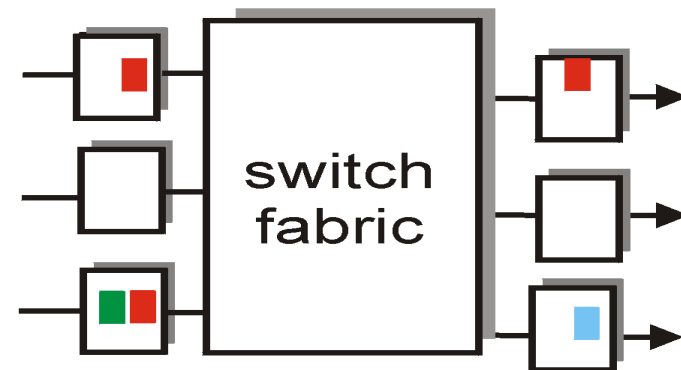
- ❑ **Drop-Tail:** Drop the arriving packet
 - ❑ **Random Early Drop (RED):** Drop arriving packets even before the queue is full
 - ❑ Routers measure average queue and drop incoming packet with certain probability
- ⇒ **Active Queue Management (AQM)**

Head-of-Line Blocking

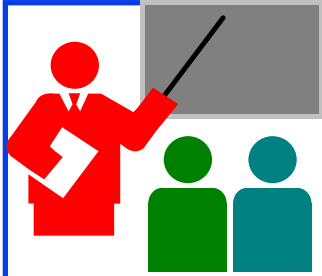
- Packet at the head of the queue is waiting
⇒ Other packets can not be forwarded even if they are going to other destination



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. ~~ATM provides a connection-oriented network layer and provides many services including throughput, delay, and jitter guarantees.~~
3. IP provides only best effort service (KISS).
4. Routers consist of input/output ports, switching fabric, and processors.
5. Datagrams may be dropped even if the queues are not full (Random early drop).
6. Queueing at input may result in head of line blocking.

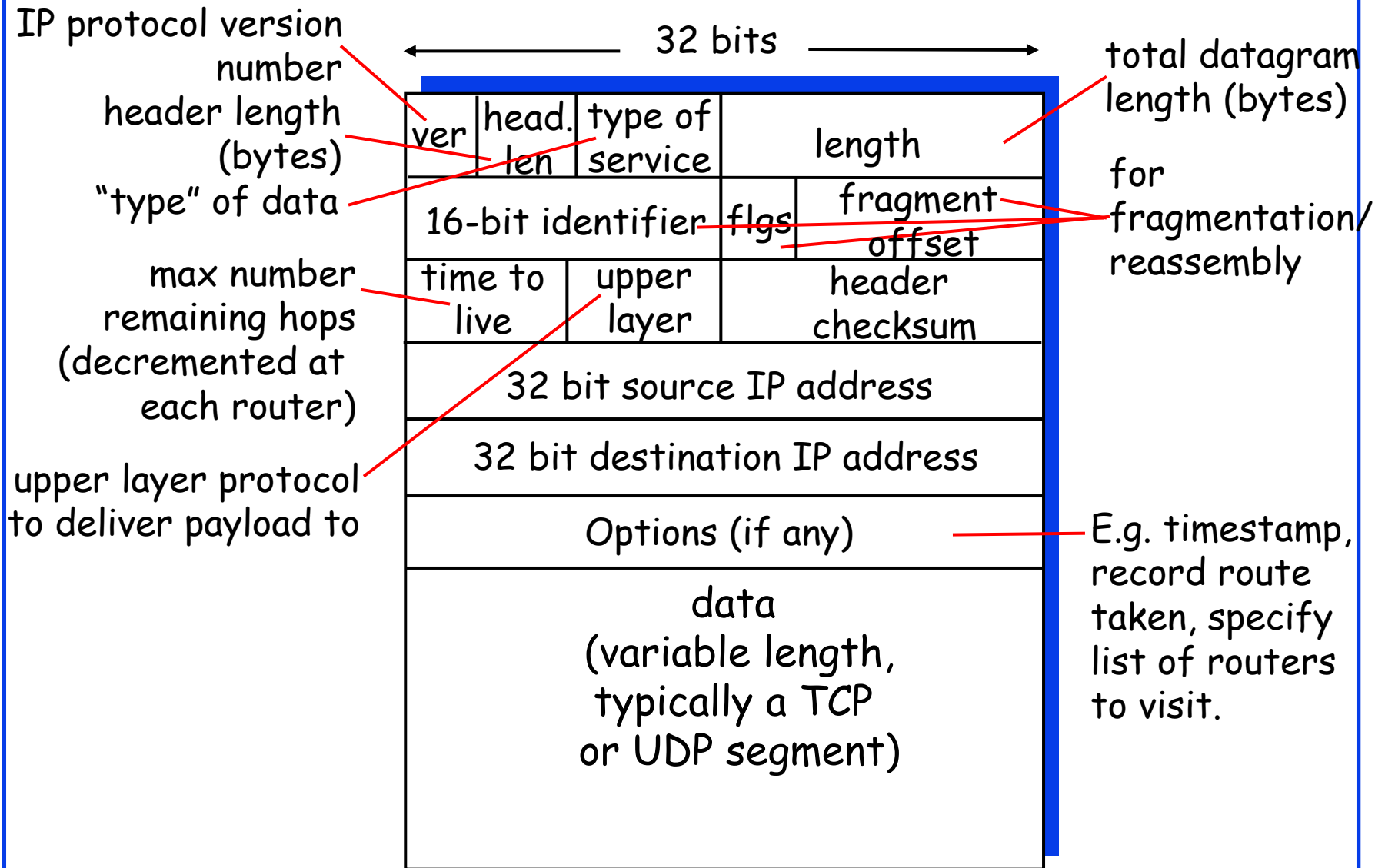
Ref: Read Sections 4.1, 4.1.1, ~~4.1.2~~, 4.2, ~~4.2.1~~, 4.2.2, 4.2.3, 4.3 full, Page 315-341 of the textbook. Try R1, R2, R3, ~~R4, R5, R6~~, R7, R8, R9, R10, R11



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

Example

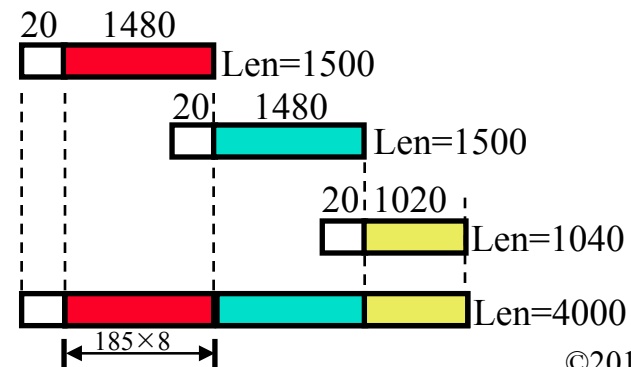
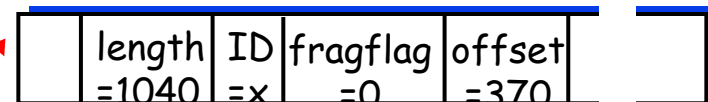
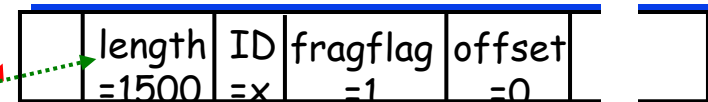
- ❑ 4000 byte datagram
- ❑ MTU = 1500 bytes



One large datagram becomes several smaller datagrams

1480 bytes in data field

offset = $1480/8$



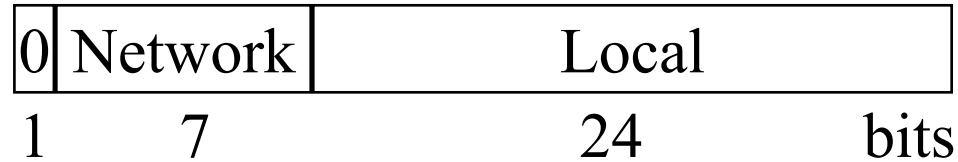
Ref: Section 4.4.1. Try R12, R13

Homework 4B

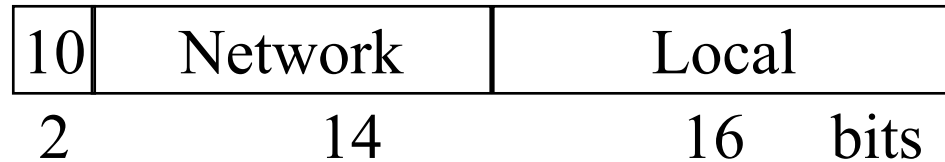
- Consider sending a 2400-byte datagram into a link that has an MTU of 720 bytes. Suppose the original datagram is stamped with the identification number 422. How many fragments are generated? What are the values in the various fields in the IP datagram(s) generated related to fragmentation?

