Chapter 4
Network Layer

- Teacher: Xu Yang

Chapter 4: network layer

chapter goals:

- understand principles behind network layer services:
  - network layer service models
  - forwarding versus routing
  - how a router works
  - routing (path selection)
  - broadcast, multicast
- instantiation, implementation in the Internet

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

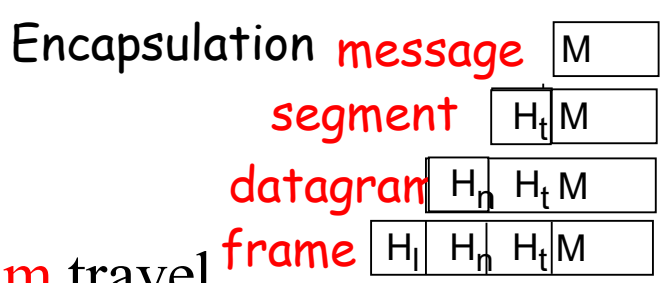
- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

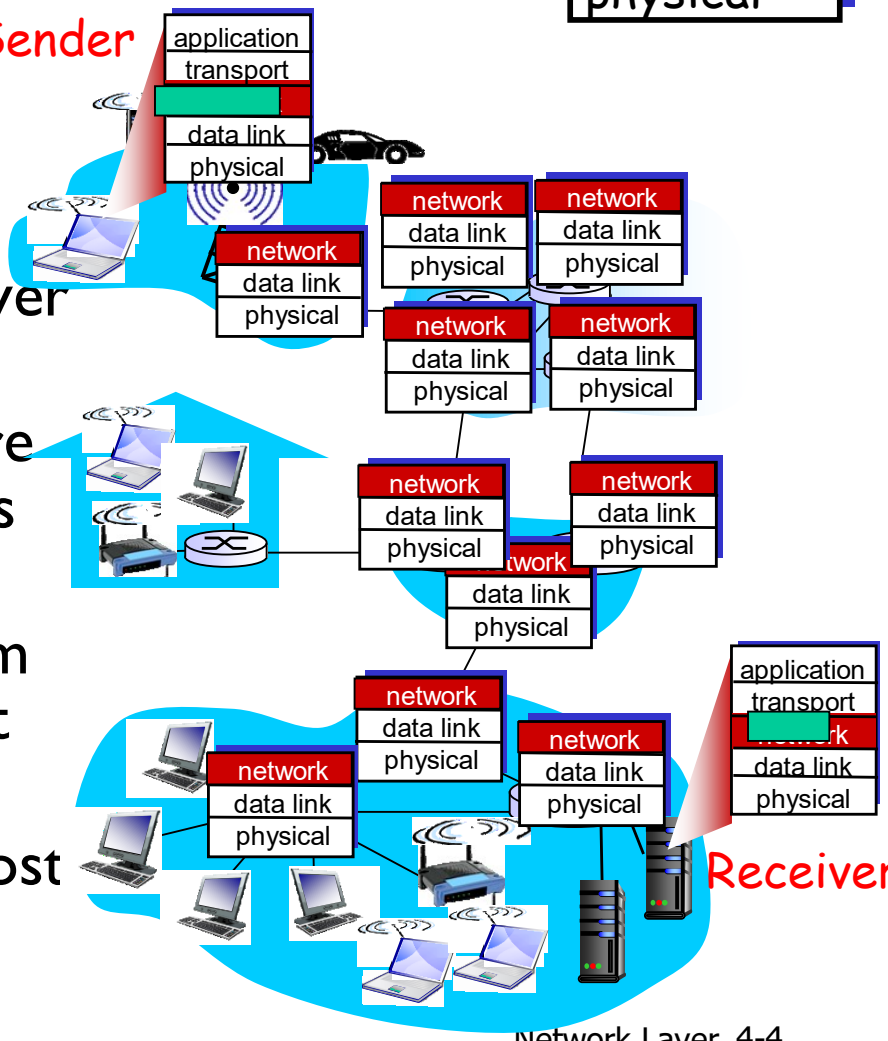
4.7 broadcast and multicast routing

Network layer



How does network layer datagram travel from sending host to receiving host ?

- sender side: network layer takes segments from the transport layer, encapsulates each segment into a datagram (the name of network-layer packet), passes to the link layer.
- Intermediate routers: datagrams are routed through intermediate nodes (routers)
- receiver side: receives the datagram from lower layer, extracts segment and passes to the transport layer.
- network layer protocols in *every* host router : provide logical communication between hosts

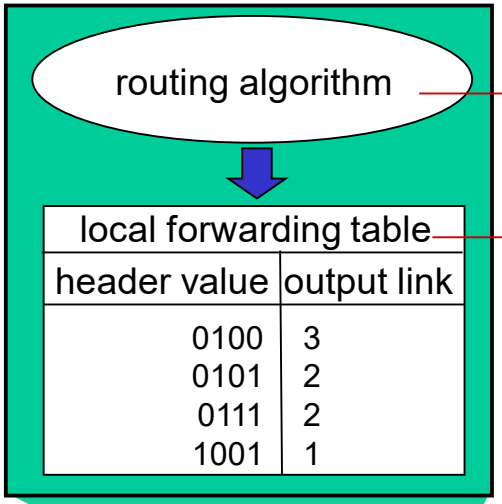


Two key network-layer functions

- *forwarding:* move network layer packets from router's input to appropriate router output
  - Routers have a forwarding table.

- *routing:* determine the path of packets as they flow from a sender to a receiver
  - Routing algorithm: run at routers to determine the path of packets

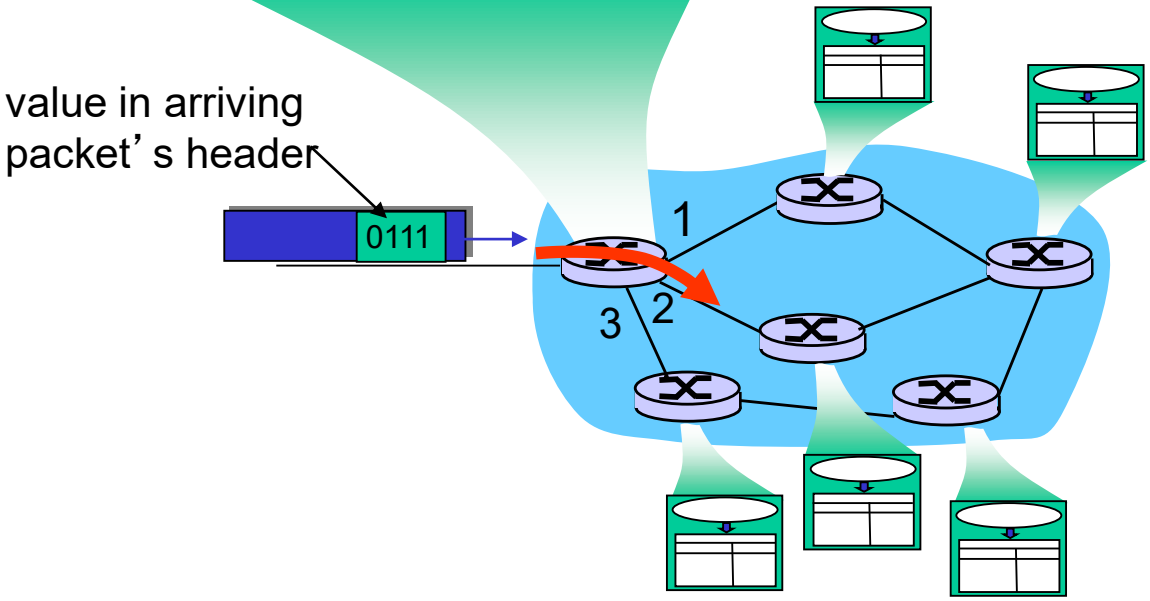
Interplay between routing and forwarding



routing algorithm determines end-end-path through network

forwarding table determines local forwarding at this router

Routing algorithms determine values in forwarding tables。



Connection setup

- 3rd important network-layer function in *some* network architectures: connection setup
  - ATM, frame relay, X.25
- before datagrams flow, two end hosts *and* intervening routers establish virtual connection
  - routers get involved
- network vs transport layer connection service:
  - *network:* between two hosts (may also involve intervening routers in case of VCs)
  - *transport:* between two processes

Network service model

In terms of quality of service, network-layer protocol may provide the following possible service for host-to-host transmission

- In terms of individual datagram
  - Guaranteed delivery. This service guarantees that the packet will eventually arrive at its destination.
  - Guaranteed delivery with bounded delay. This service not only guarantees delivery of the packet, but delivery within a specified host-to-host delay bound (for example, within 100 msec).
- In terms of a flow of datagram
  - In-order datagram delivery: this service guarantees that packets arrive at the destination in the order that they were sent.
  - Guaranteed minimum bandwidth to flow. This network-layer service emulates the behavior of a transmission link of a specified bit rate (for example, 1 Mbps) between sending and receiving hosts.

Network layer service models:

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

ATM: Asynchronous Transfer Mode
CBR: constant bit rate; VBR: variable bit rate;
ABR: available bit rate; UBR: unspecified bit rate

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Two types of Network Architecture

- *Connection-Oriented* and *Connection-Less*

Virtual Circuit (VC) Switching

Example:ATM
Analogy: Telephone

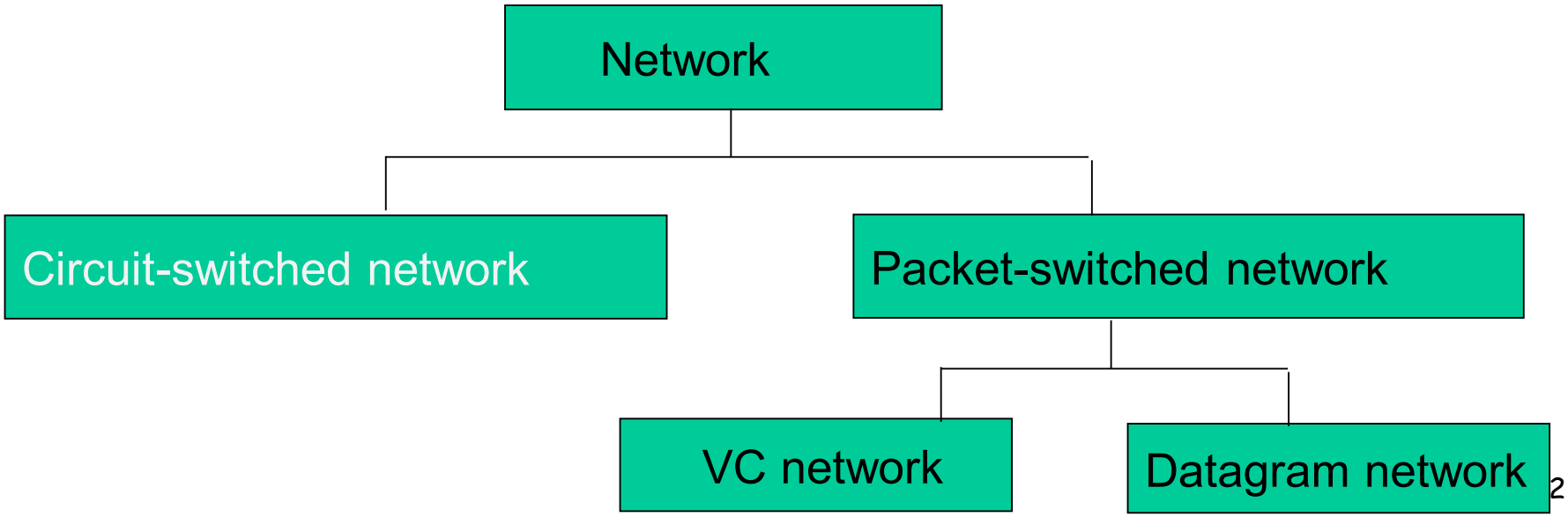
Datagram Forwarding

Example: IP networks
Analogy: Postal service

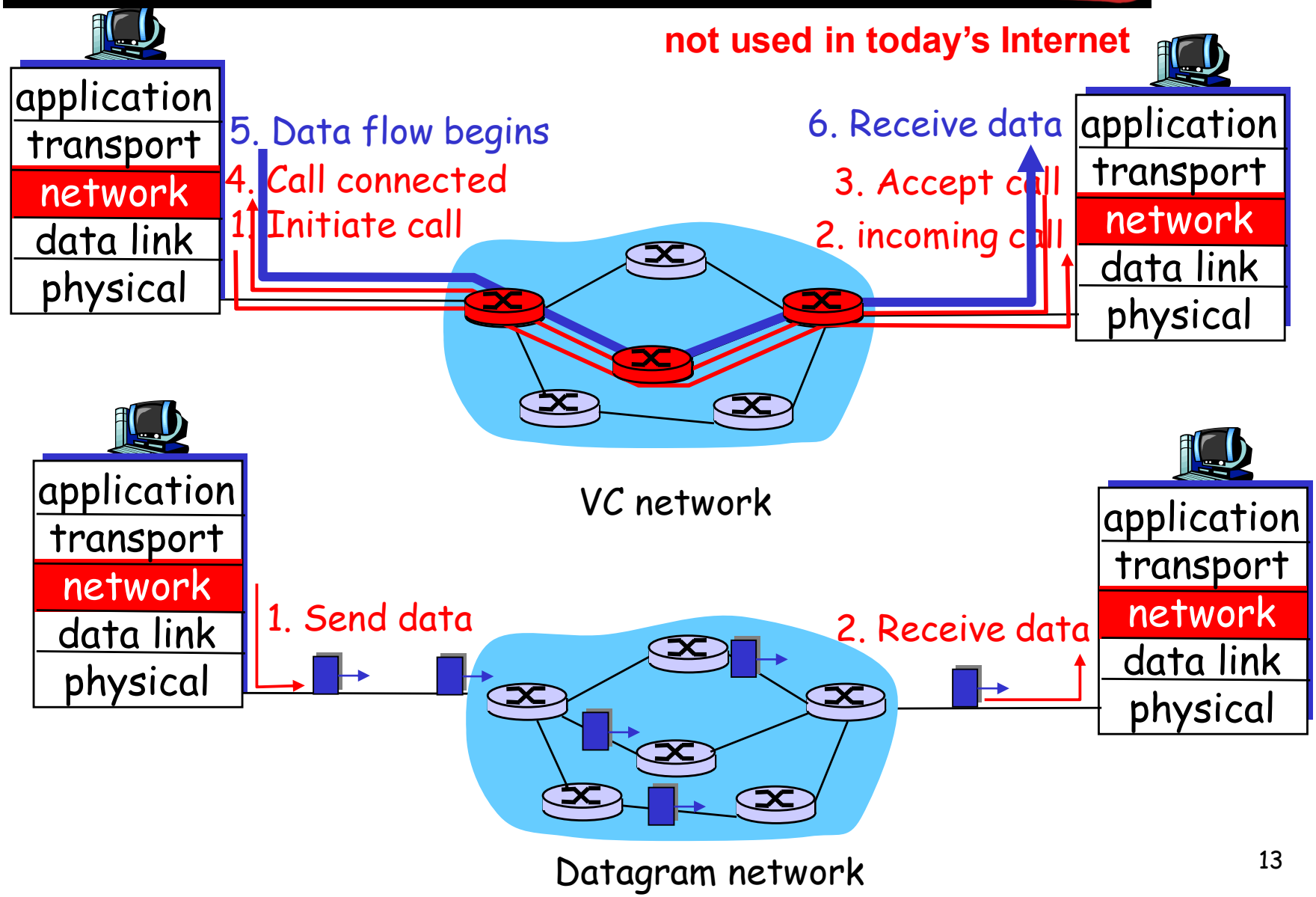
service: host-to-host
no choice: network provides one or the other
implementation: in network core

Network layer connection and connection-less service

- Both datagram network and virtual circuit (VC) network belong to packet-switched network
- Virtual circuit (VC) network provides network-layer connection service (ATM)
- Datagram network provides network-layer connectionless service (Internet)

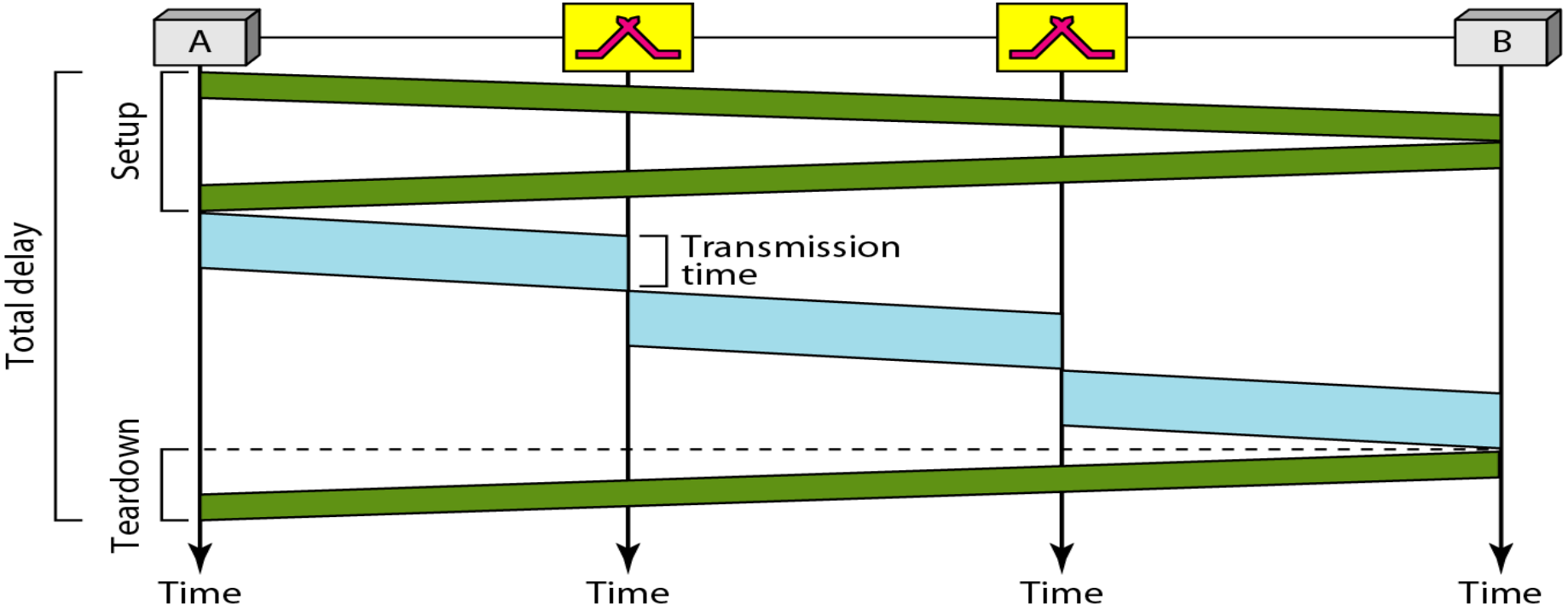


Virtual Circuit (VC) Network and Datagram Network



Virtual Circuit (VC) Network

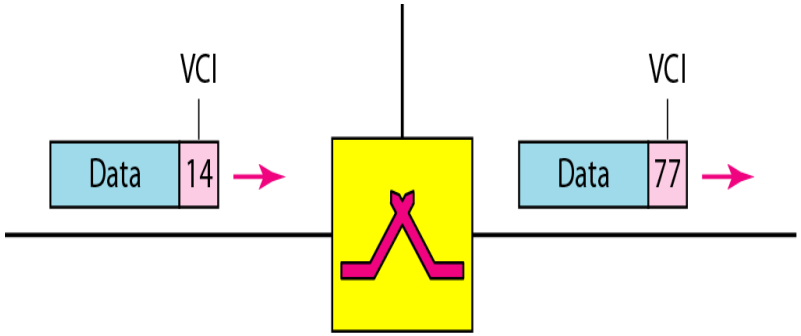
- Have to setup and teardown a virtual circuit for each connection *before* and after data transfer, respectively.

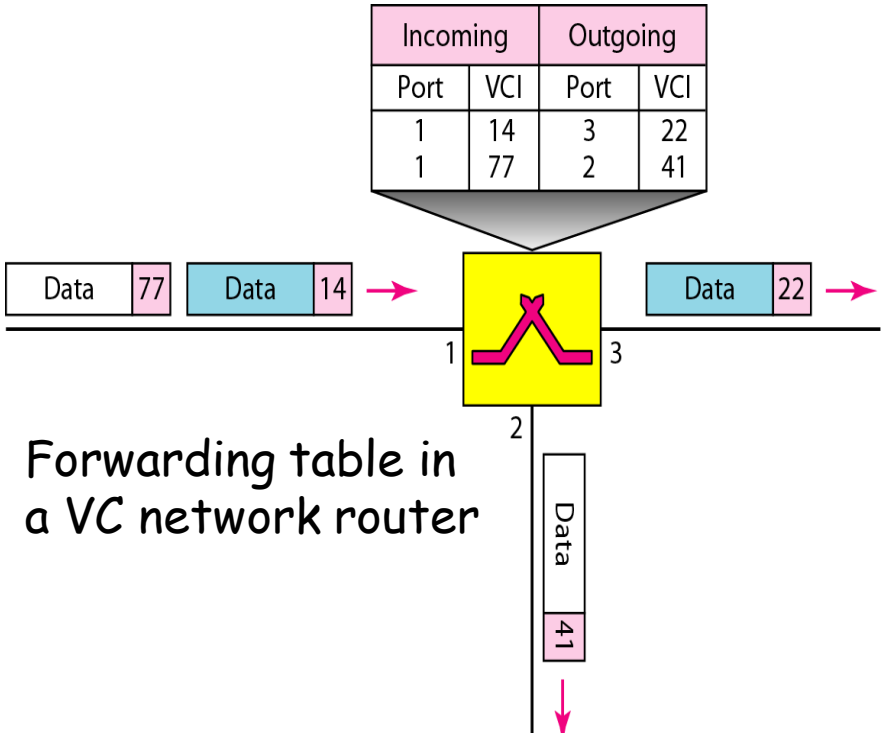


Virtual Circuit (VC) Network

- A VC consists of
  - Path from source to destination
  - VC identifiers/numbers, one number for each link along path
  - Entries in forwarding tables in routers along path
- Each packet carries VC identifier (not destination host address)
- *VC routers maintain connection state information!*

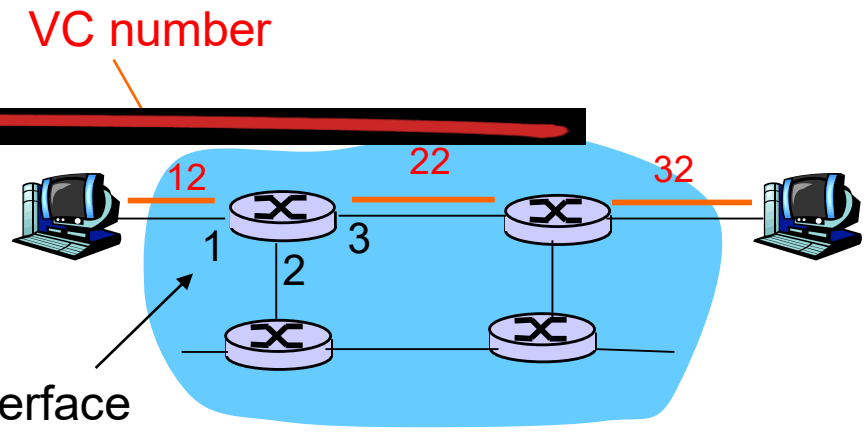
- VC identifier will be changed on each link
  - New VC number comes from forwarding table





Forwarding table in a VC network router

Forwarding table



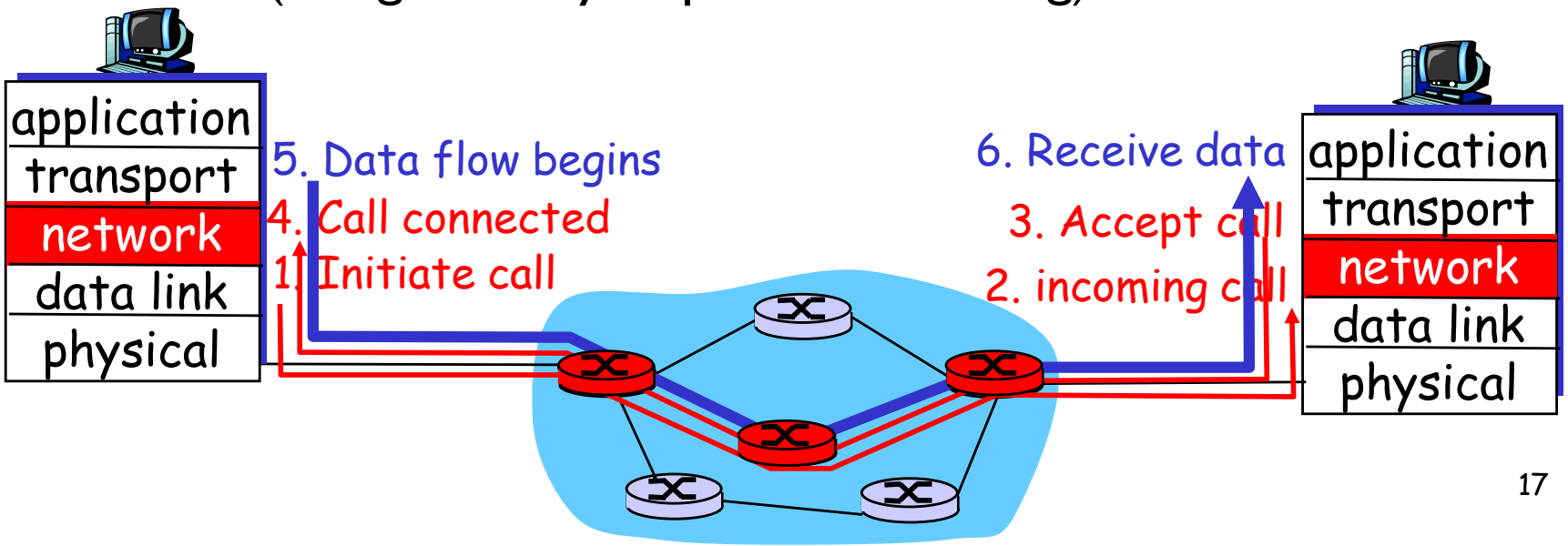
Forwarding table in northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

Routers maintain connection state information!

Virtual Circuit (VC) Network

- Every router on source-destination path maintains "state" for each passing connection
- In VC switching, all packets belonging to the same source and destination pass the same path, but packets may arrive at the destination with different delays if resource allocation is on demand.
- Router resources (bandwidth, buffers) may be allocated to VC using the way of reservation; router resources may be allocated on demand (using the way of packet switching).



Virtual Circuit (VC) Network

- A VC network
  - Have to setup and teardown a virtual circuit for each connection before and after data transfer, respectively.
  - "Source-to-destination" path behaves much like "telephone circuit (real circuit)" (e.g., setup, teardown, etc)
  - Possibly on demand resource allocation at the intermediate router (packet switching).
  - Router resources may be allocated on demand (using the way of packet switching).
  - In VC switching, all packets belonging to the same source and destination follow the same path, but packets may arrive at the destination with different delays if resource allocation is on demand.
- VC Network-signaling protocols:
  - used to setup, maintain teardown VC
  - used in ATM, frame-relay, X.25
  - not used in today's Internet

Datagram networks

- no connection setup at network layer
- Routers do not maintain state about end-to-end connections
  - no network-level concept of "connection"
  - uses a routing/forwarding table that is based on the destination host address to forward packets.
  - The destination address in the header remains the same during the entire delivery of packet.
- packets between same source-destination pair may take different paths (due to the forwarding table update)

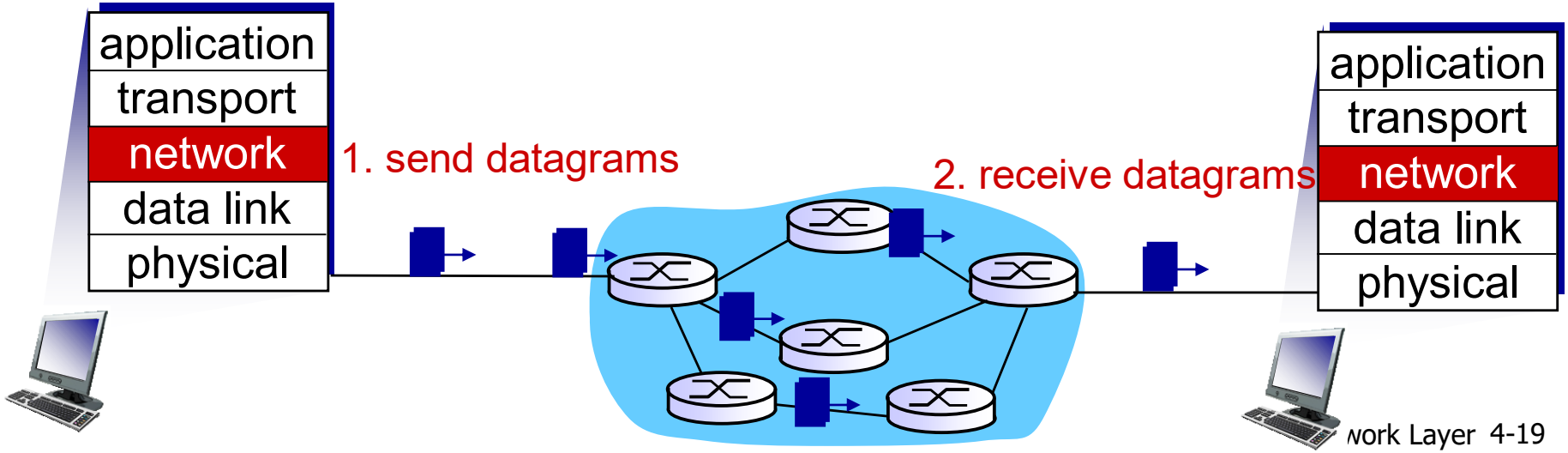
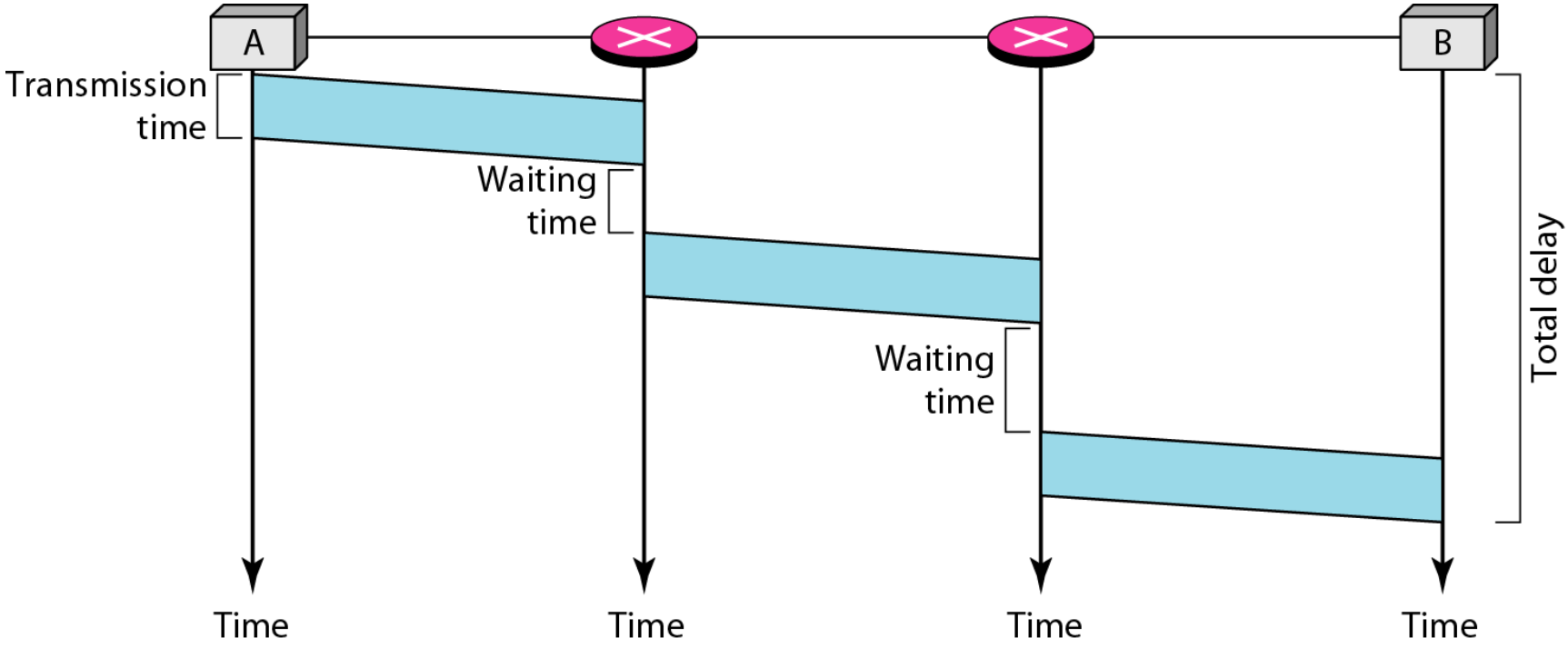
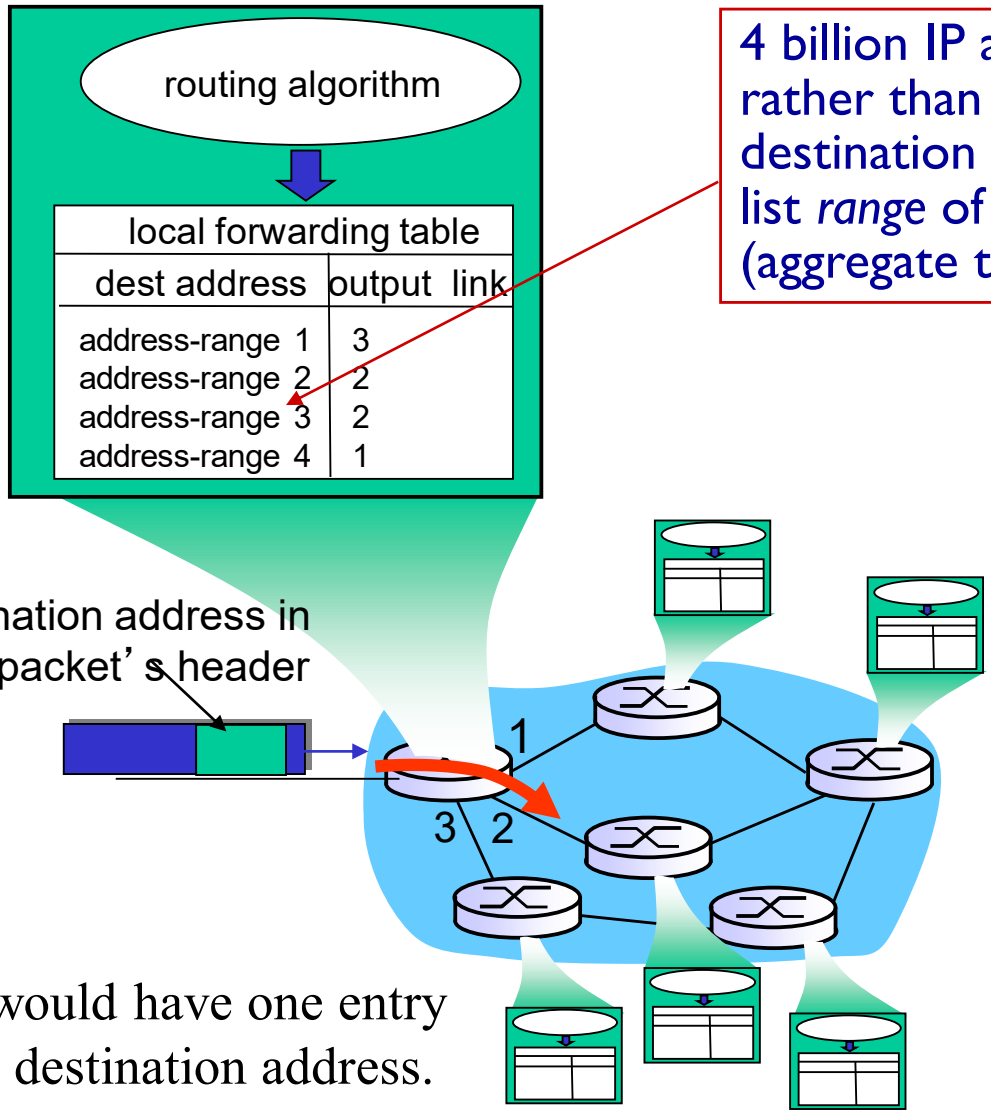


Figure 8.9 *Delay in a datagram network*



Datagram forwarding table



forwarding table would have one entry for every possible destination address.