IP Address Classes

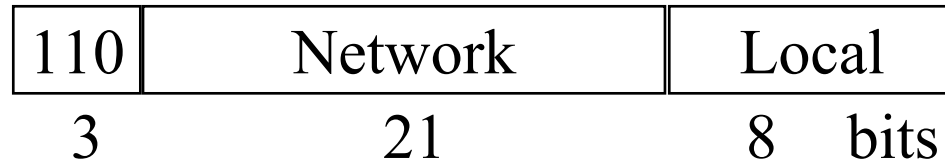
- Class A:



- Class B:



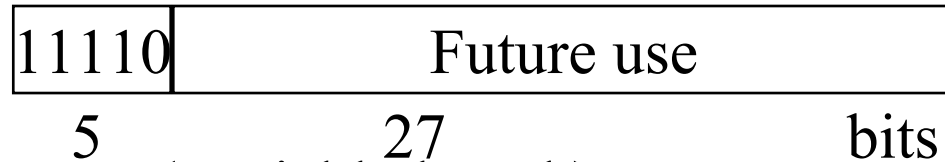
- Class C:



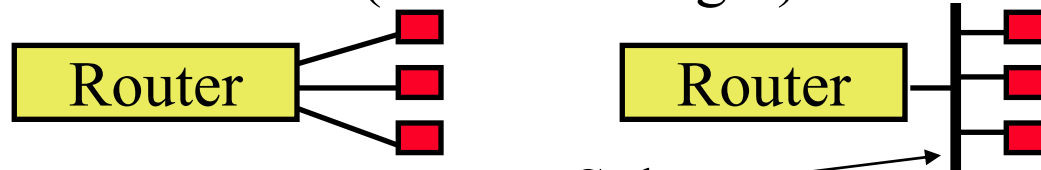
- Class D:



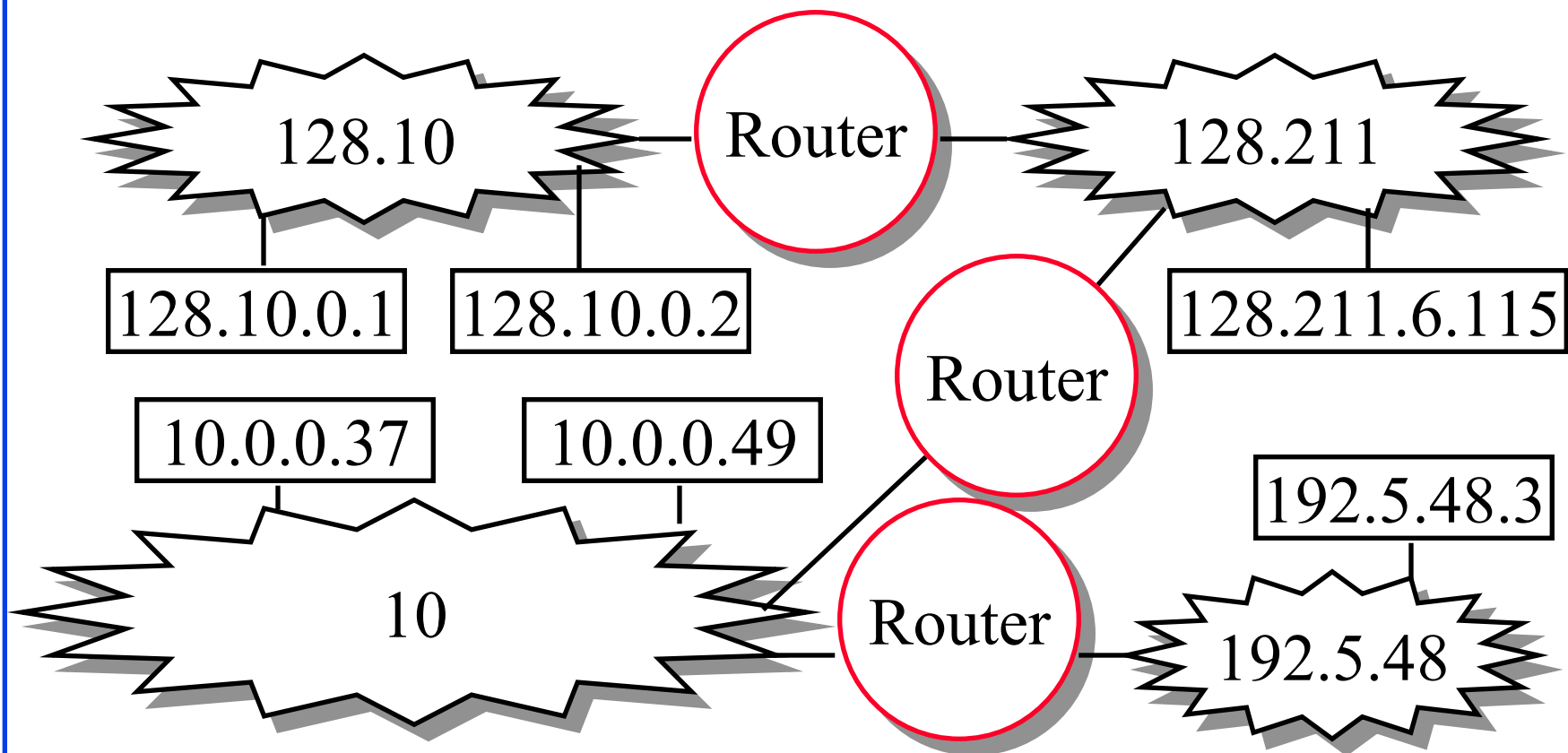
- Class E:



- Local = Subnet + Host (Variable length)

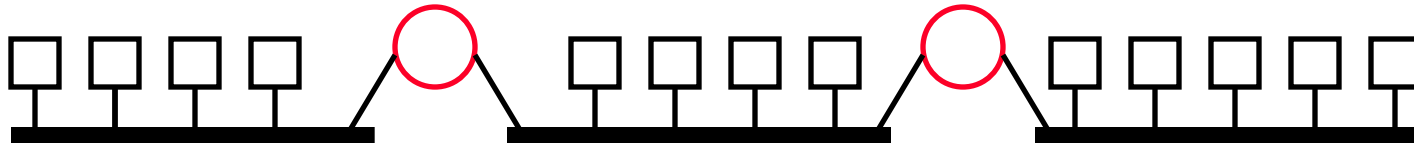


IP Addressing

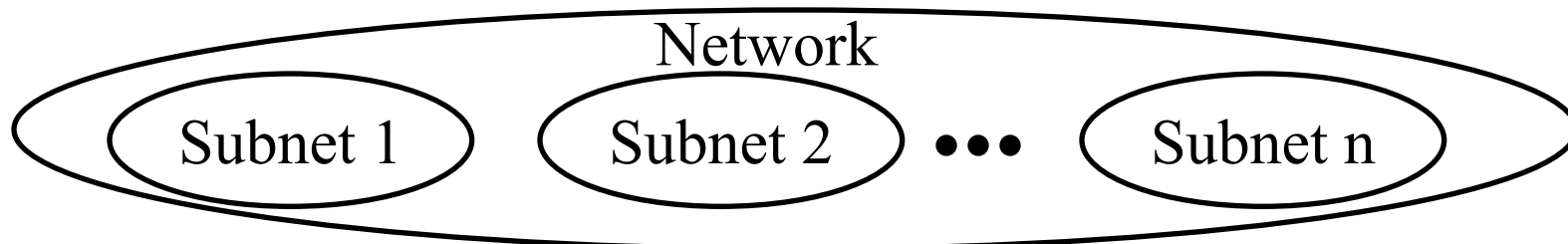


- ❑ All IP hosts have a 32-bit address. 128.10.0.1
= 1000 0000 0000 1010 0000 0000 0000 0001
- ❑ All hosts on a network have the same network prefix

Subnetting

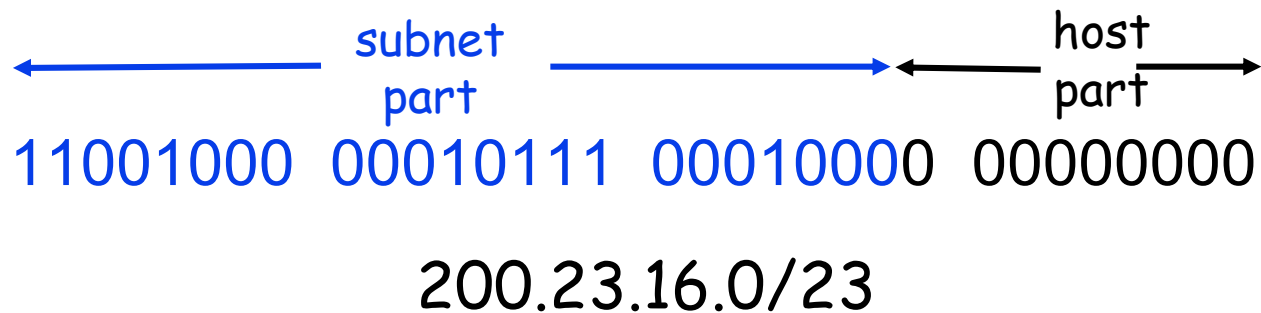


- All hosts on a subnetwork have the same prefix.
Position of the prefix is indicated by a “subnet mask”
- Example: First 23 bits = subnet
Address: 10010100 10101000 00010000 11110001
Mask: 11111111 11111111 11111110 00000000
.AND. 10010100 10101000 00010000 00000000



IP addressing: CIDR

- CIDR: Classless 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 **datagrams** to destination network (subnet)
- ❑ Routers maintain a “routing table” of “next hops”
- ❑ Next Hop field does not appear in the datagram

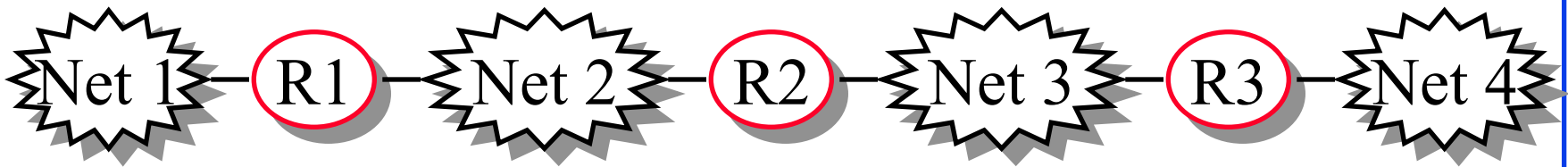


Table at R2:

Destination Next Hop

Net 1	Forward to R1
Net 2	Deliver Direct
Net 3	Deliver Direct
Net 4	Forward to R3

Route Aggregation

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

Organization 0

200.23.16.0/23

Organization 2

200.23.20.0/23

Organization 7

200.23.30.0/23

Organization 1

200.23.18.0/23

Fly-By-Night-ISP

ISPs-R-Us

"Send me anything
with addresses
beginning
200.23.16.0/20"

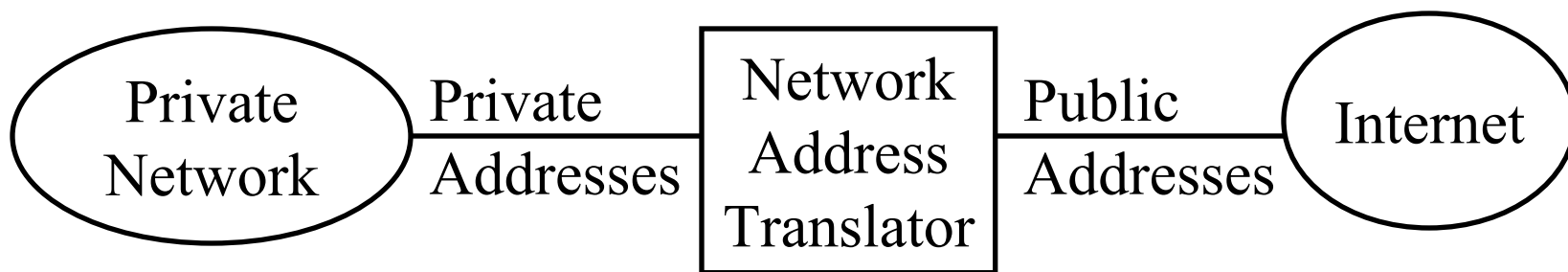
"Send me anything
with addresses
beginning 199.31.0.0/16
or 200.23.18.0/23"

Internet

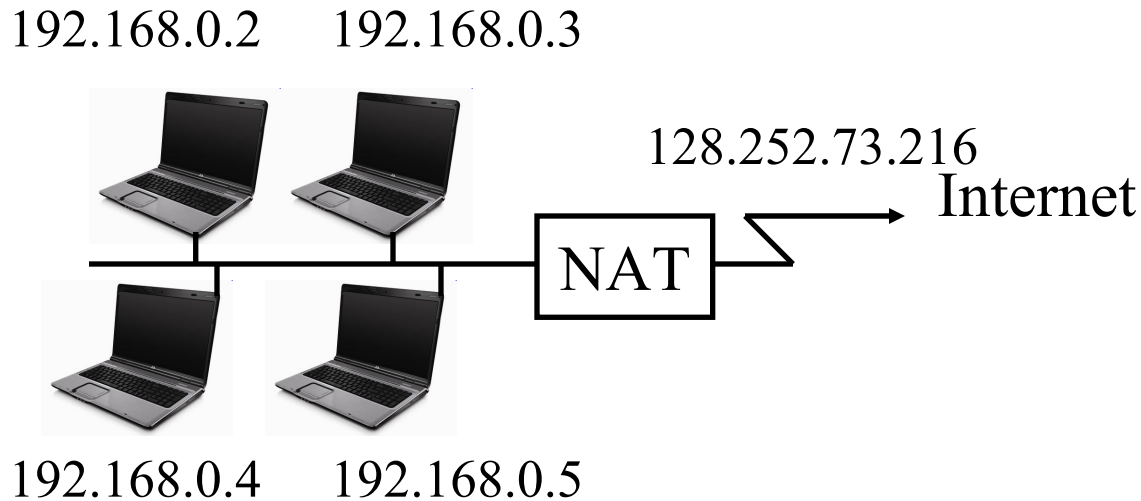
Ref: Section 4.4.2 pages 348-355. Try R15, R16, R17

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)



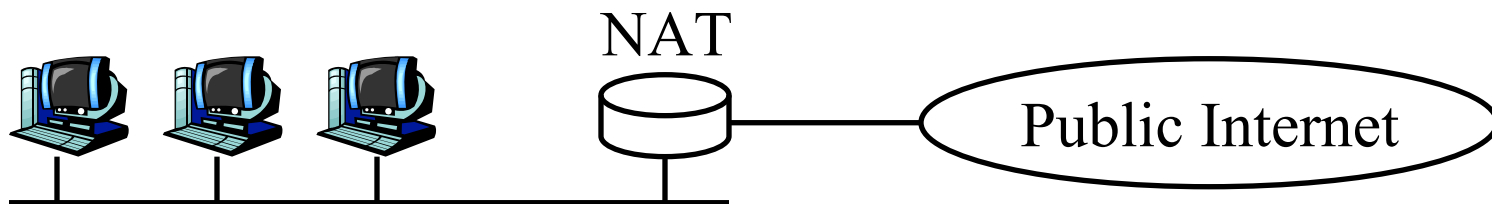
Network Address Translation (NAT)



- ❑ 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)



Ref: Pages 359-362 of the textbook. Try R18.