Longest prefix matching

longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

DA: 11001000 00010111 00010110 10100001 which interface?

DA: 11001000 00010111 00011000 10101010 which interface?

Datagram or VC network: why?

Internet (datagram)

- Simple
- data exchange among computers
  - "elastic" service, provide best effort services, no strict timing req.
- many link types
  - different characteristics
  - uniform service difficult
- "smart" end systems (computers)
  - can adapt, perform control, error recovery
  - ***simple inside network, complexity at "edge"***

ATM (VC)

- complicated
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- "dumb" end systems
  - telephones
  - ***complexity inside network while simple at end systems***

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

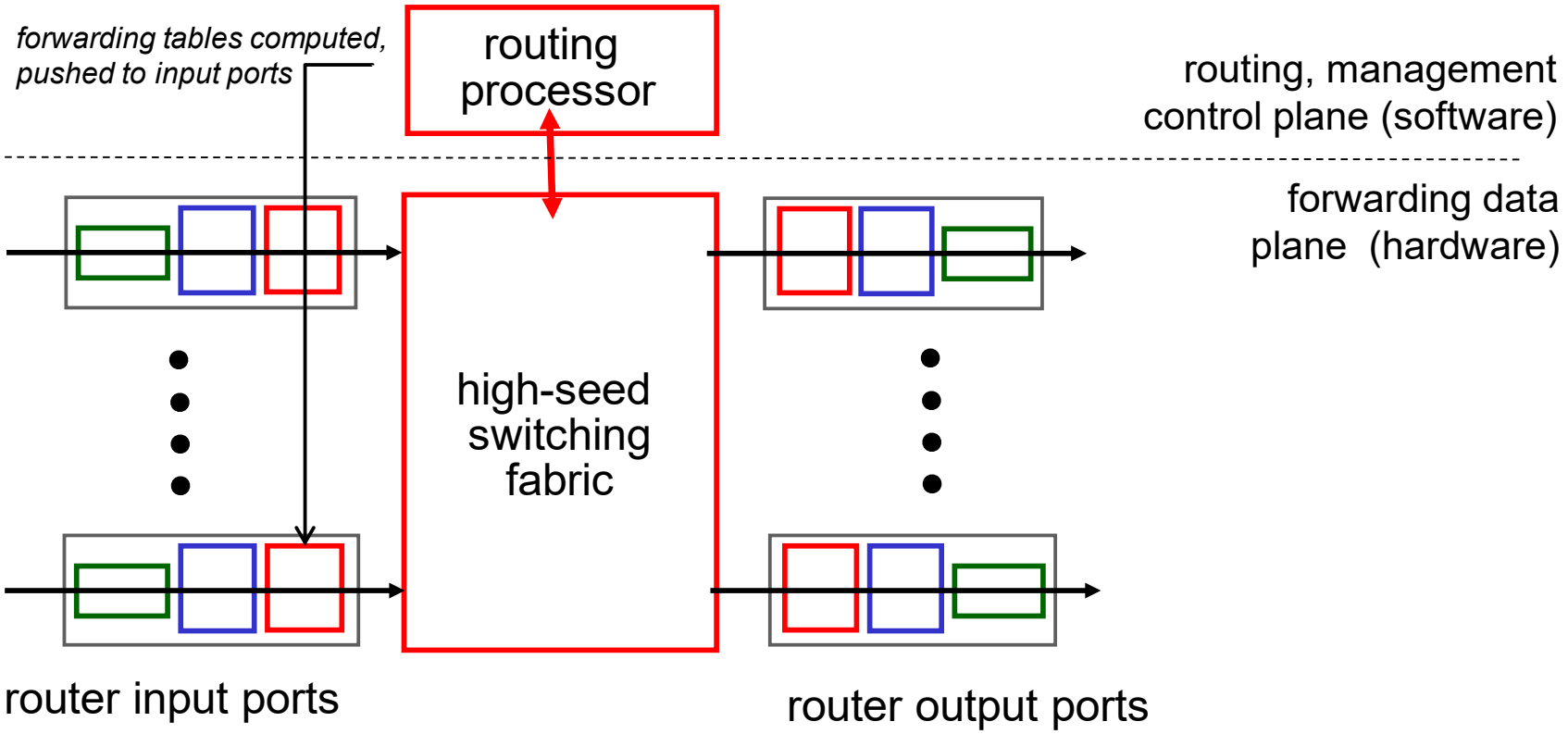
- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Router architecture overview

two key router functions:

- run routing algorithms/protocol (RIP, OSPF, BGP)
- *forwarding* datagrams from incoming to outgoing link



Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

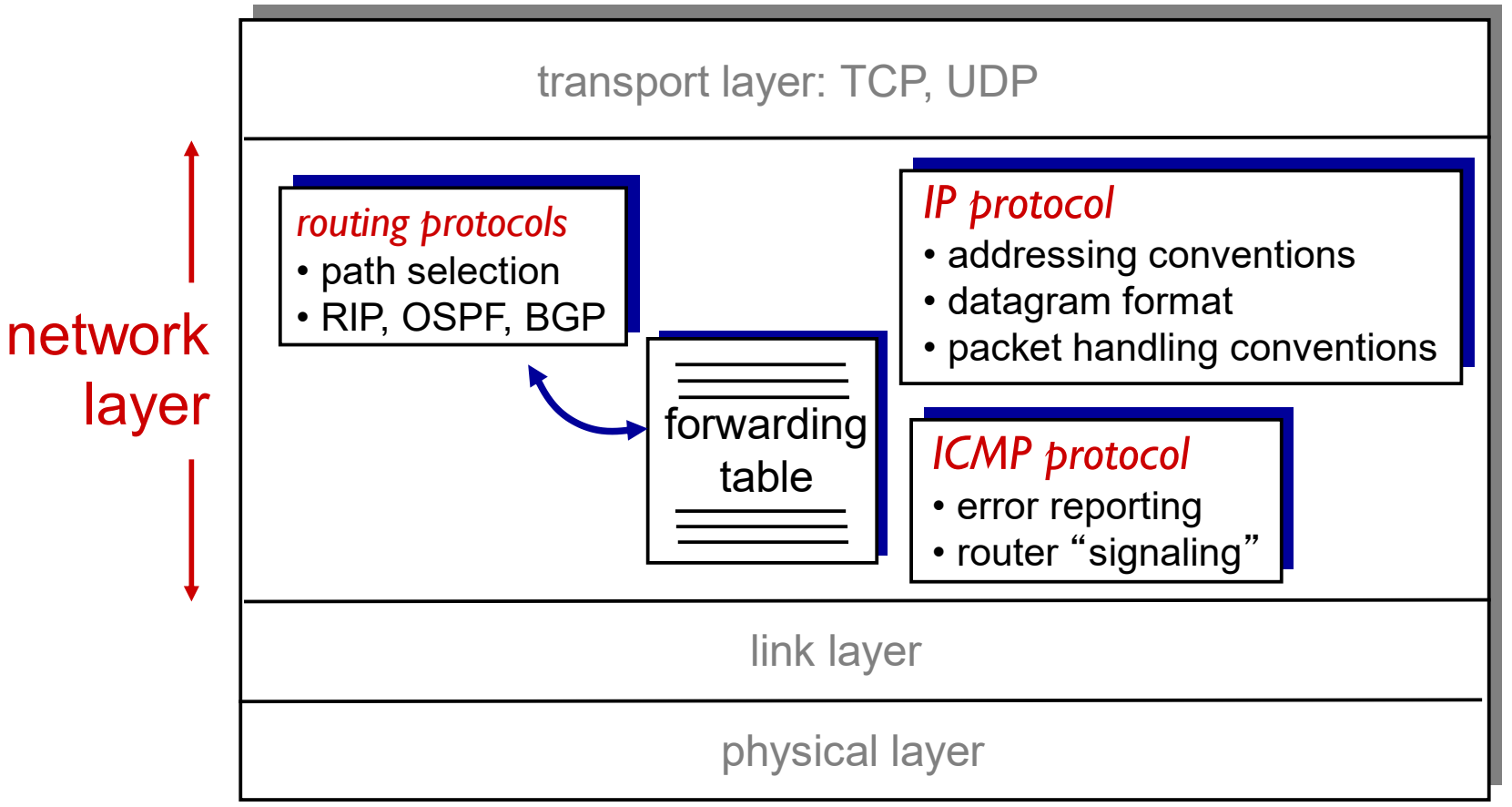
4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

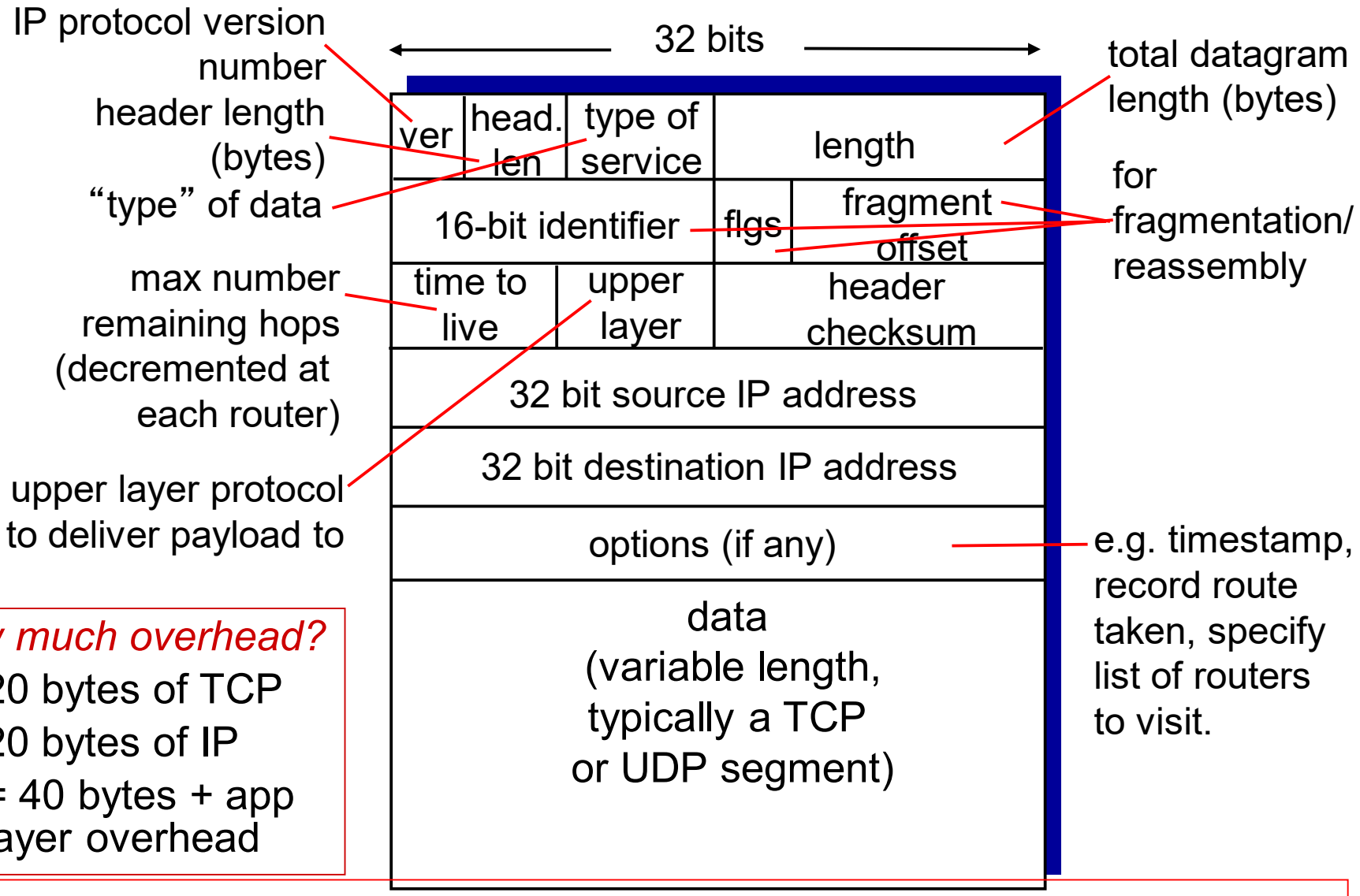
The Internet network layer

host, router network layer functions:



Three main components: IP protocol, Routing protocol, other supporting protocol.

IP datagram format (IPV4)

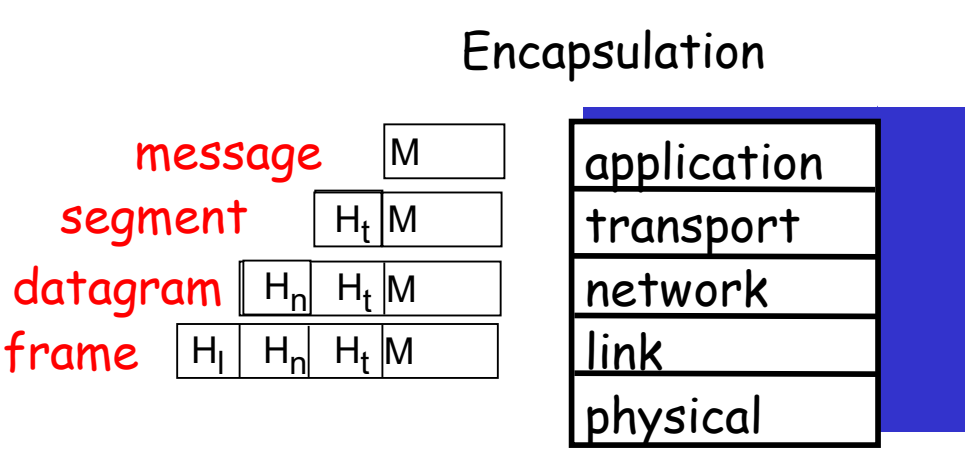


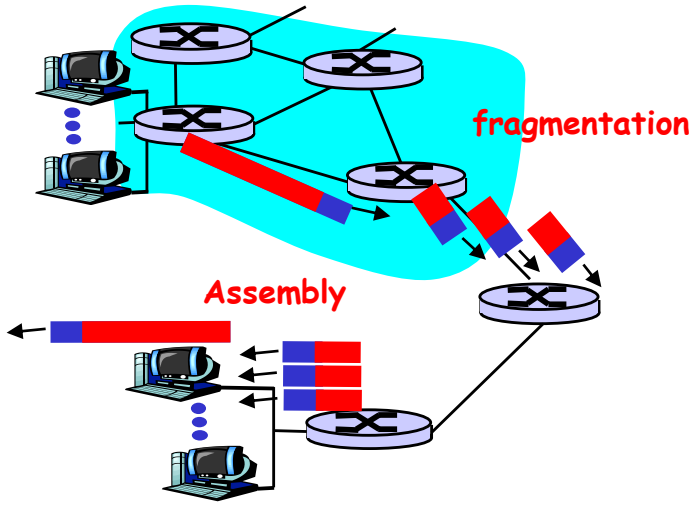
how much overhead?