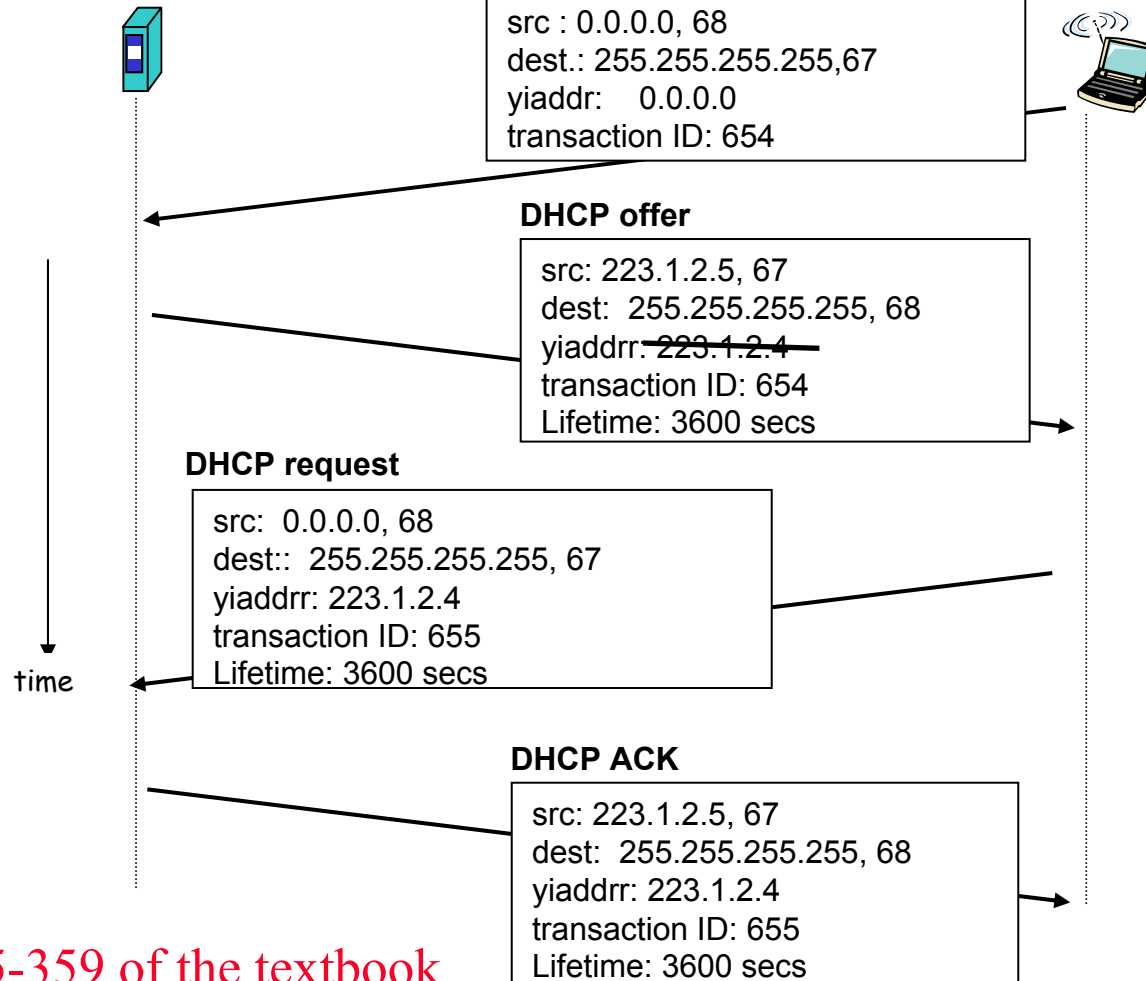
DHCP

- ❑ 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

DHCP Example

DHCP server: 223.1.2.5

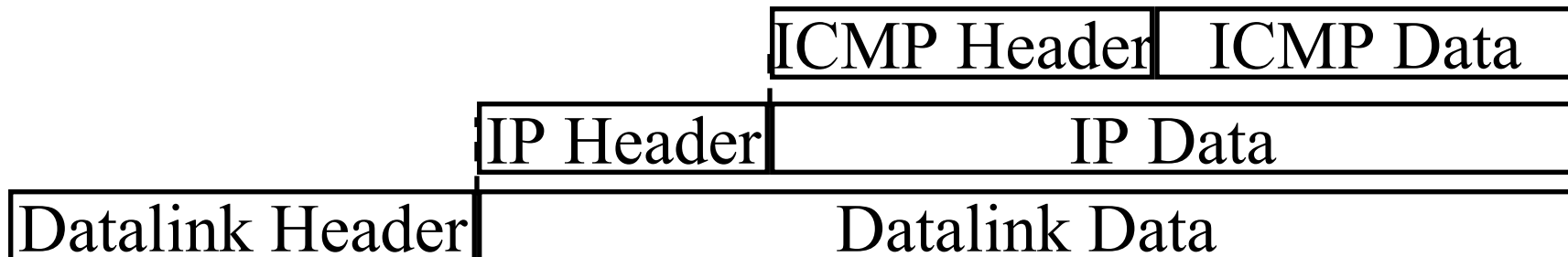
arriving client



Ref: Pages 355-359 of the textbook

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



ICMP: Message Types

IP Header	
Type of Message	8b
Error Code	8b
Checksum	16b
Parameters, if any	Var
Information	Var

Type	Message
0	Echo reply
3	Destination unreachable
4	Source quench
5	Redirect
8	Echo request
11	Time exceeded
12	Parameter unintelligible
13	Time-stamp request
14	Time-stamp reply
15	Information request
16	Information reply
17	Address mask request
18	Address mask reply

ICMP Messages

- ❑ Source Quench: Please slow down! I just dropped one of your datagrams.
- ❑ Time Exceeded: Time to live field in one of your packets became zero.” or “Reassembly timer expired at the destination.
- ❑ Fragmentation Required: Datagram was longer than MTU and “No Fragment bit” was set.
- ❑ Address Mask Request/Reply: What is the subnet mask on this net? Replied by “Address mask agent”
- ❑ PING uses ICMP echo
- ❑ Tracert uses TTL expired

Ref: Section 4.4.3

Trace Route Example

```
C:\>tracert www.google.com
```

```
Tracing route to www.l.google.com [74.125.93.147]  
over a maximum of 30 hops:
```

1	3 ms	1 ms	1 ms	192.168.0.1
2	12 ms	10 ms	9 ms	bras4-10.stlsmo.sbcglobal.net [151.164.182.113]
3	10 ms	8 ms	8 ms	dist2-vlan60.stlsmo.sbcglobal.net [151.164.14.163]
4	9 ms	7 ms	7 ms	151.164.93.224
5	25 ms	22 ms	22 ms	151.164.93.49
6	25 ms	22 ms	22 ms	151.164.251.226
7	30 ms	28 ms	28 ms	209.85.254.128
8	61 ms	57 ms	58 ms	72.14.236.26
9	54 ms	52 ms	51 ms	209.85.254.226
10	79 ms	160 ms	67 ms	209.85.254.237
11	66 ms	57 ms	68 ms	64.233.175.14
12	60 ms	58 ms	58 ms	qw-in-f147.google.com [74.125.93.147]

```
Trace complete.
```


IPv6

- ❑ Shortage of IPv4 addresses \Rightarrow Need larger addresses
- ❑ IPv6 was designed with 128-bit addresses
- ❑ $2^{128} = 3.4 \times 10^{38}$ addresses
 $\Rightarrow 665 \times 10^{21}$ addresses per sq. m of earth surface
- ❑ If assigned at the rate of $10^6/\mu\text{s}$, 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 Header

□ IPv6:

Version	Priority	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address			
Destination Address			

□ IPv4:

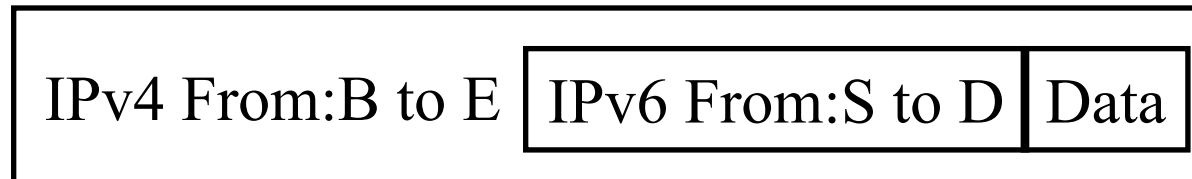
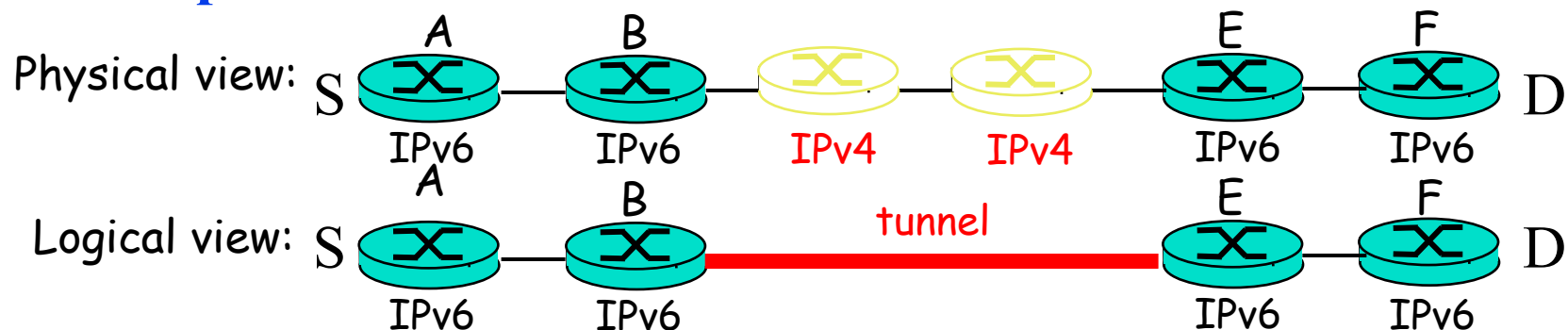
Version	IHL	Type of Service	Total Length	
Identification		Flags	Fragment Offset	
Time to Live	Protocol		Header Checksum	
Source Address				
Destination Address				
Options			Padding	

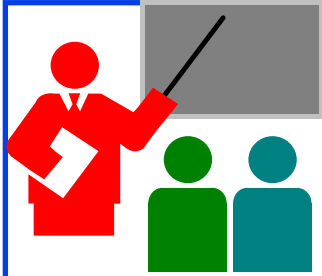
IPv6 vs. IPv4

- ❑ 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
⇒ 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.

Ref: Read Sections 4.4 (pages 341–374) of the textbook. Skip Section 4.4.5. Try R19, R20.



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

Ref: Piscitello and Chapin, p413

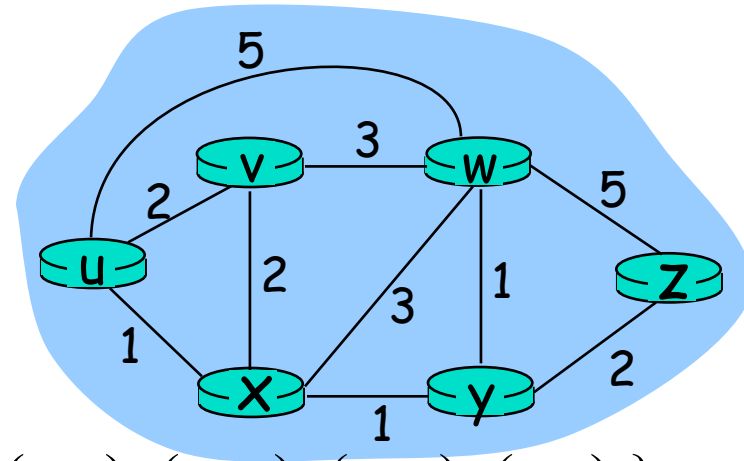
Routeing or Routing

- ❑ 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.

Ref: Piscitello and Chapin, p413

Graph abstraction

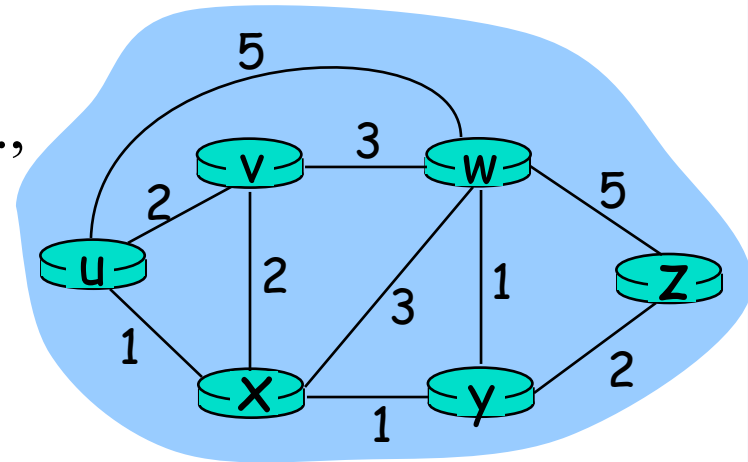
- ❑ Graph: $G = (N, E)$
- ❑ $N =$ Set of routers
 $= \{ u, v, w, x, y, z \}$
- ❑ $E =$ Set of links
 $= \{ (u, v), (u, x), (v, x), (v, w), (x, w), (x, y), (w, y), (w, z), (y, z) \}$
- ❑ Each link has a cost, e.g., $c(w, z) = 5$
- ❑ Cost of path $(x_1, x_2, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$
- ❑ 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.



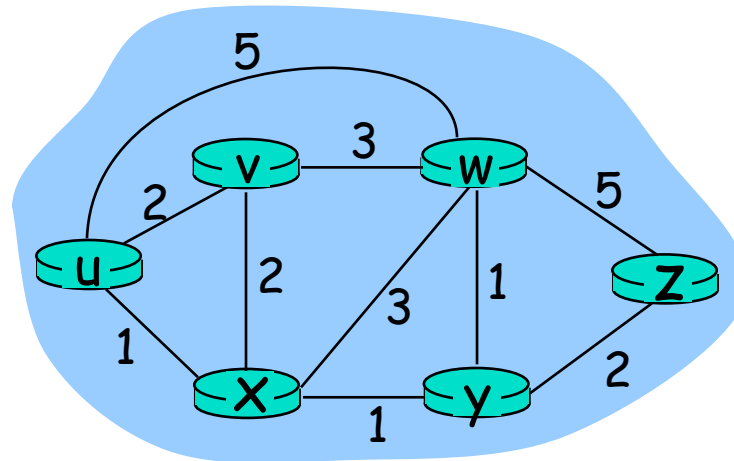
Link State:

- ❑ 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

node x table

		cost to		
		x	y	z
from	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

node y table

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

node z table

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	7	1	0

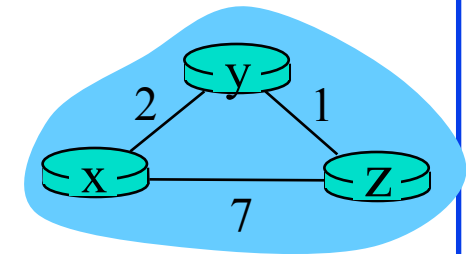
		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

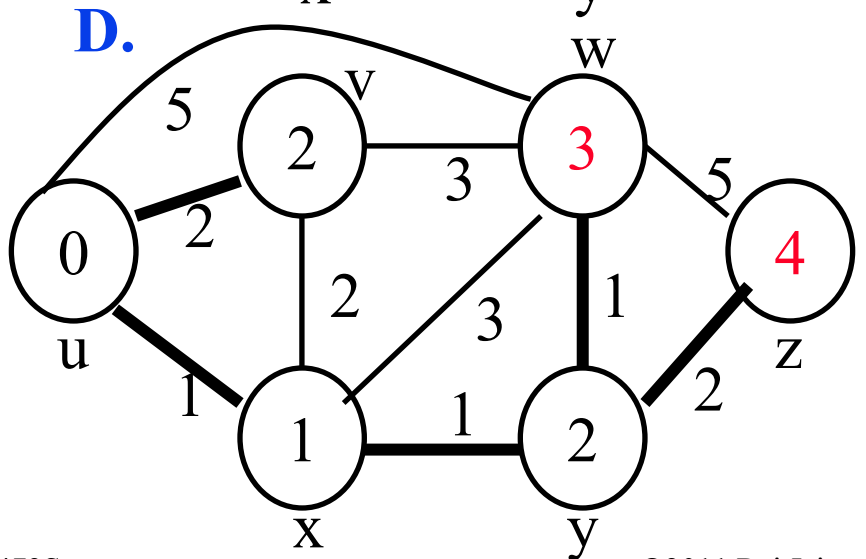
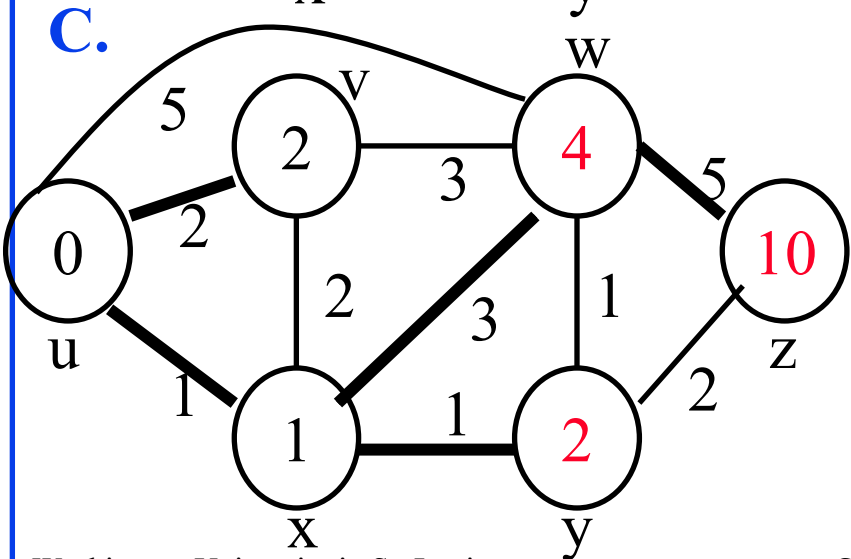
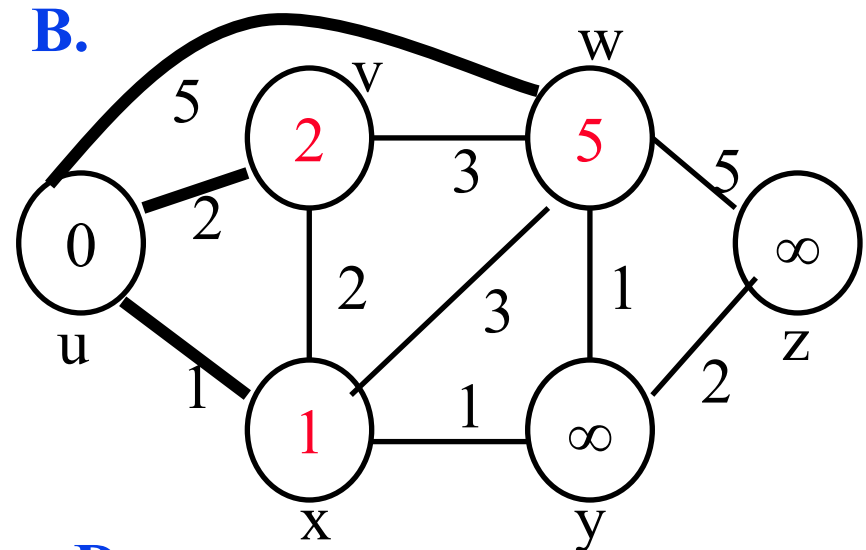
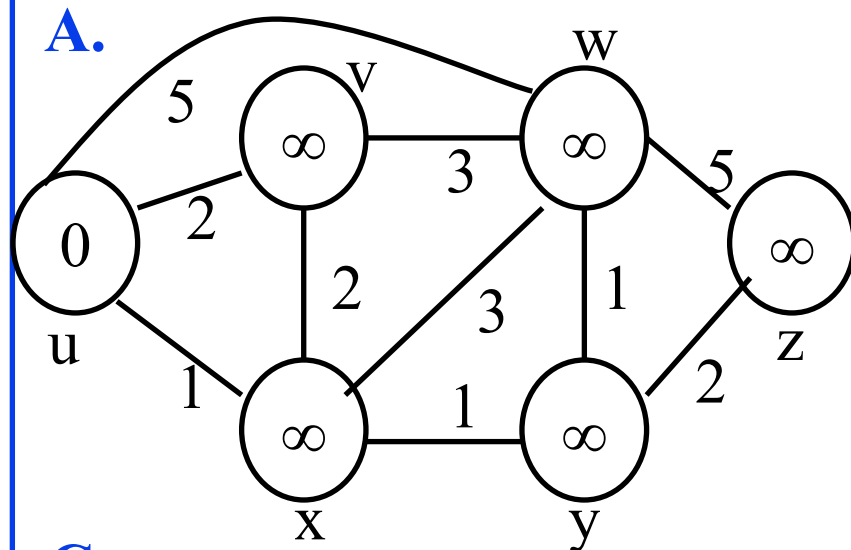
		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