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

Size of IP datagram: IP header (20 bytes + optional field) +payload data

IP Datagram and Link-layer Frame

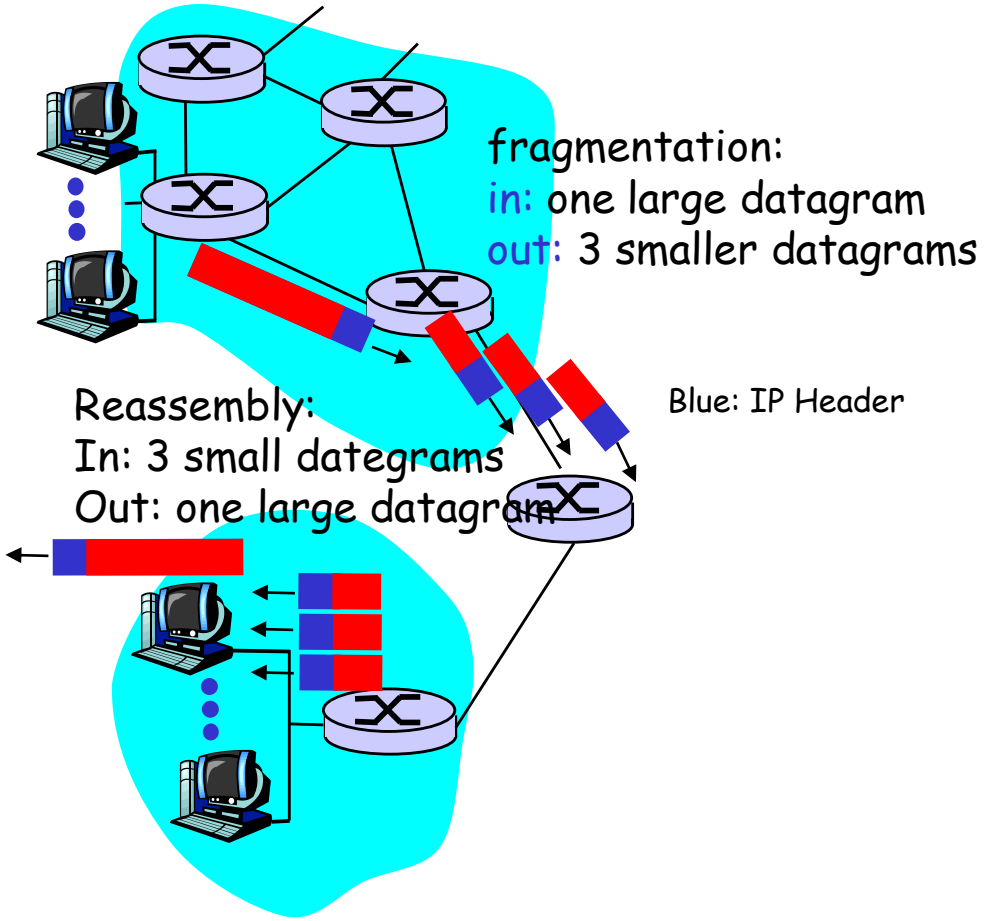




- IP datagram is encapsulated within the link-layer frame for transmission from one router to the next router
- MTU (Maximum Transmission Unit): MTU is the size of the largest possible transmission unit. Different link types may have different MTUs
  - MTU of Ethernet is 1500 byte: Ethernet frames can carry up to 1500 bytes for each frame.
  - Some wide-area links have MTU=576 bytes
- What happens if each of the links along the path between the sender to the destination? ---**Fragmentation and Reassembly**

IP Fragmentation & Reassembly

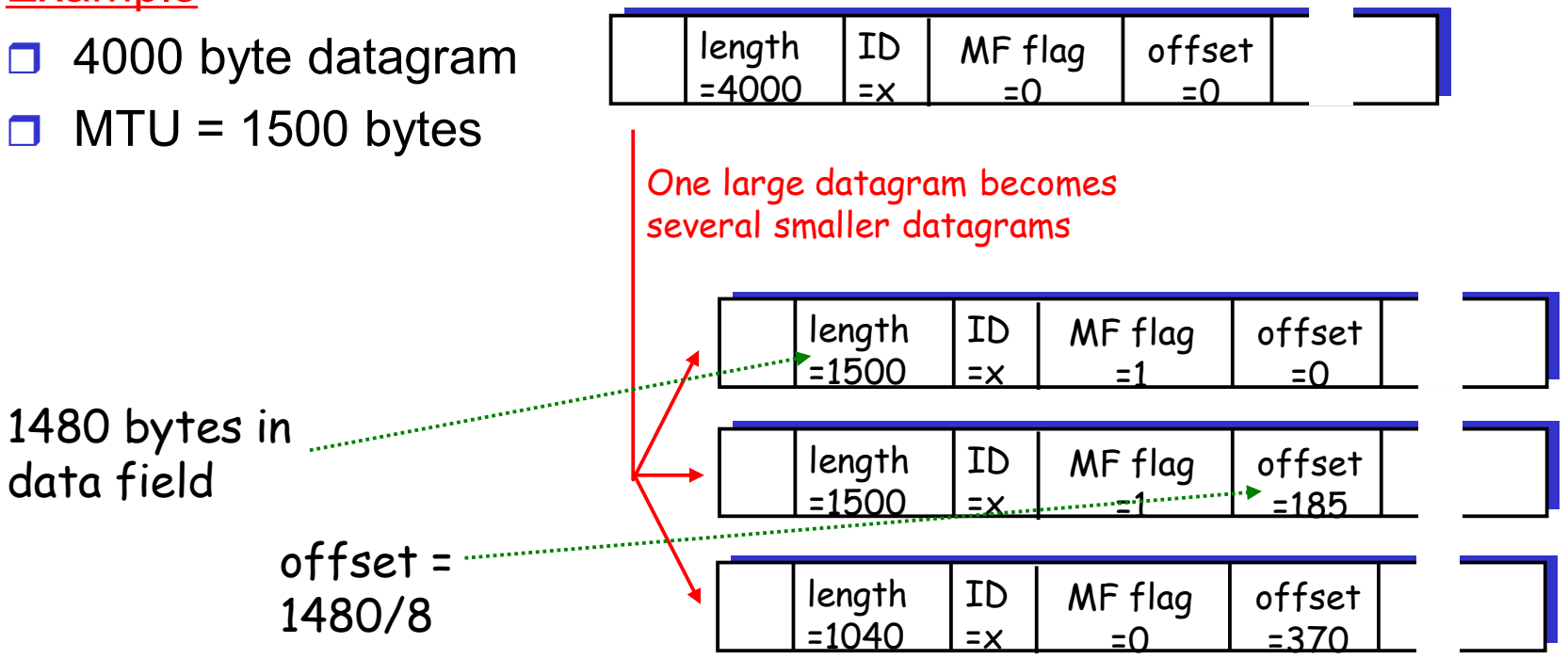
- Fragmentation:
  - Divide a large IP datagram into two or more smaller IP datagrames;
  - Encapsulate each of these smaller IP datagrames in a separate link-layer frames;
  - Send these frames over the outgoing link.
  - The process is called as fragmentation, and each of these smaller datagrams is referred to as a fragment.
- Reassembly: when these fragments arrive at their destination, the destination reassembles these fragments to reconstruct the original larger size datagram.
  - Length, Identification, flags, and fragmentation offset fields in the IP header are used to perform the fragment and reassembly task



IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes



Steps:

1. Subtract 20 from original length: 4000 -20 = 3980 (bytes of "IP payload")
2. Subtract 20 from new MTU: 1500- 20 = 1480 (max. bytes of data in each fragment)
3. Divide "maximum data bytes" by 8 **to get offset increment**: 1480/8 = 185
4. Offset of each fragment "n" (n = 1, 2, 3, ...) = (n-1) x "offset increment": 0, 185, 370. ...
5. Length of each fragment (except the last fragment) = MTU = 1500 bytes.
   Length of last fragment = 20 + remaining data bytes = 20 + 3980 - 2 x 1480 = 1040.

Note that fragment offset is in unit of 8-byte

IP Fragmentation & Reassembly

Advantages of fragmentation

- Fragmentation plays an important role in gluing together the many different link-layer technologies

Disadvantages of fragmentation

- Fragmentation complicates routers and end systems.
- Security issue: the attacker sends a stream of small fragments to the target host, none of which has an offset. The target can collapse as it attempts to rebuild datagrams from these packets.

IP version 6 doesn't support fragmentation

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

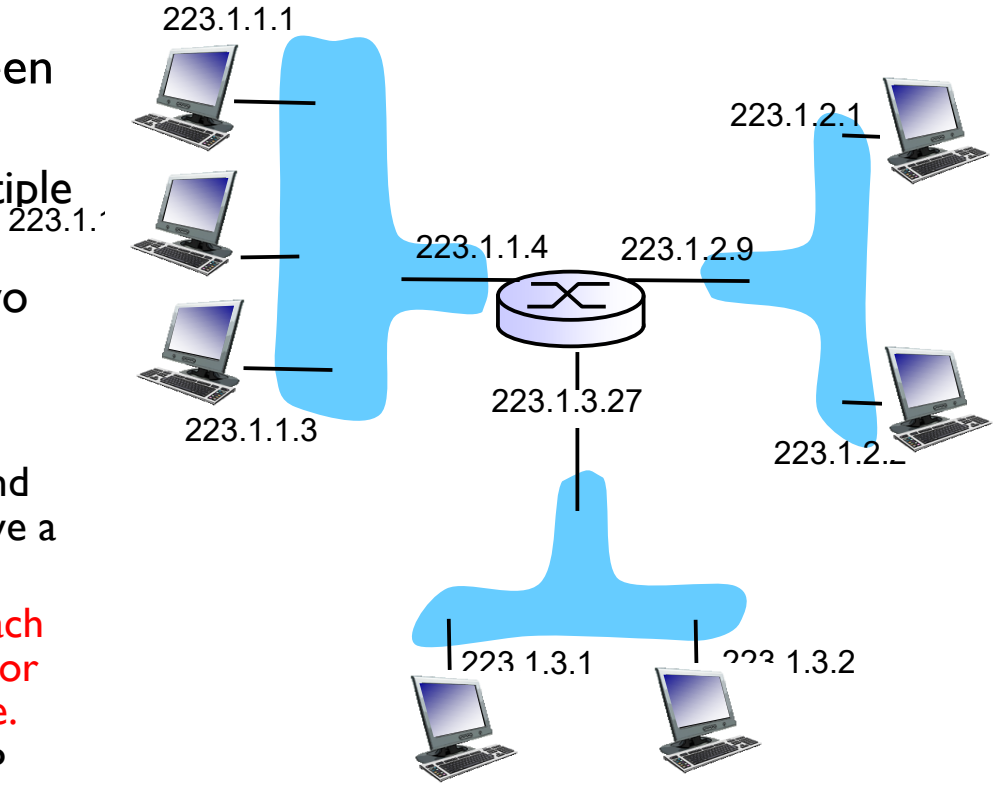
4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

IP addressing: introduction

- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces
- *IP Address*
  - Each interface on every host and router in the Internet must have a globally unique IP address.
  - IP address is associated with each interface, rather than the host or router containing that interface.
  - A router usually has multiple IP addresses, each of IP addresses corresponds to each of its interfaces
  - A host usually has one interface, therefore, we can say the IP address of a host.
  - 4-byte IP address is written in dotted-decimal notation.

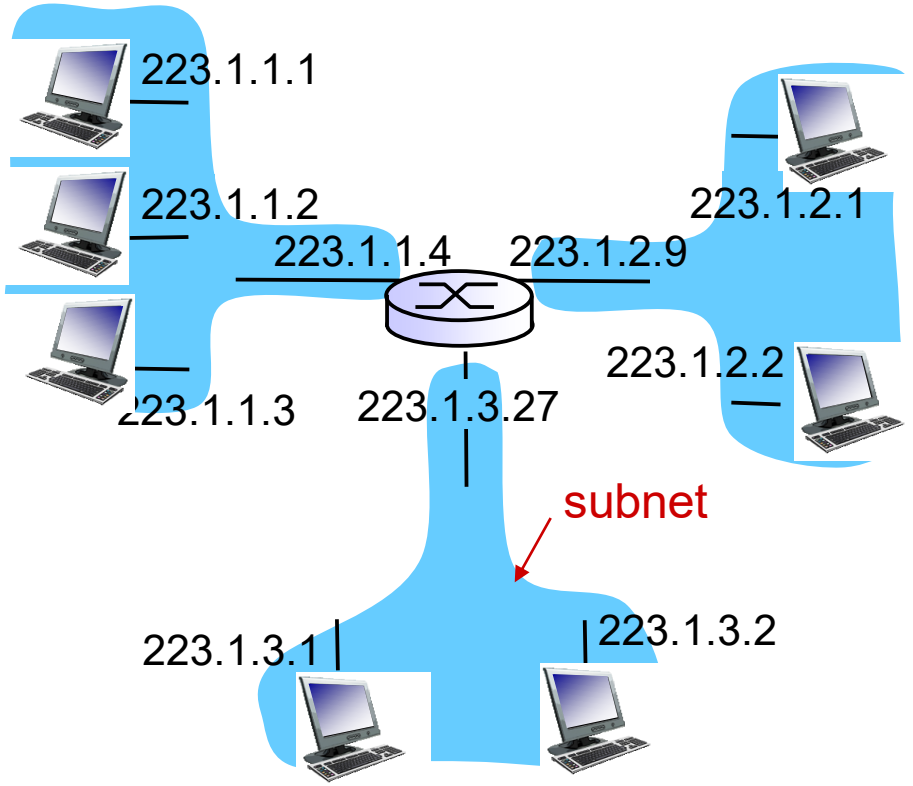


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Subnets

- The structure of IP address:
  - subnet part - high order bits
  - host part - low order bits
- *what's a subnet ?*
  - A subnet is a network where interfaces can physically reach each other *without intervening router*



network consisting of 3 subnets

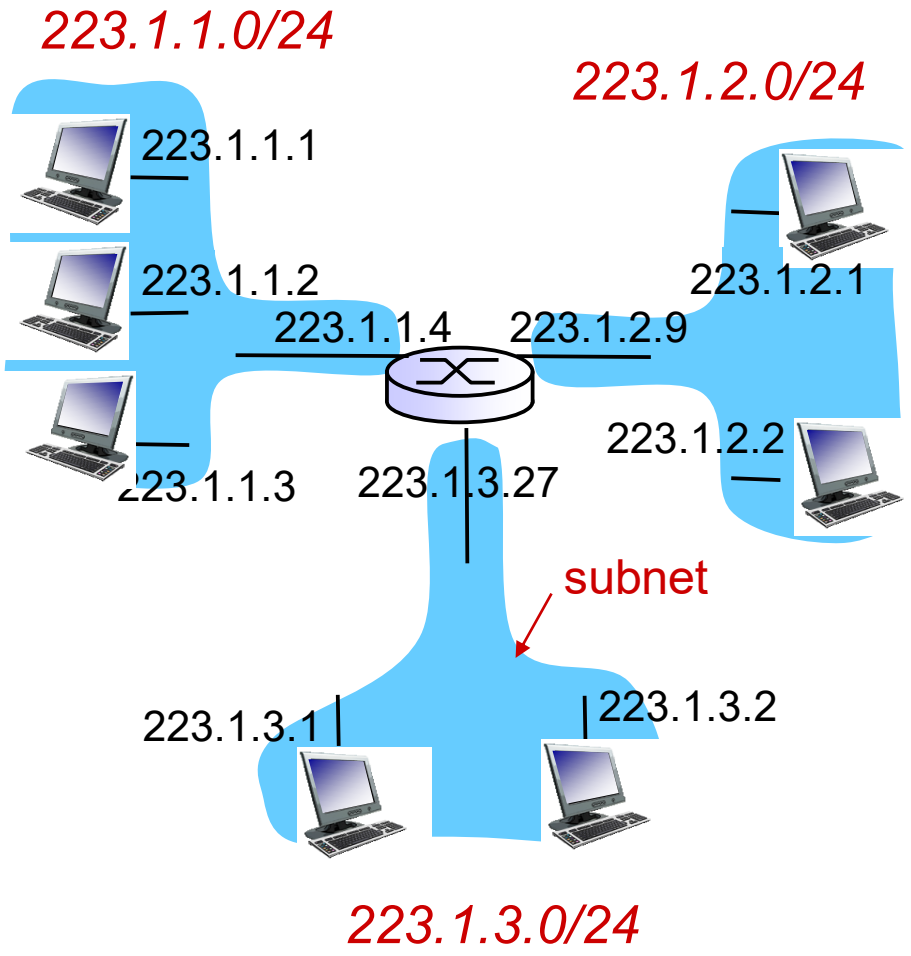
IP address of subnet

- Dotted-decimal notation: a.b.c.d/x, where the notation /x is known as **subnet mask**, it indicate that x leftmost bits of the 32-bit IP address is the subnet part, and remaining (32-x) bits is the host part
- Interfaces in the same subnet have the same subnet part of IP address
- Example: the above network includes three subnets with subnet mask /24.
- The subnet mask is used by the TCP/IP protocol to determine whether a host is on the local subnet or on a remote network.