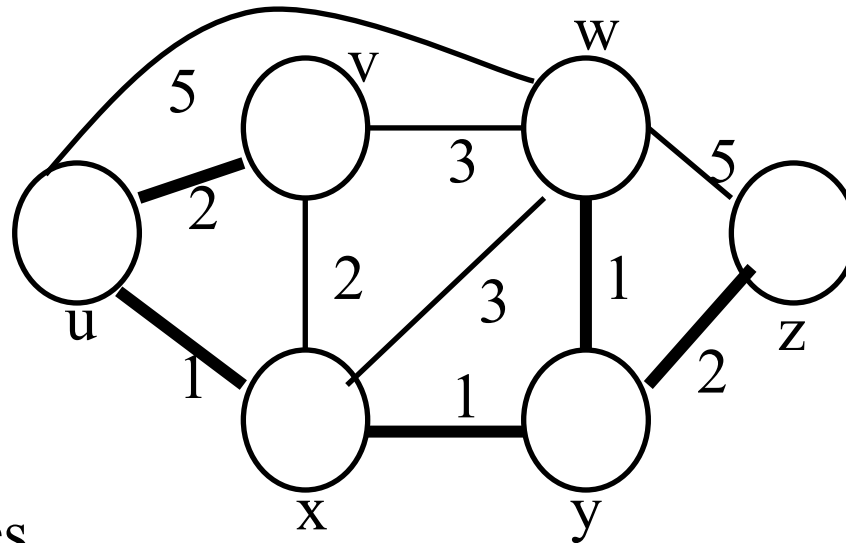


► time

Bellman-Ford Example 2



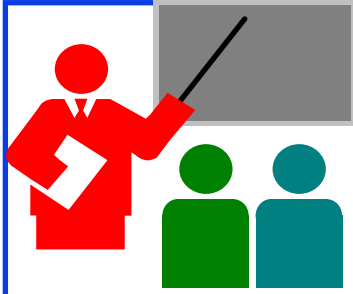
Bellman-Ford: Tabular Method



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

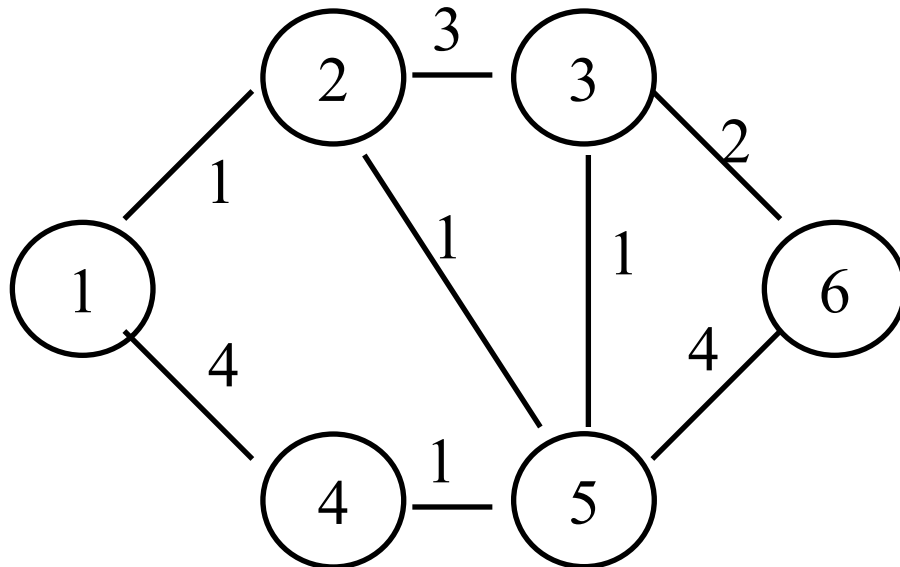
1. Distance Vectors: Distance to all nodes in the network sent to neighbors
2. Link State: Cost of link to neighbors sent to entire network
3. Dijkstra's algorithm is used to compute shortest path using link state
4. Bellman Ford's algorithm is used to compute shortest paths using distance vectors

Ref: Read Section 4.5 (pages 374–393) of the textbook.

Section 4.5.3 is covered in the next part. Try R21 (Not ready for R22 or R23)

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

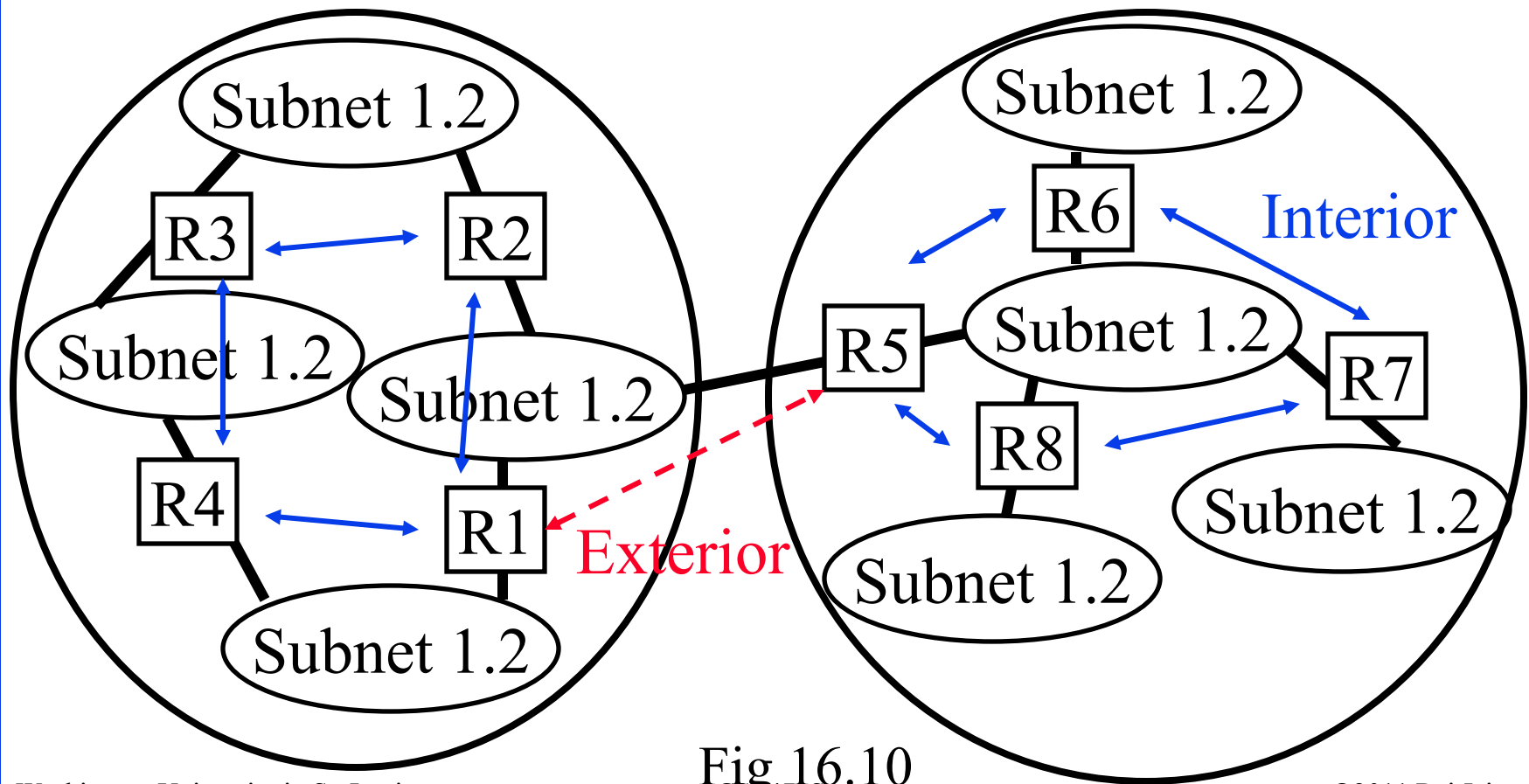


Fig 16.10

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.