Subnets

recipe

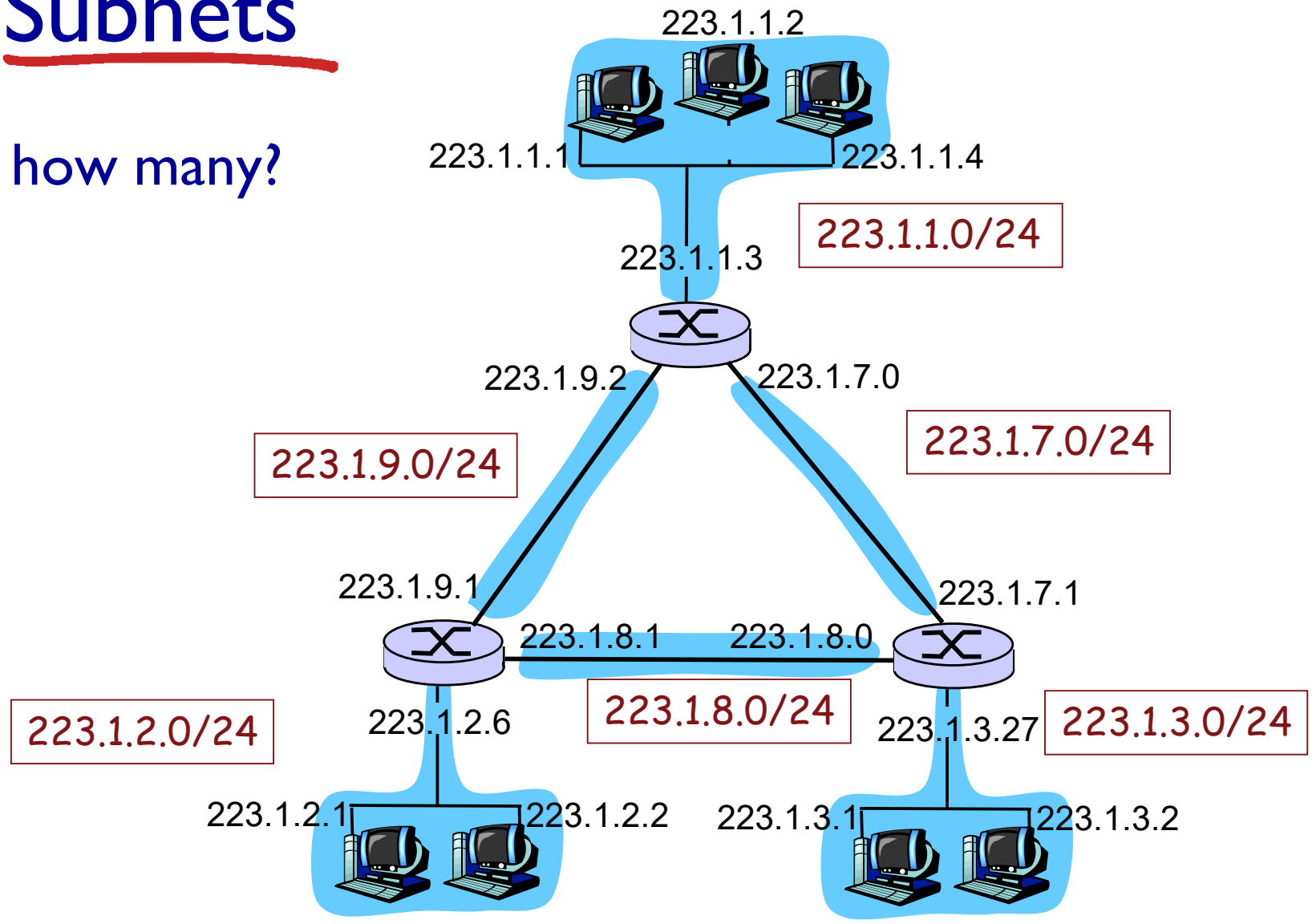
- to determine the subnets, detach each interface from its host or router, creating islands of isolated networks
- each isolated network is called a *subnet*



subnet mask: /24

Subnets

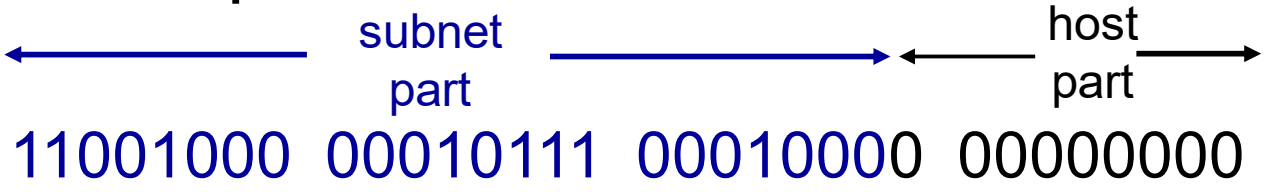
how many?



IP addressing: CIDR

CIDR: The Internet's address assignment strategy is known as Classless Interdomain Routing (CIDR—pronounced cider)

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



200.23.16.0/23

For a general interconnected system of routers and hosts, we can use the following recipe to define the subnets in the system:

- *IP address of subnet*
  - When the node part of the IP address is set to all 0s, it is used to identify the network address of a subnet.
  - For example, the subnet mask of the left subnet is /24; and its network address is 223.1.1.0
  - When the node part of the IP address is set to all 1s, it is a muticast address. That is, this packet is sent to all nodes in this subnet.
  - For example, the packet with the destination address 223.1.1.255 will be send to all nodes in the subnet 223.1.1.0 with network mask /24.

IP addresses: how to get one?

Q: How does a *host* get IP address?

- hard-coded by system admin in a file
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - "plug-and-play"
- *goal:* allow host to *dynamically* obtain its IP address from network server when it joins network

DHCP: Dynamic Host Configuration Protocol

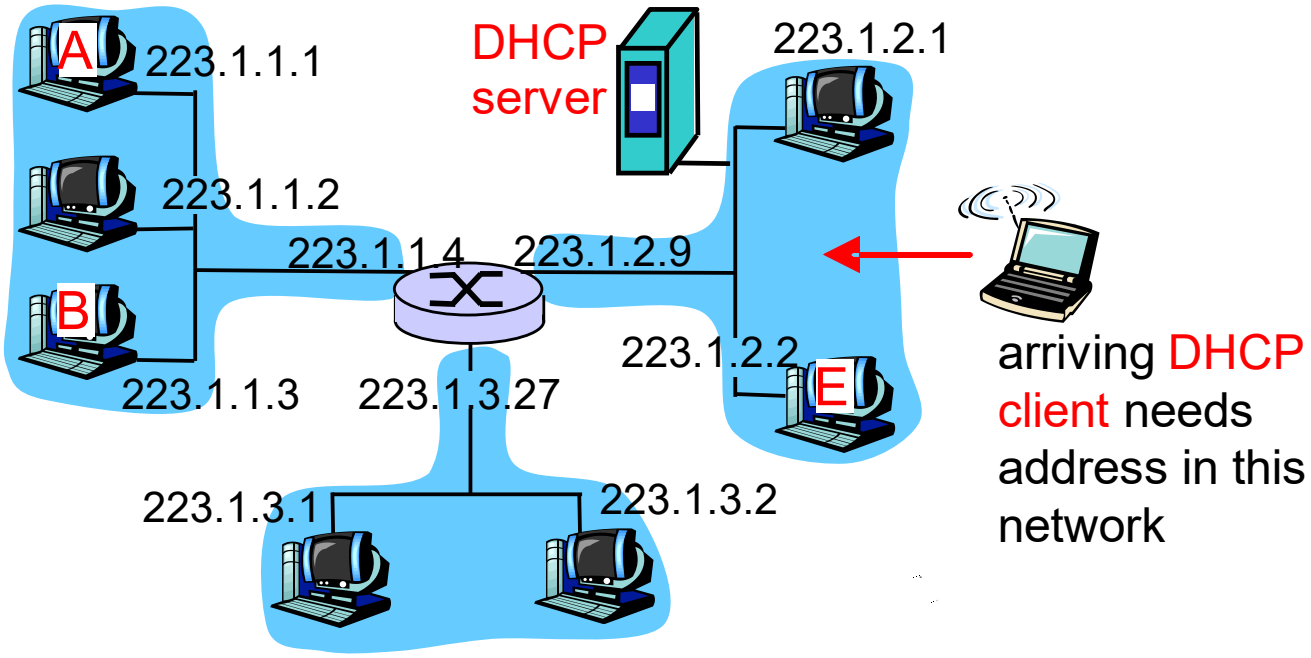
Goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- Can renew its lease on address in use
- Allows reuse of addresses (only hold address while connected an "on")
- Support for mobile users who want to join network (more shortly)

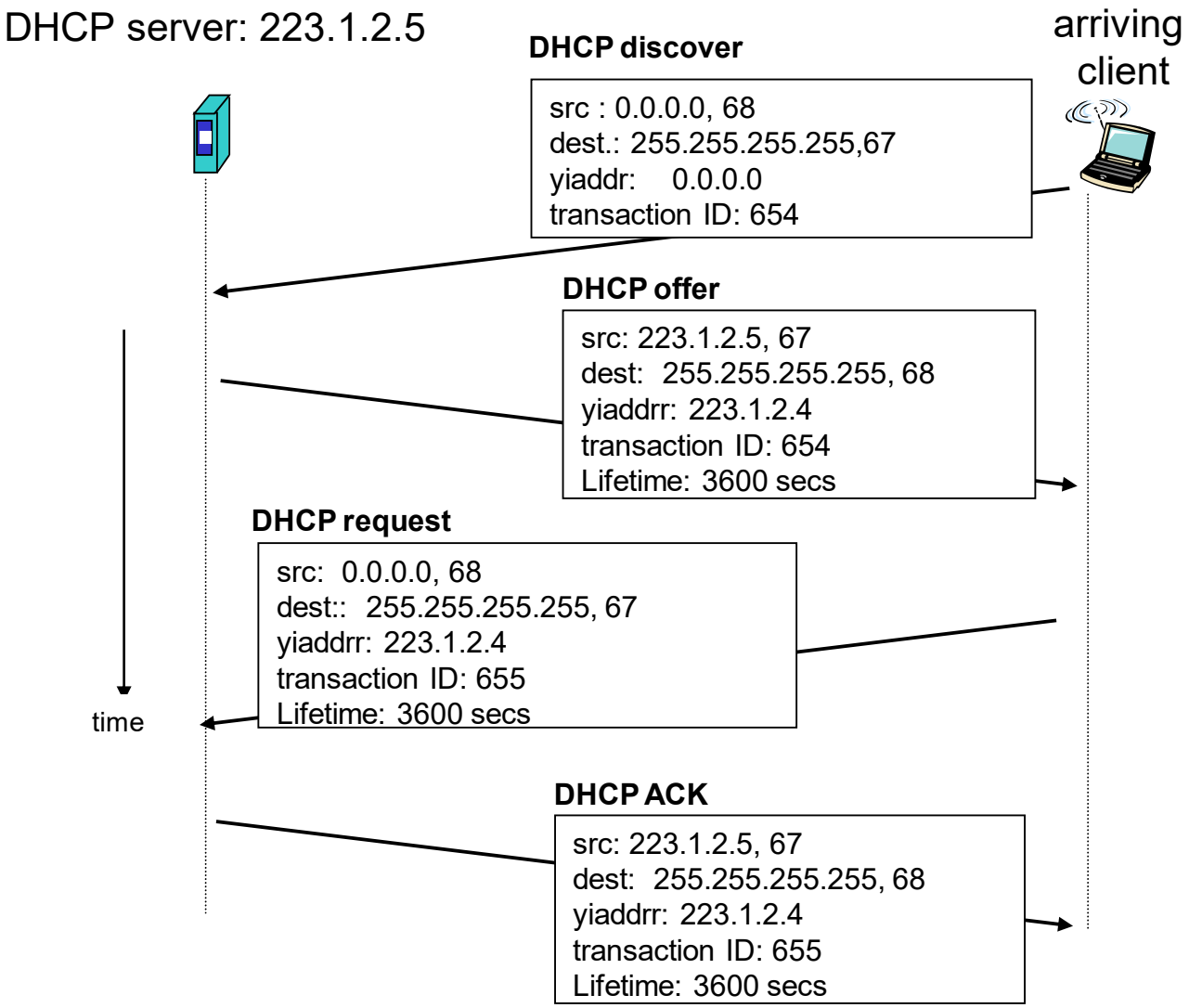
DHCP overview:

- host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario



DHCP client-server scenario

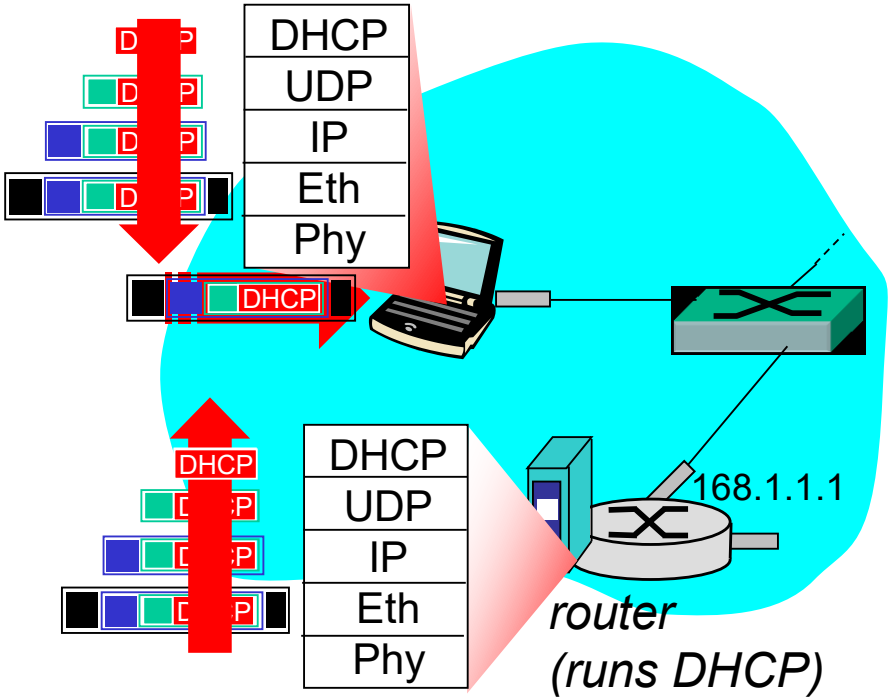


DHCP: more than IP address

DHCP can return more than just allocated IP address on subnet:

- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)

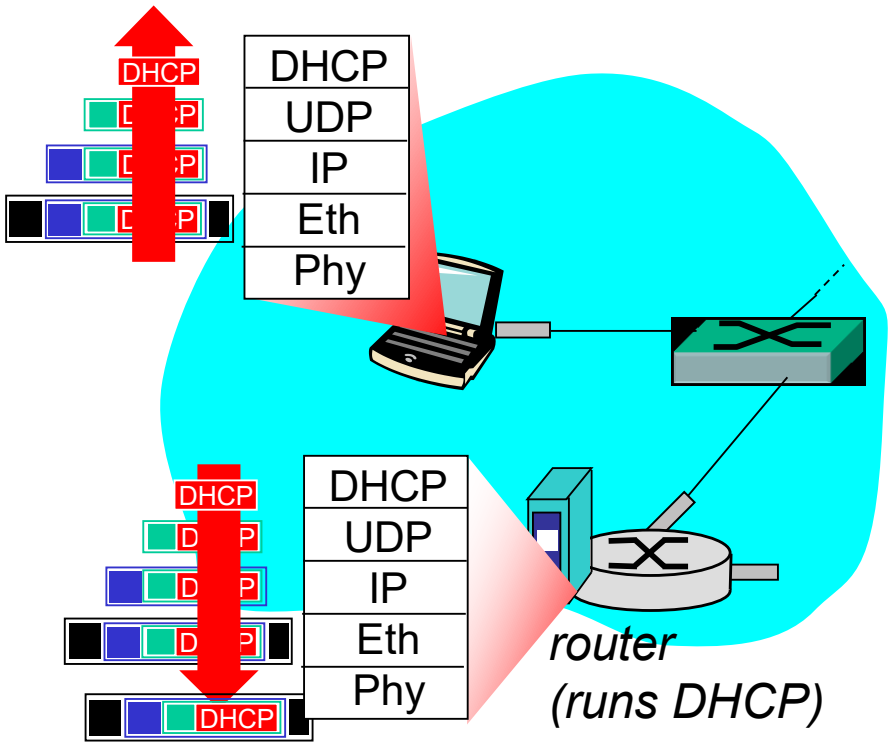
DHCP: example



- connecting laptop needs its IP address, addr of first-hop router, addr of DNS server: use DHCP

- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet demux'ed to IP demux'ed, UDP demux'ed to DHCP

DHCP: example



- DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server

- encapsulation of DHCP server, frame forwarded to client, demux'ing up to DHCP at client
- client now knows its IP address, name and IP address of DSN server, IP address of its first-hop router

DHCP: wireshark output (home LAN)

request

Message type: **Boot Request (1)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) **DHCP Message Type = DHCP Request**
Option: (61) Client identifier
    Length: 7; Value: 010016D323688A;
    Hardware type: Ethernet
    Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
    Length: 11; Value: 010F03062C2E2F1F21F92B
    1 = Subnet Mask; 15 = Domain Name
    3 = Router; 6 = Domain Name Server
    44 = NetBIOS over TCP/IP Name Server
    ……

reply

Message type: **Boot Reply (2)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
    Length: 12; Value: 445747E2445749F244574092;
    IP Address: 68.87.71.226;
    IP Address: 68.87.73.242;
    IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."

IP addresses: how to get one?

Q: How does an ISP (or organization) get a block of addresses?