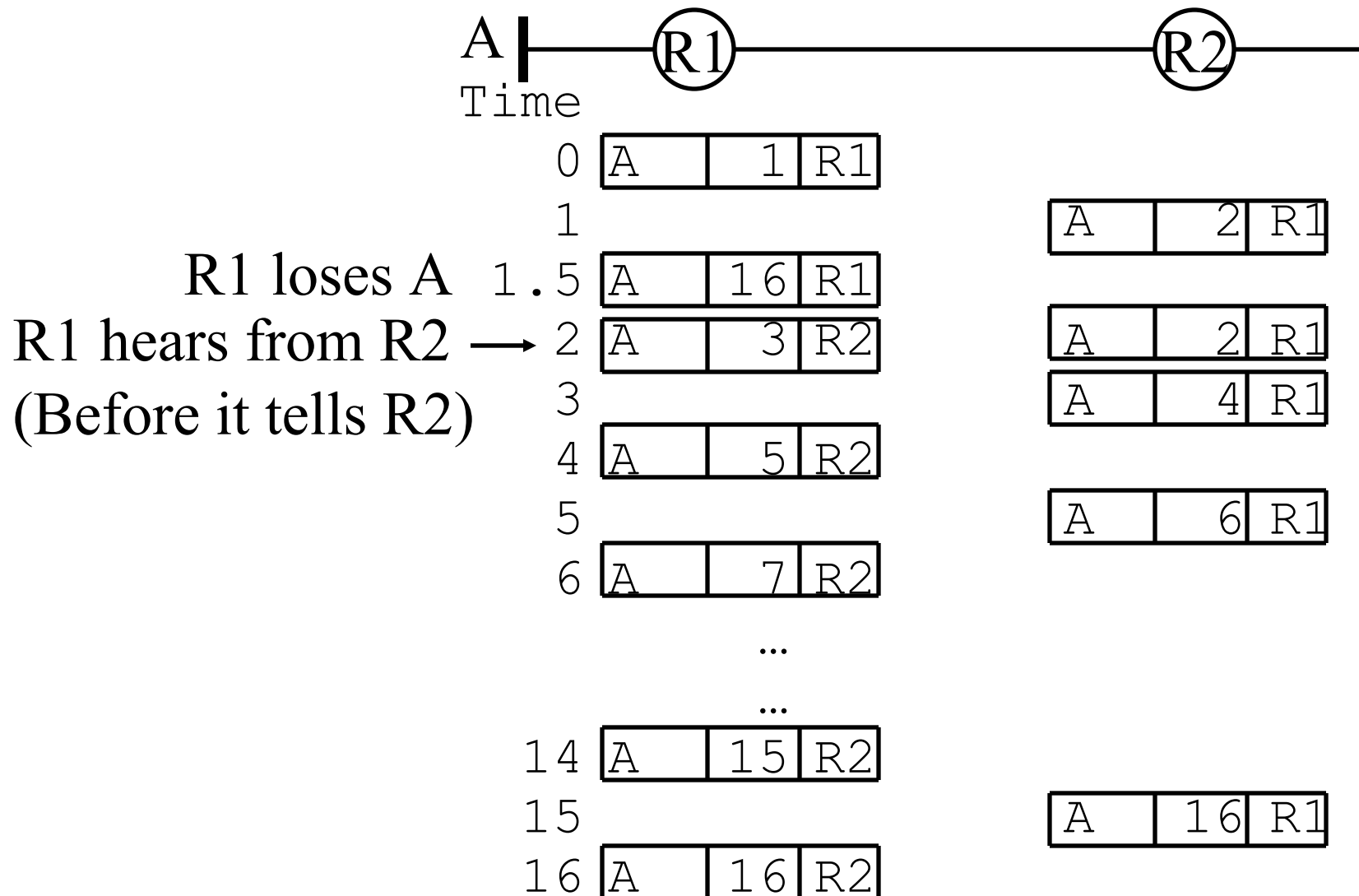
Routing Information Protocol

- ❑ 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

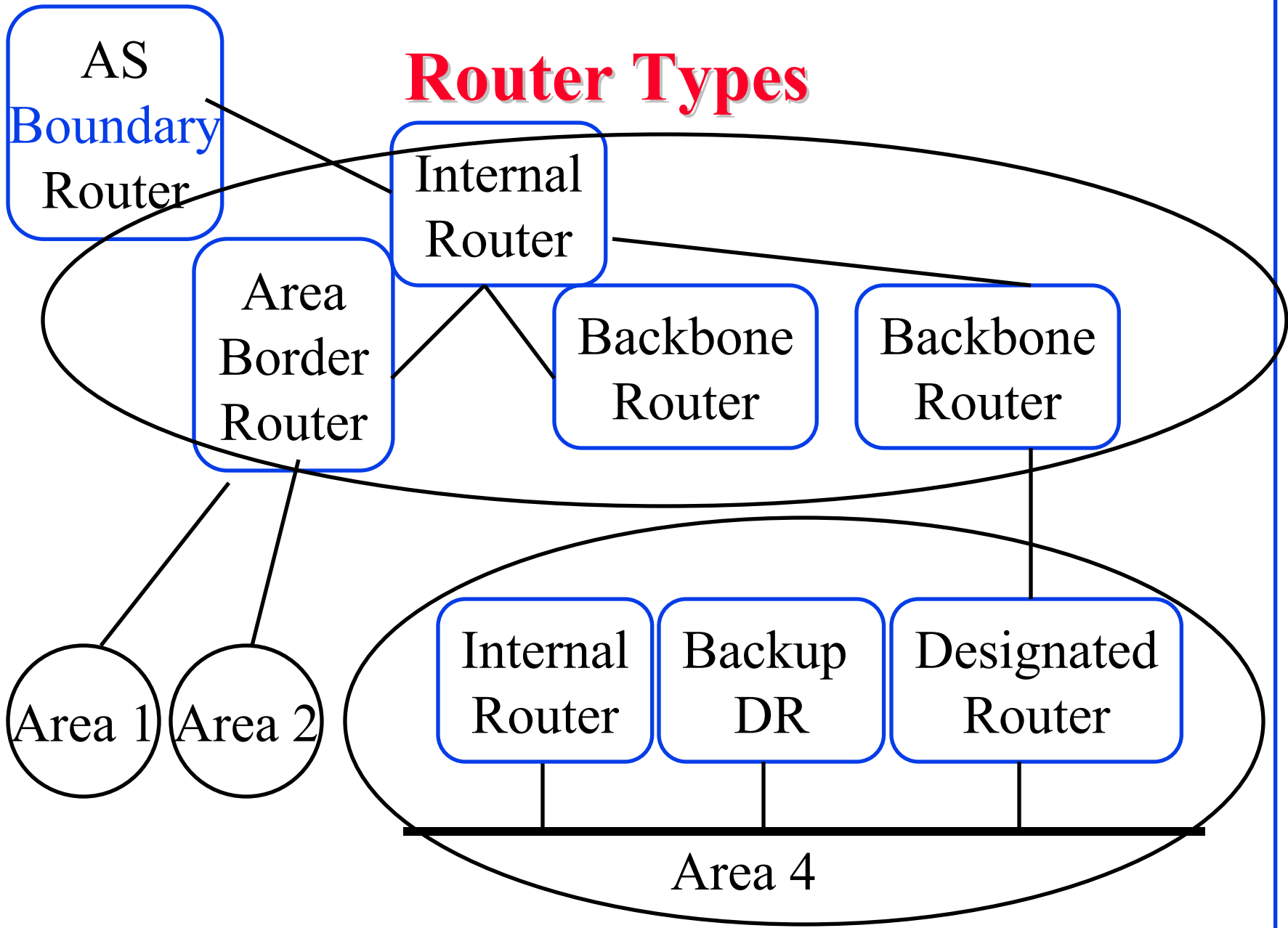
Counting to Infinity Problem



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.

Metrics (Cost)

- RFC 1253: Metric = $10^8/\text{Speed}$

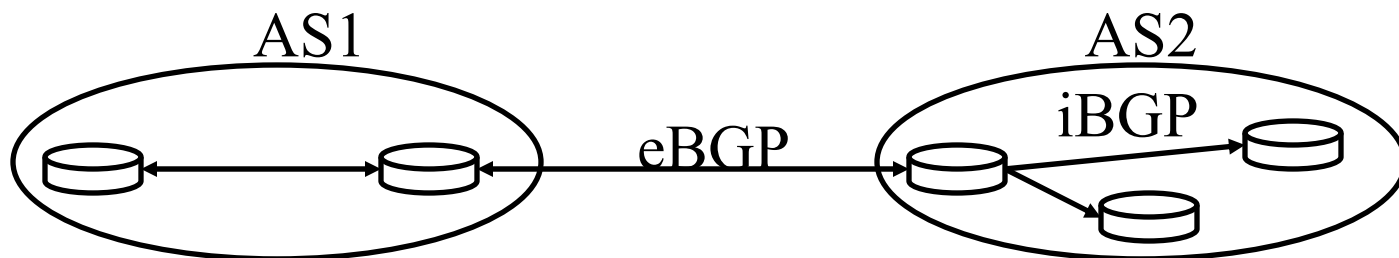
Bit Rate	Metric
9.6 kbps	10,416
19.2 kbps	5208
56 kbps	1785
64 kbps	1562
T1 (1.544 Mbps)	65
E1 (2.048 Mbps)	48
Ethernet/802.3 (10 Mbps)	10
100 Mbps or more	1

Maintaining the Database

- ❑ Databases are continually checked for synchronization by flooding LSAs
- ❑ All flooded LSAs are acked. Unacked LSAs are flooded again.
- ❑ Database information is checked. If new info, it is forwarded to other adjacencies using LSAs.
- ❑ When an entry is aged out, the info is flooded.
- ❑ Dijkstra's algorithm is run on every new info, to build new routing tables.