- The nonprofit organization ICANN (the Internet Corporation for Assigned Names and Numbers) manages the IP address space and root DNS servers. It allocates the blocks of IP addresses to regional Internet registries (e.g., CNNIC (China Network Information Center), Network Solutions, CDNCC (Canadian Domain name Consultative Committee), etc.)
- These regional Internet registries allocate IP addresses to ISPs and organizations in their regions.
- Once an organization has obtained a block of addresses, it can assign individual IP address to the host and router interfaces in its organization usually using the DHCP (Dynamic Host Configuration Protocol)

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP: ICMP message are carried as IP payload (like UDP or TCP segment).
- ICMP message: type, code plus first 8 bytes of IP datagram causing error
- a node recognizing a transmission problem (TTL exceed, destination unreachable, etc.) generates ICMP messages

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

IPv6: motivation

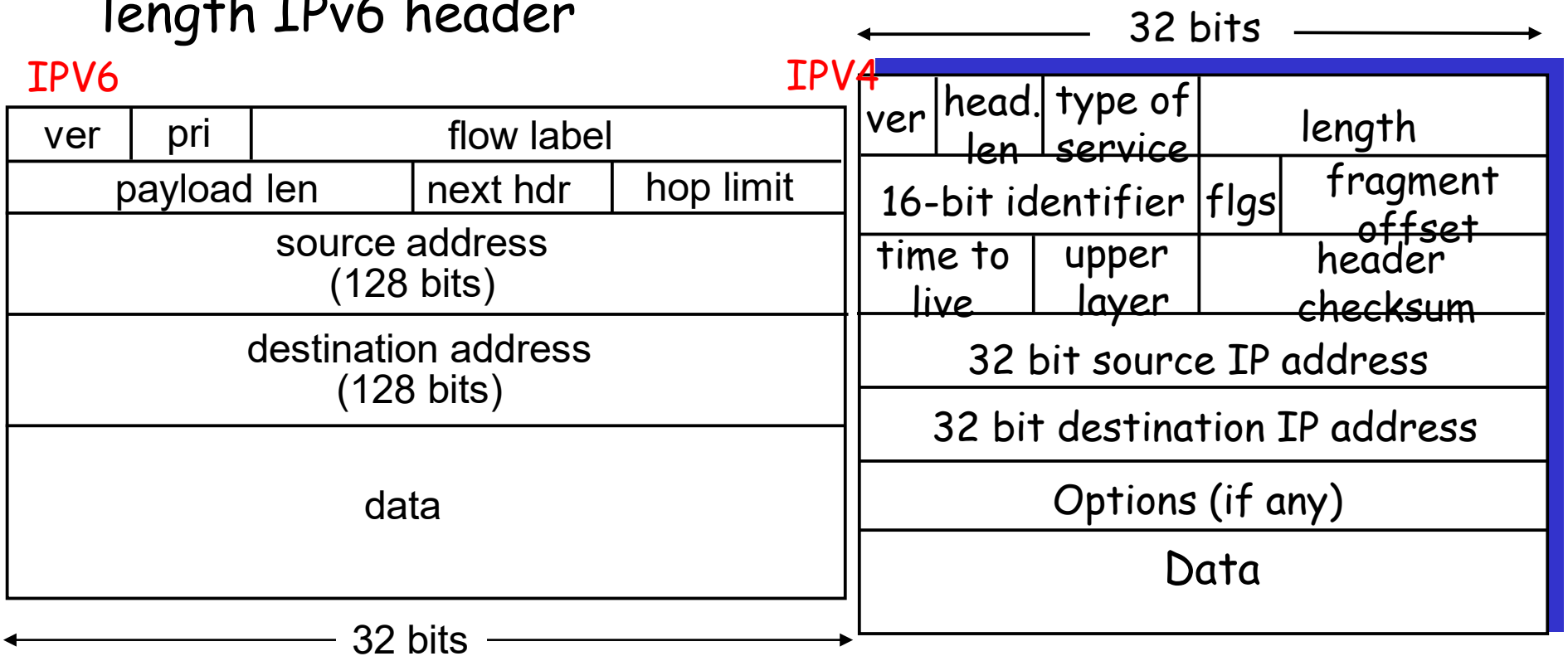
- *initial motivation:* 32-bit address space soon to be completely allocated.
  - IPv6 increases the size of the IP address from 32 to 128 bits.
- additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS

IPv6 datagram format:

- fixed-length 40 byte header
- no fragmentation allowed

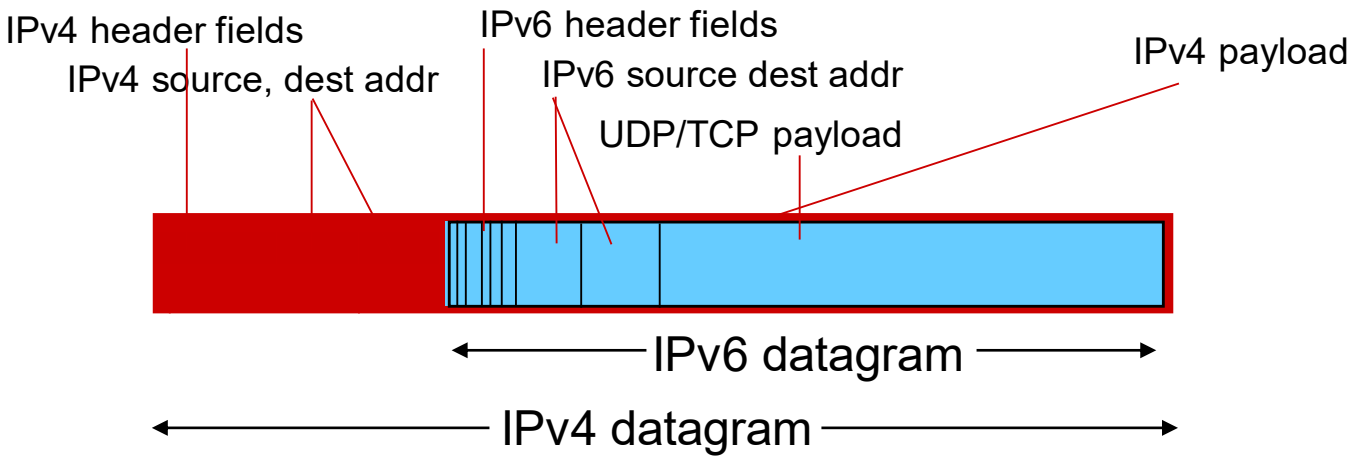
IPv6 Header

- Large address space: 32 bits to 128 bits
- No fragmentation in IPv6
- No checksum field in order to reduce processing time at each hop
- No options field in IPv6, which leads to 40-byte fixed length IPv6 header

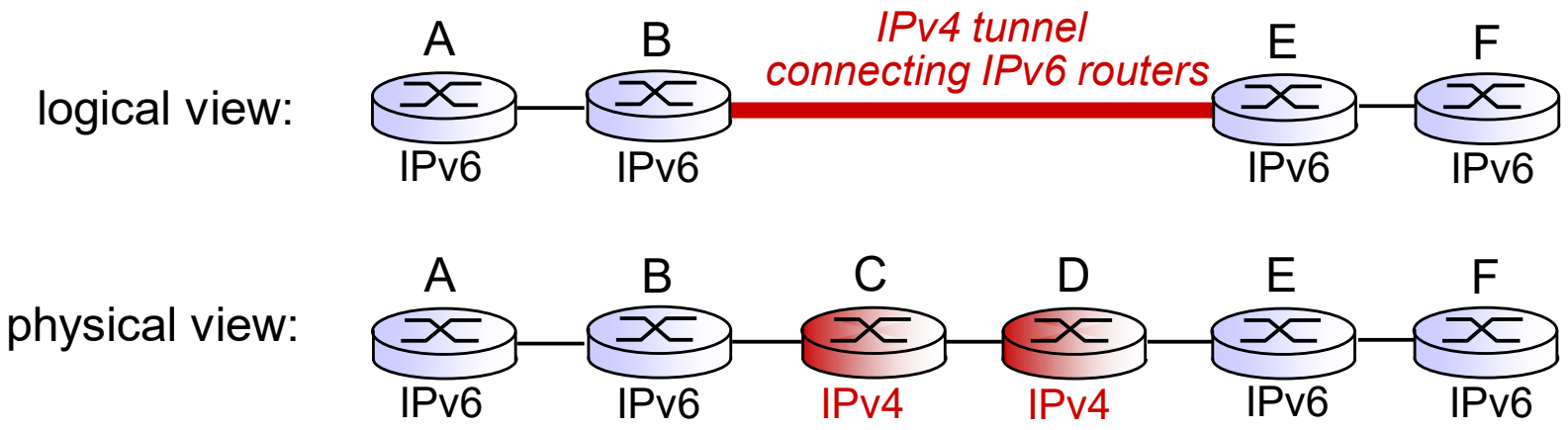


Transition from IPv4 to IPv6

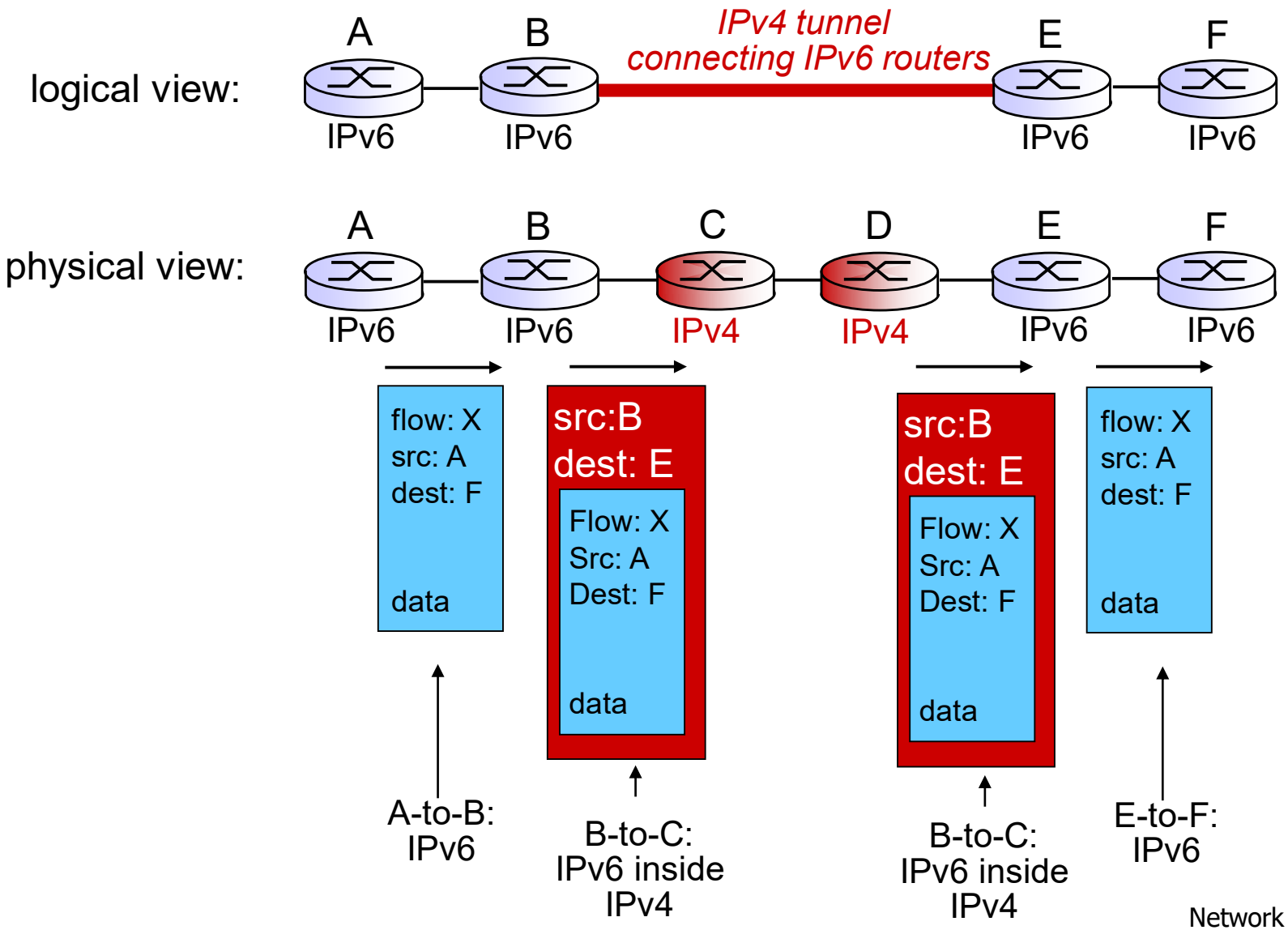
- not all routers can be upgraded simultaneously
  - no "flag days"
  - how will network operate with mixed IPv4 and IPv6 routers?
- *tunneling:* IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers



Tunneling


logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

Tunneling


logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
flow: X
src: A
dest: F
data
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
E-to-F:
IPv6

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

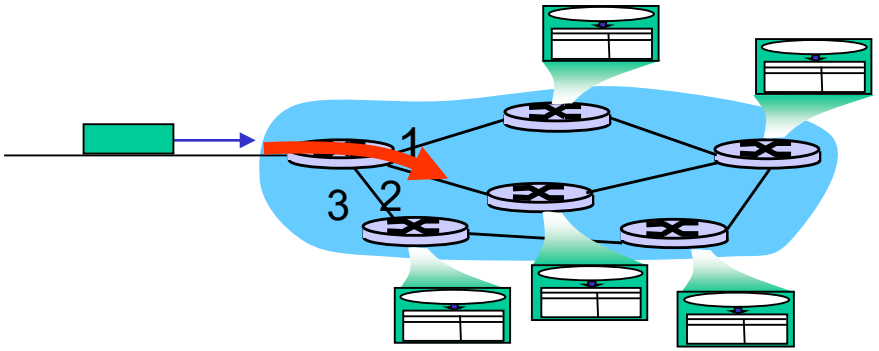
4.6 routing in the Internet

- RIP
- OSPF
- BGP

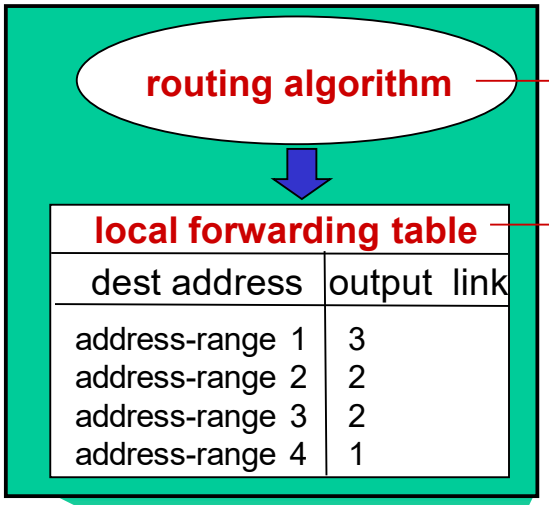
4.7 broadcast and multicast routing

Routing Algorithm

- The purpose of a routing algorithm: given a set of routers, with links connecting the routers, a router algorithm is to find a "good" path from source router and destination router.
- Routing algorithm operates in network routers to exchange and compute the information that is used to configure these forwarding tables.



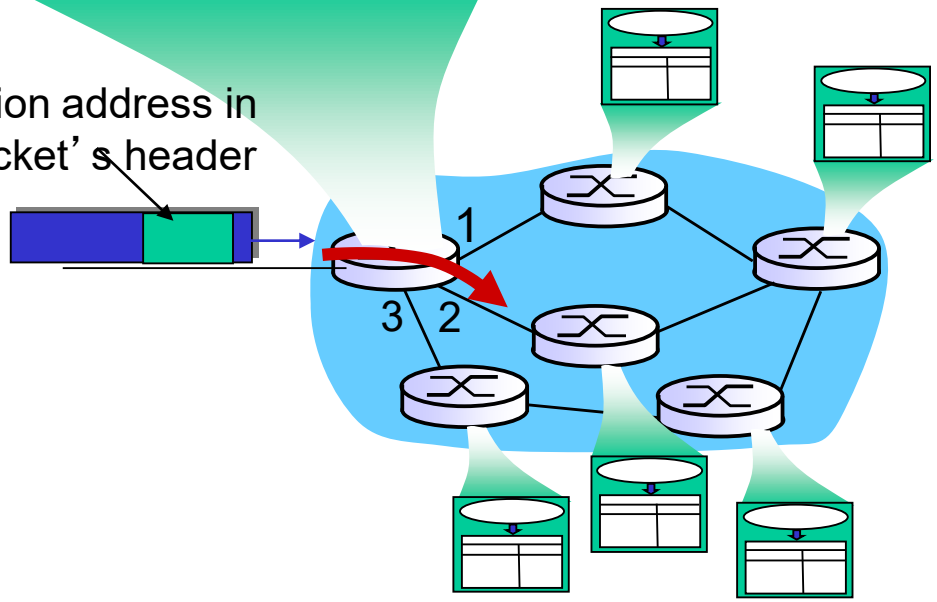
Interplay between routing, forwarding



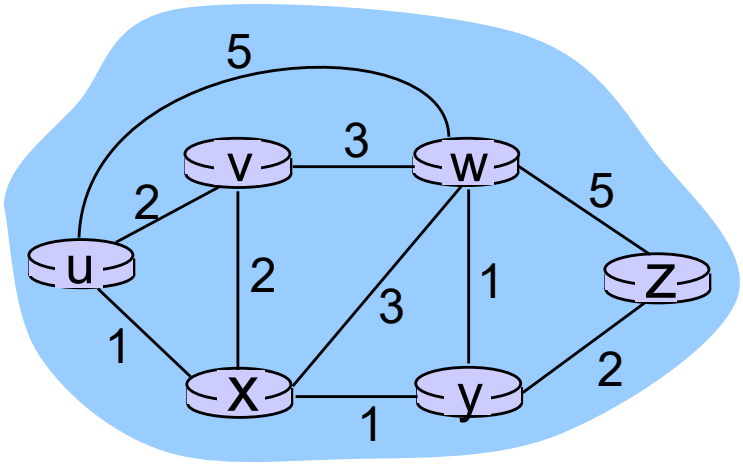
routing algorithm determines end-end-path through network

forwarding table determines local forwarding at this router

IP destination address in arriving packet's header



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) }

Path: a path in a graph G=(N, E) is s sequence of nodes $(x_1,x_2,\ldots,x_p)$ such that each of the consecutive pairs $(x_1,x_2),(x_2,x_3),\ldots(x_{p-1},x_p)$ are edges in E.

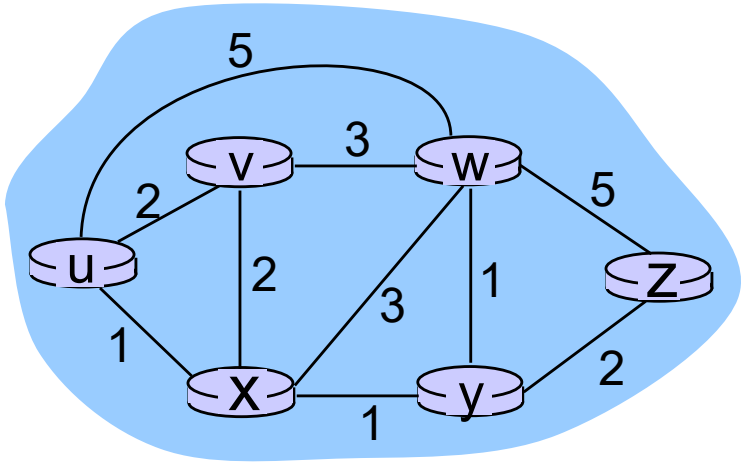
c(x,x') = cost of link (x,x')

e.g., c(w,z) = 5

- if (x, y) does not belong to E (i.e., there doesn't exist link between router x and router y), we set the cost c(x, y)=∞.
- We consider the undirected graphs (i.e., edges do not have a direction). Therefore c(x,y)=c(y,x)

cost of path $(x_1, x_2, x_3,\ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Graph abstraction: costs



Least-cost path: a path with the least cost.

Least-cost path problem: find the least-cost path between the source router and destination Router.

For example:

Shortest path: The path with the smallest number of links.

Shortest path problem: find the shortest path between the source router and destination router.

If all edges in the graph have the same cost, the least-cost problem is also the shortest path problem.

Routing algorithm classification

Q: 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

Q: static or dynamic?

static:

- routes change slowly over time

dynamic:

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

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

A Link-State Routing Algorithm

Dijkstra's algorithm

- net topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ('source") to all other nodes
  - gives *forwarding table* for that node
- iterative: after k iterations, know least cost path to k dest.'s

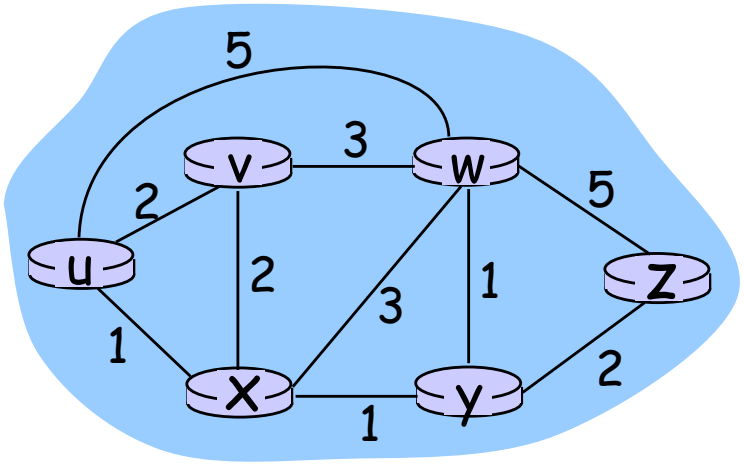
notation:

- c(x,y): link cost from node x to y; = ∞ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