Border Gateway Protocol

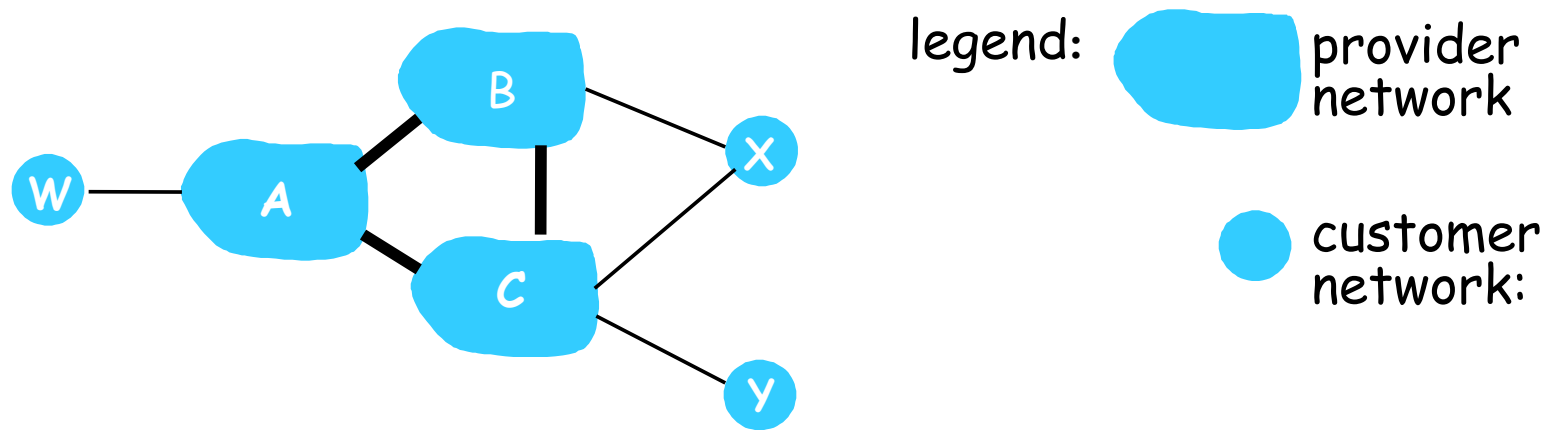
- ❑ Inter-autonomous system protocol [RFC 1267]
- ❑ Used since 1989 but not extensively until recently
- ❑ Runs on TCP (segmentation, reliable transmission)
- ❑ Advertises all transit ASs on the path to a destination address
- ❑ A router may receive multiple paths to a destination \Rightarrow Can choose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



BGP Operations

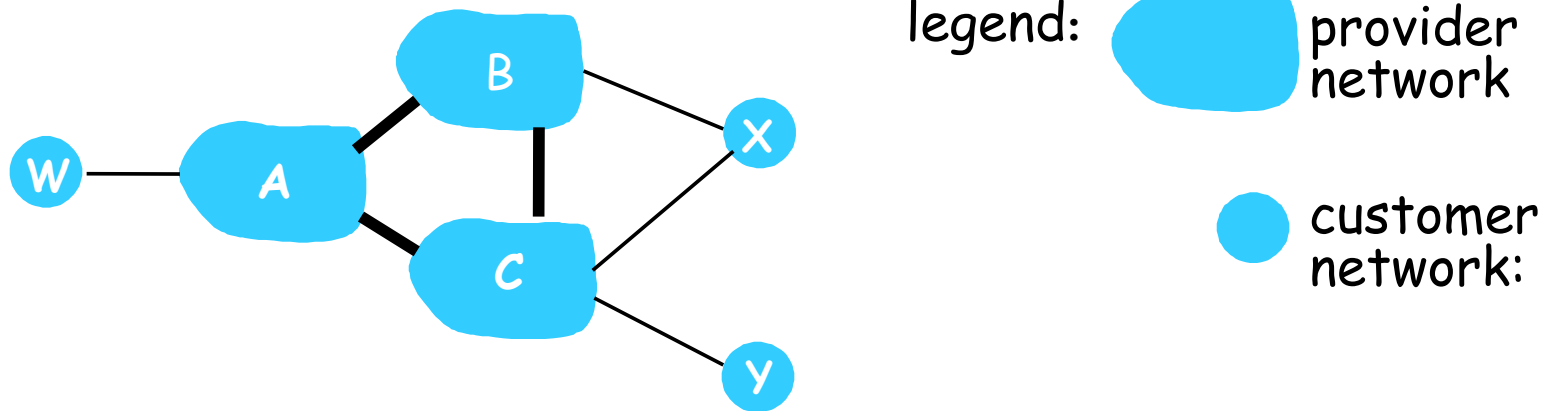
- ❑ BGP systems initially exchange entire routing tables. Afterwards, only updates are exchanged.
- ❑ BGP messages have the following information:
 - ❑ Origin of path information: RIP, OSPF, ...
 - ❑ AS_Path: List of ASs on the path to reach the dest
 - ❑ Next_Hop: IP address of the border router to be used as the next hop to reach the dest
 - ❑ Unreachable: If a previously advertised route has become unreachable
- ❑ BGP speakers generate update messages to all peers when it selects a new route or some route becomes unreachable.

BGP Routing Policy Example



- ❑ A,B,C are **provider networks**
- ❑ X,W,Y are customer (of provider networks)
- ❑ X is **dual-homed**: attached to two networks
 - ❑ X does not want to route from B via X to C
 - ❑ .. so X will not advertise to B a route to C

BGP Routing Policy Example (Cont)



- ❑ A advertises path A-W to B
- ❑ B advertises path B-A-W to X
- ❑ Should B advertise path B-A-W to C?
 - ❑ No way! B gets no “revenue” for routing C-B-A-W since neither W nor C are B’s customers
 - ❑ B wants to force C to route to w via A
 - ❑ B wants to route *only* to/from its customers!

Intra- vs. Inter-AS Routing

□ Policy:

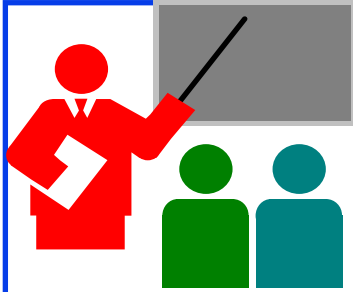
- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

□ 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

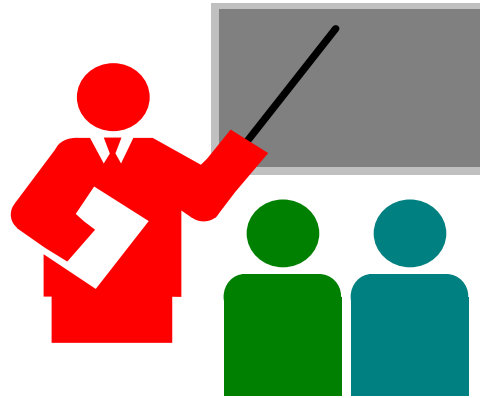
1. RIP uses distance-vector routing
2. RIP v2 fixes the slow convergence problem
3. OSPF uses link-state routing and divides the autonomous systems into multiple areas.
Area border router, AS boundary router, designated router
4. BGP is an inter-AS protocol \Rightarrow Policy driven

Ref: Read Section 4.6 (pages 393-407) of the textbook.

Section 4.7 is not covered in this course.

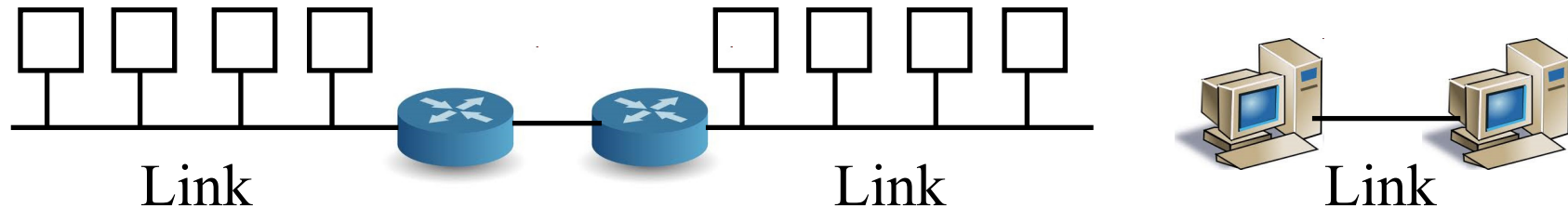
Try R24, R25, R26, R27, R29, R30

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

Link Local Addresses



- ❑ Link Local IP addresses can be used on a link (point-to-point or point-to-multipoint Ethernet) to communicate with other nodes on that link
- ❑ The datagrams with Link-Local source or destination addresses will not be forwarded out of that link
- ❑ Address range 169.254.1.0 through 169.254.254.255 is reserved for link local IP address. This is 169.254.0.0/16 with the first 255 and the last 255 addresses reserved for future use.
- ❑ A node assigns a random address from this range and ARPs to ensure that no one else has that address.
- ❑ A set of nodes when directly connected can talk to each other even when there is no DHCP server.

Ref: http://en.wikipedia.org/wiki/Link-local_address