Dijsktra's Algorithm

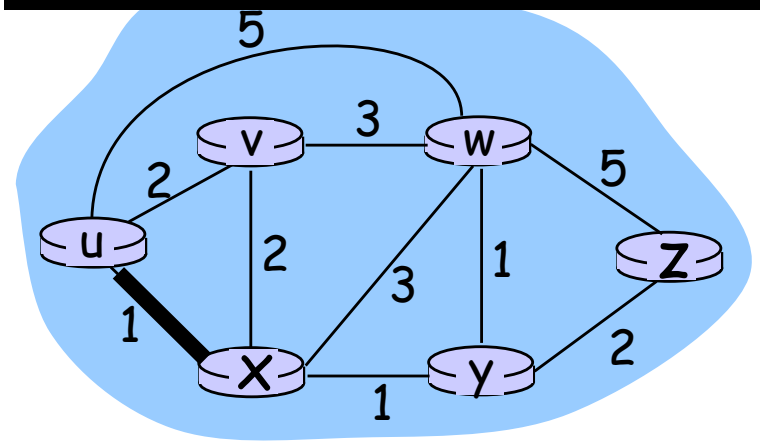
```
1  Initialization:
2    N' = {u}
3    for all nodes v
4      if v adjacent to u
5          then D(v) = c(u,v)
6      else D(v) = ∞
7
8  Loop
9     find w not in N' such that D(w) is a minimum
10    add w to N'
11    update D(v) for all v adjacent to w and not in N' :
12       D(v) = min( D(v), D(w) + 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: Find the least cost path from the node u to any other nodes.

Dijkstra's algorithm:



Initialization:

Add the source node into the set N': N'={u}

Initialize the cost other nodes: we classify these nodes into two types:

(a) For adjacent nodes of u: D(v)=2, D(x)=1, D(w)=5

(a) For all other nodes: D(y)=D(z)= ∞

The 1st iteration:

Compare all nodes without the set N' to find the node with the minimum cost D, that is x, then

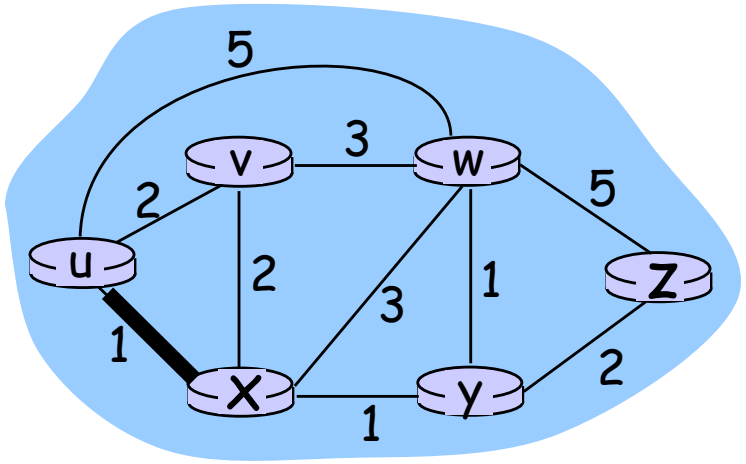
(1) add node x into the set N'. Therefore, N'={u,x}.

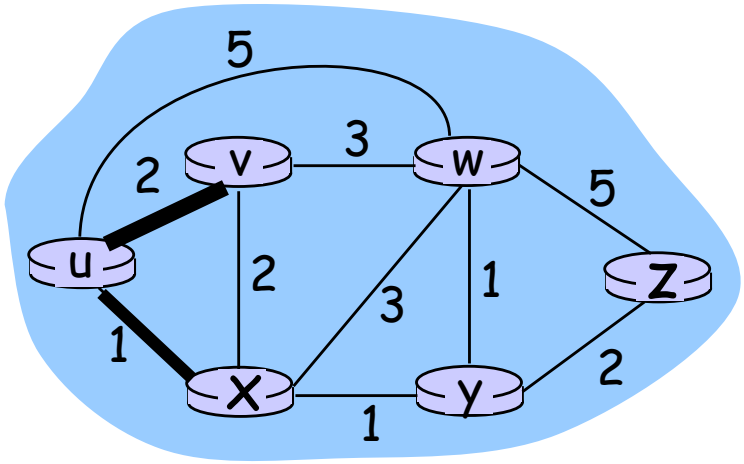
(2) the shortest path from source u to x is (u,x) with the cost D(x)=1

(3) The set of nodes without the set N' is $\overline{N}$'= {v,w,y,z}.

Update the cost for the nodes with $\overline{N}$'. We also classify these nodes into two types

(a) For nodes adjacent to the newly added node x. we use the formula D(i) = min(D(i), D(x) + c(x,i)) to update the cost





D(v)=min(D(v), D(x)+c(x,v))=min(2, 1+2)=2 through the path (u,v)

D(w)=min(D(w),D(x)+c(x,w))=min(5, 1+3)=4 through the path (u,x,w)

D(y)=min(D(y), D(x)+c(x,y))=min(∞, 1+1)= 2 through the path (u,x, y)

(a) For all other nodes without the set N' and not adjacent to newly added node x, we just copy their previous value.

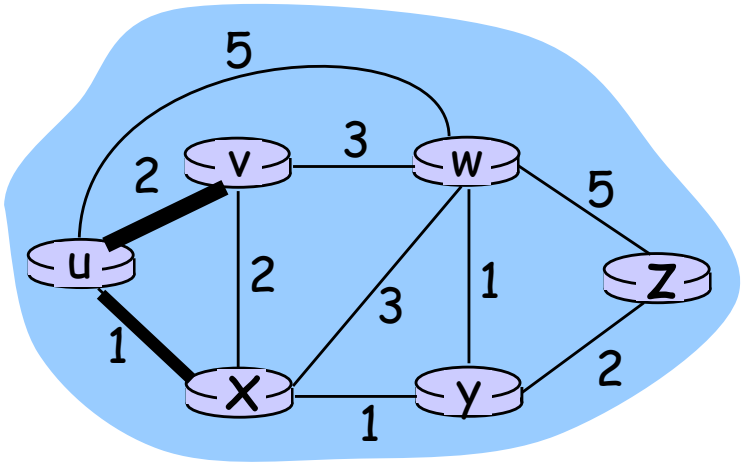
D(z) = ∞

The 2nd iteration:

Compare all nodes without the set N' to find the node with the minimum cost D, that is v (note that D(v)=D(y)=2 in this example, we just randomly choose node v), then

(1) add node v into the set N'. Therefore, N'={u,x,v}.

(2) the least cost from source u to v is D(v)= 2 through the path (u,v)



(3) The set of nodes without the set N' is $\overline{N'}$= {w,y,z}.

Update the cost for the nodes without the set N'. We also classify these nodes into two types based on the relation with the newly added node v:

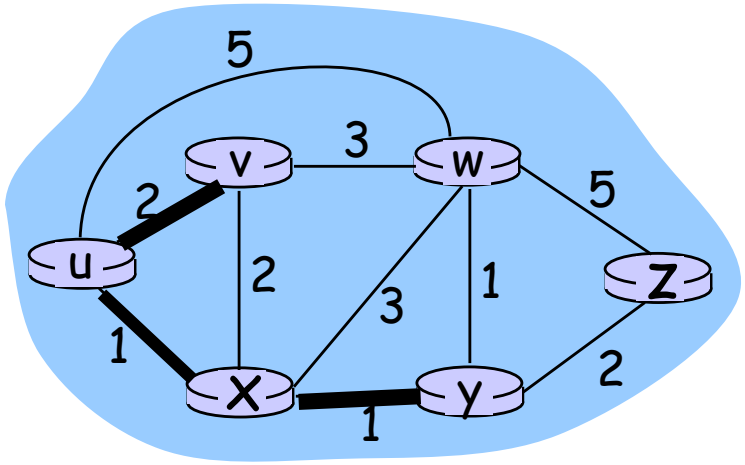
(a) For nodes adjacent to the newly added node v. we use the formula D(i) = min(D(i), D(v) + c(v,i)) to update their costs. We have

D(w)=min(D(w),D(v)+c(v,w))=min(4, 2+3)=4 through the path (u, x, w) ;

(b) For other nodes (y and z), we just copy their previous value.

D(y)=2 through the path (u, x, y).

D(z) = ∞



The 3rd iteration:

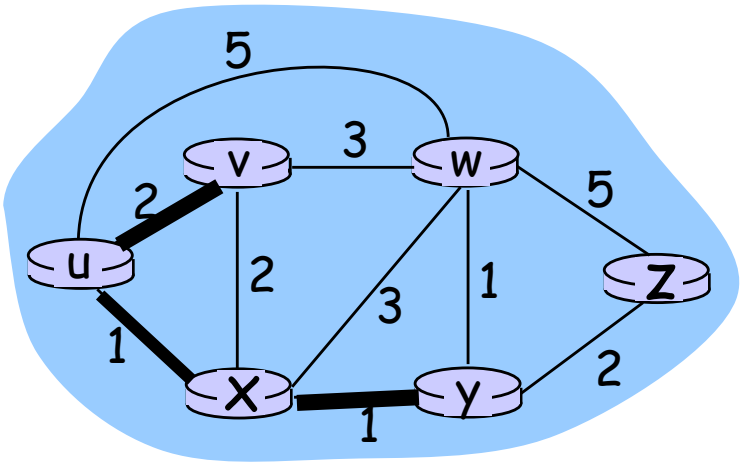
Compare all nodes without the set N' to find the node with the minimum cost D, that is y, then

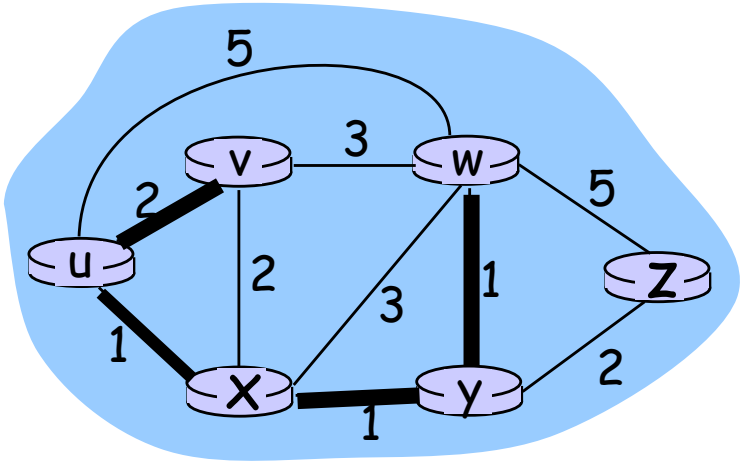
(1) add node y into the set N'. Therefore, N'={u,x,v,y}.

(2) the least cost from source u to y is D(y)= 2 through the path (u, x, y)

(3) The set of nodes without the set N' is $\overline{N'}$= {w,z}.

Update the cost for the nodes without the N'. We also classify these nodes into two types based on the relation with the newly added node y:





(a) For nodes adjacent to the newly added node y. we use the formula D(i) = min(D(i), D(y) + c(y,i)) to update their costs. We have

D(w)=min(D(w),D(y)+c(y,w))=min(4, 2+1)=3 through the path (u,x,y,w) .

D(z)=min(D(z), D(y)+c(y,z))=min(∞, 2+2)=4 through the path (u,x,y,z).

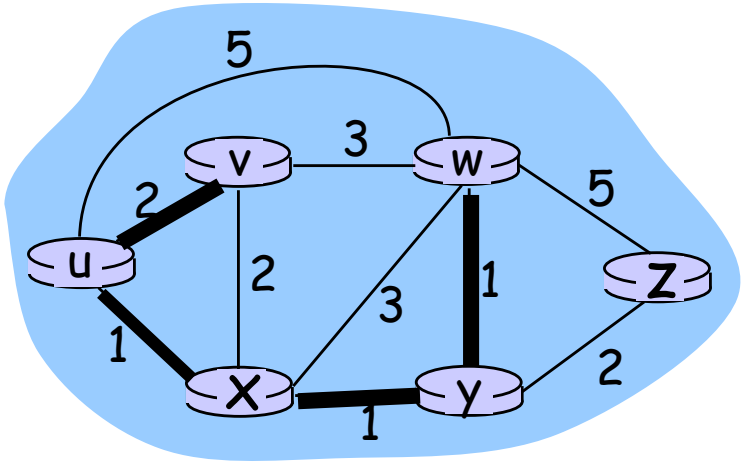
(b) Omit this step since all nodes are adjacent to y.

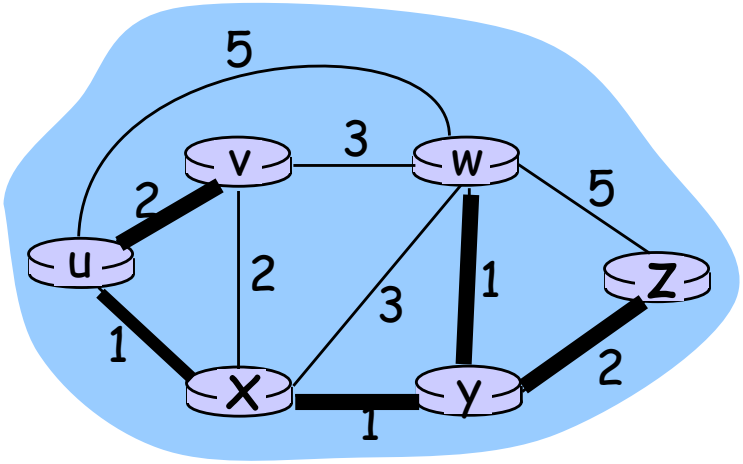
The 4th iteration:

Compare all nodes without the set N' to find the node with the minimum cost D, that is w, then

(1) add node w into the set N'. Therefore, N'={u,x,v,y,w}.

(2) the least cost from source u to w is D(w)= 3 through the path (u,x,y,w) .





(3) The set of nodes without the set N' is {z}.

Update the cost for the nodes without N'. We also classify these nodes into two types based on the relation with the newly added node w:

(a) For nodes adjacent to the newly added node w. we use the formula D(i) = min(D(i), D(w) + c(w,i)) to update their costs. We have

D(z)=min(D(z), D(w)+c(w,z))=min(4, 3+5)=4 through the path (u,x,y,z).

(b) omit this step since all nodes are adjacent to the node w.

The 5th iteration:

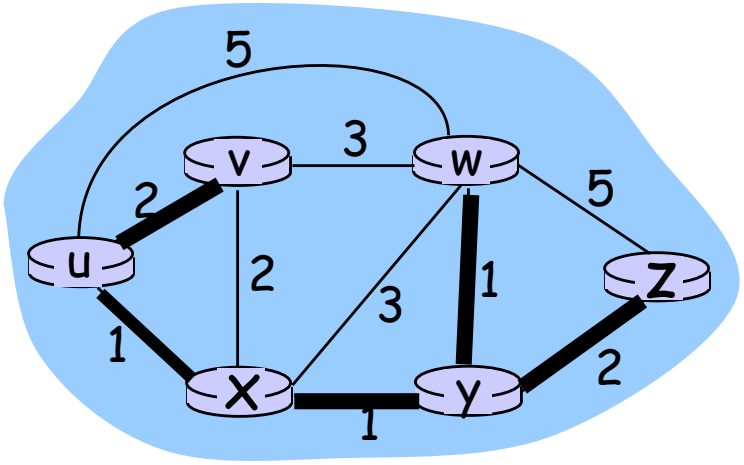
Compare all nodes without the set N' to find the node with the minimum cost D, that is z, then

(1) add node z into the set N'. Therefore, N'={u,x,v,y,w,z}.

(2) the least cost from source u to z is D(z)= 4 through the path (u,x,y,z).

(3) the set of nodes without the set N' is empty. Therefore, finish the algorithm.

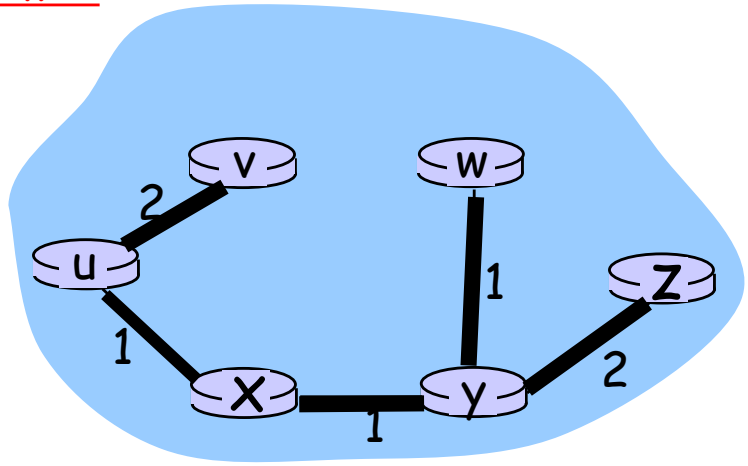
Dijkstra's algorithm



Finally, we find the least cost and least cost path from the node u to any other nodes, which is is called the "shortest-path tree", or "sink tree," for node u.

Dijkstra's algorithm

Resulting shortest-path tree from u:



Resulting forwarding table in u:

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Distance Vector (DV) Algorithm

DV algorithm is iterative, asynchronous, and distributed.

- Distributed: each node receives information from its directly attached neighbors, performs a calculation, and then distributes the results of this calculation back to its neighbors. To find Destination, source node S asks each neighbor X
- Asynchronous: it dons not require all of the nodes to operate simultaneously.
- Iterative: this process continues on until no more information is exchanged between neighbors.

Distance vector algorithm

Bellman-Ford equation (dynamic programming)

let

$d_x(y)$:= cost of least-cost path from x to y

then

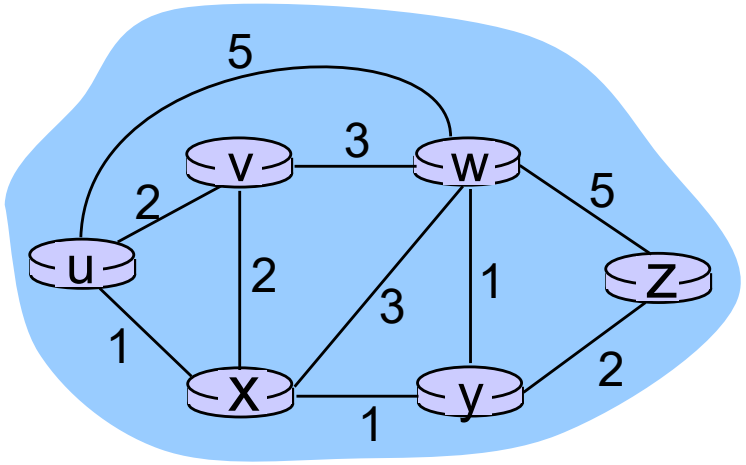
$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

cost from neighbor v to destination y

cost to neighbor v

min taken over all neighbors v of x

Bellman-Ford example



clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

B-F equation says:

$$d_u(z) = \min \{ c(u,v) + d_v(z), \; c(u,x) + d_x(z), \; c(u,w) + d_w(z) \}$$
$$= \min \{2 + 5, \; 1 + 3, \; 5 + 3\} = 4$$

node achieving minimum is next hop in shortest path, used in forwarding table

Distance vector algorithm

- $D_x(y)$ = estimate of least cost from x to y
  - x maintains distance vector $\mathbf{D}_x = [D_x(y): y \in N]$
- node x:
  - knows cost to each neighbor v: $c(x,v)$
  - maintains its neighbors' distance vectors. For each neighbor v, x maintains $\mathbf{D}_v = [D_v(y): y \in N]$

Distance vector algorithm

key idea:

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

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

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

Distance vector algorithm

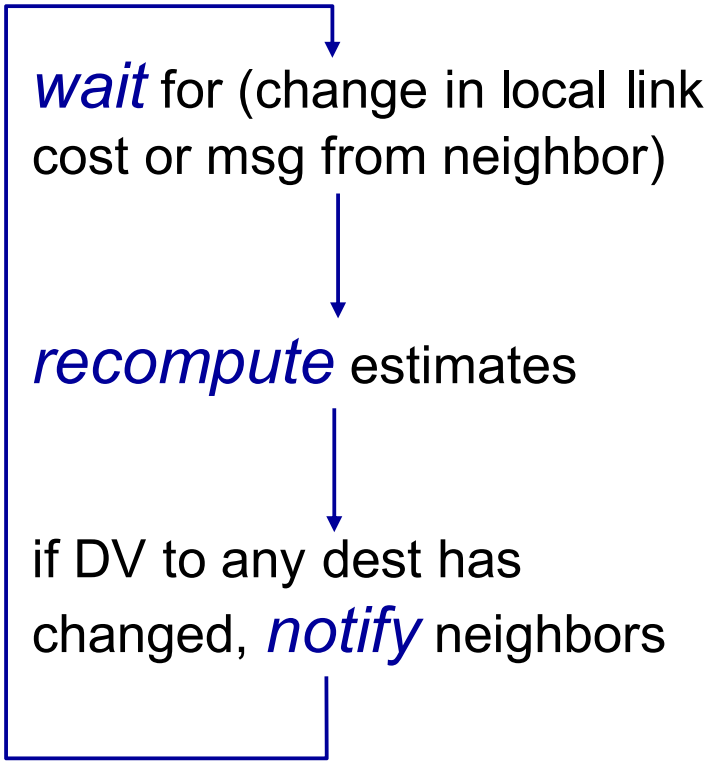
iterative, asynchronous:
each local iteration caused by:

- local link cost change
- DV update message from neighbor

distributed:

- each node notifies neighbors *only* when its DV changes
  - neighbors then notify their neighbors if necessary

each node:



$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$
$= \min\{2+0\ ,\ 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\}$
$= \min\{2+1\ ,\ 7+0\} = 3$

node x table

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

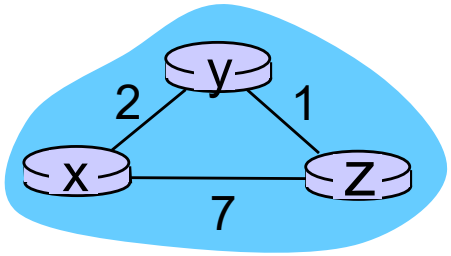
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

node y table

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

node z table

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



time

$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$
$= \min\{2+0 , 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$
$= \min\{2+1 , 7+0\} = 3$

node x table

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

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

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

node y table

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

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

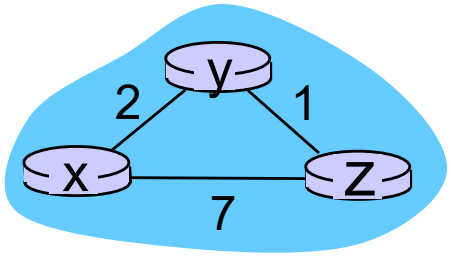
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node z table

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

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

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

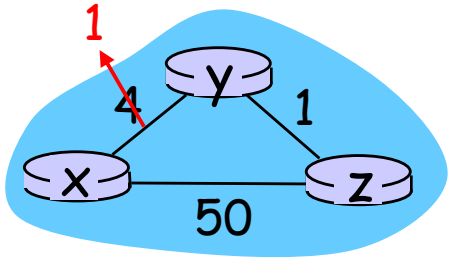


time

Distance vector: link cost changes

link cost changes:

- node detects local link cost change
- updates routing info, recalculates distance vector
- if DV changes, notify neighbors



"good news travels fast"

t_0 : *y* detects link-cost change, updates its DV, informs its neighbors.

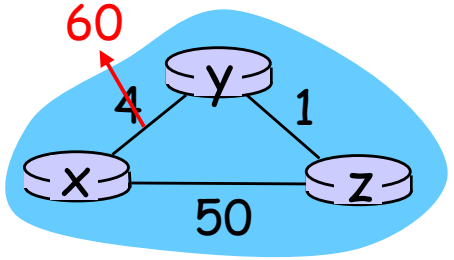
t_1 : *z* receives update from *y*, updates its table, computes new least cost to *x* , sends its neighbors its DV.

t_2 : *y* receives *z*'s update, updates its distance table. *y*'s least costs do *not* change, so *y* does *not* send a message to *z*.

Distance vector: link cost changes

link cost changes:

- node detects local link cost change
- *bad news travels slow* - "count to infinity" problem!
- 44 iterations before algorithm stabilizes: see text



poisoned reverse:

- If Z routes through Y to get to X :
  - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
- will this completely solve count to infinity problem?

Routing Table and Forwarding Table

In computer networking a ***routing table***, is a data table stored in a router or a networked computer that lists the routes to particular network destinations, and in some cases, metrics (distances) associated with those routes. The routing table contains information about the topology of the network immediately around it.

Routing tables are generally not used directly for packet forwarding in modern router architectures; instead, they are used to generate the information for a smaller forwarding table.

A forwarding table contains only the routes which are chosen by the routing algorithm as preferred routes for packet forwarding.

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Hierarchical routing : Motivation

our routing study thus far - idealization

- all routers identical
- network "flat"

… *not* true in practice

Two key reasons for the use of hierarchical routing in a large network such as Internet

scale: with 600 million destinations:

- can't store all destination's in routing tables!
- routing table exchange would swamp links!

administrative autonomy (AS)

- internet = network of networks
- each network administrator may want to control routing in its own network

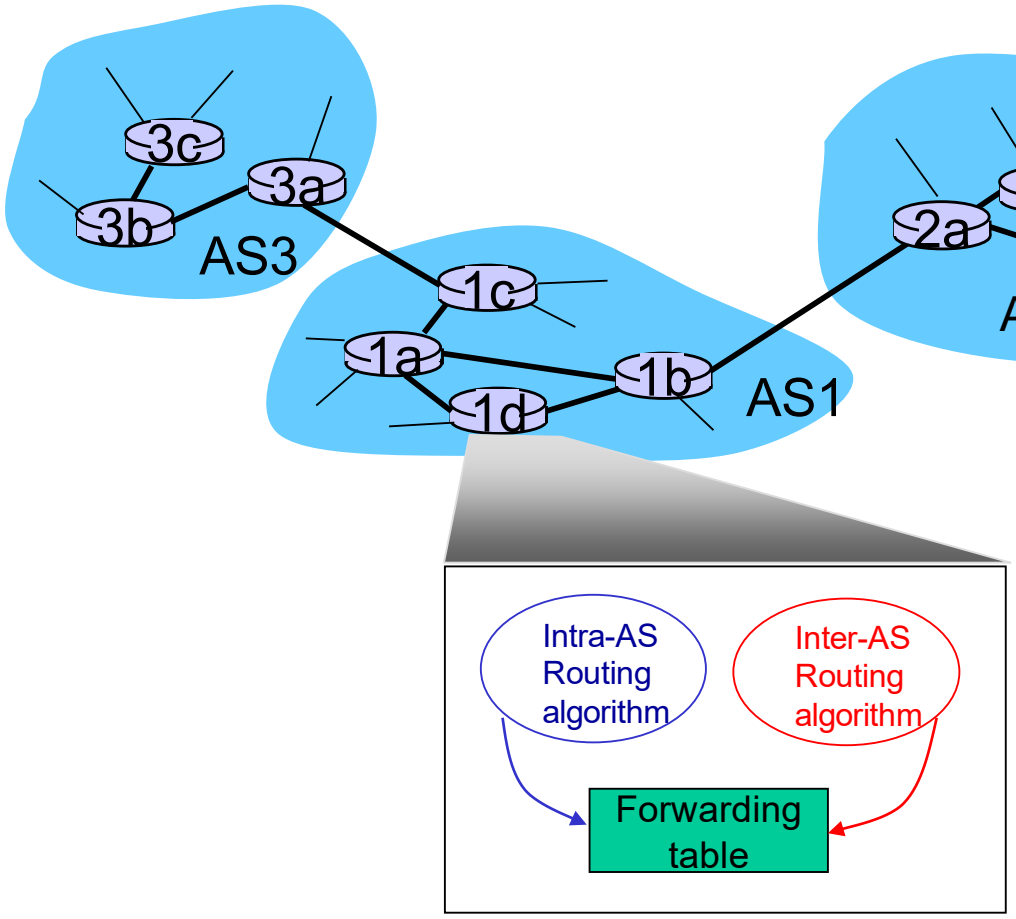
Hierarchical routing

- aggregate routers into regions, "autonomous systems" (AS)
  - Each AS consisting of a group of routers that are typically under the same administrative control.
- routers in same AS run same routing protocol
  - "intra-AS" routing protocol : determine the path within the AS.
  - routers in different AS can run different intra-AS routing protocol

gateway router:

- at "edge" of its own AS
- has link to router in another AS

Interconnected ASes



- forwarding table configured by both intra- and inter-AS routing algorithm
  - intra-AS sets entries for internal dests
  - inter-AS & intra-AS sets entries for external dests

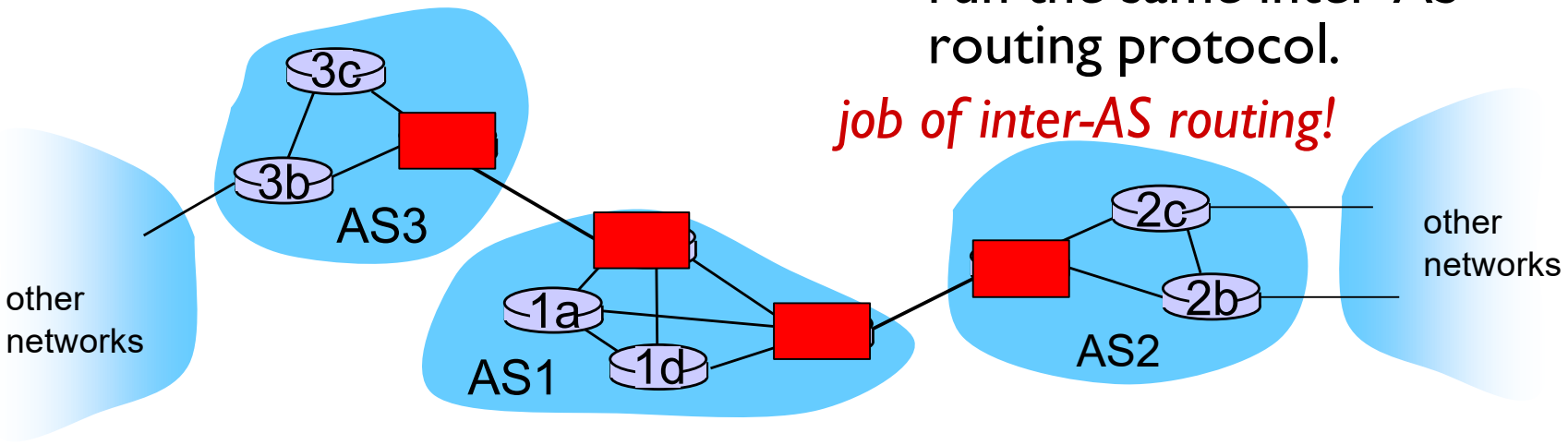
Inter-AS tasks

- suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router, but which one?

AS1 must:

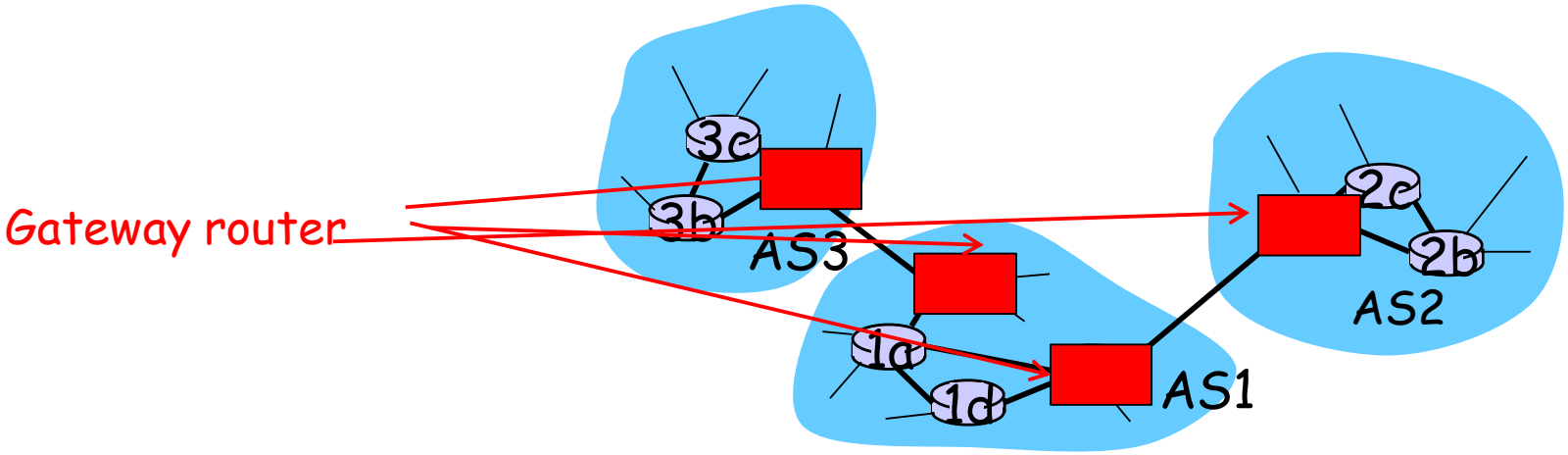
1. learn which dests are reachable through AS2, which through AS3
2. propagate this reachability info to all routers within the AS1
3. Neighboring ASs should run the same inter-AS routing protocol.

job of inter-AS routing!

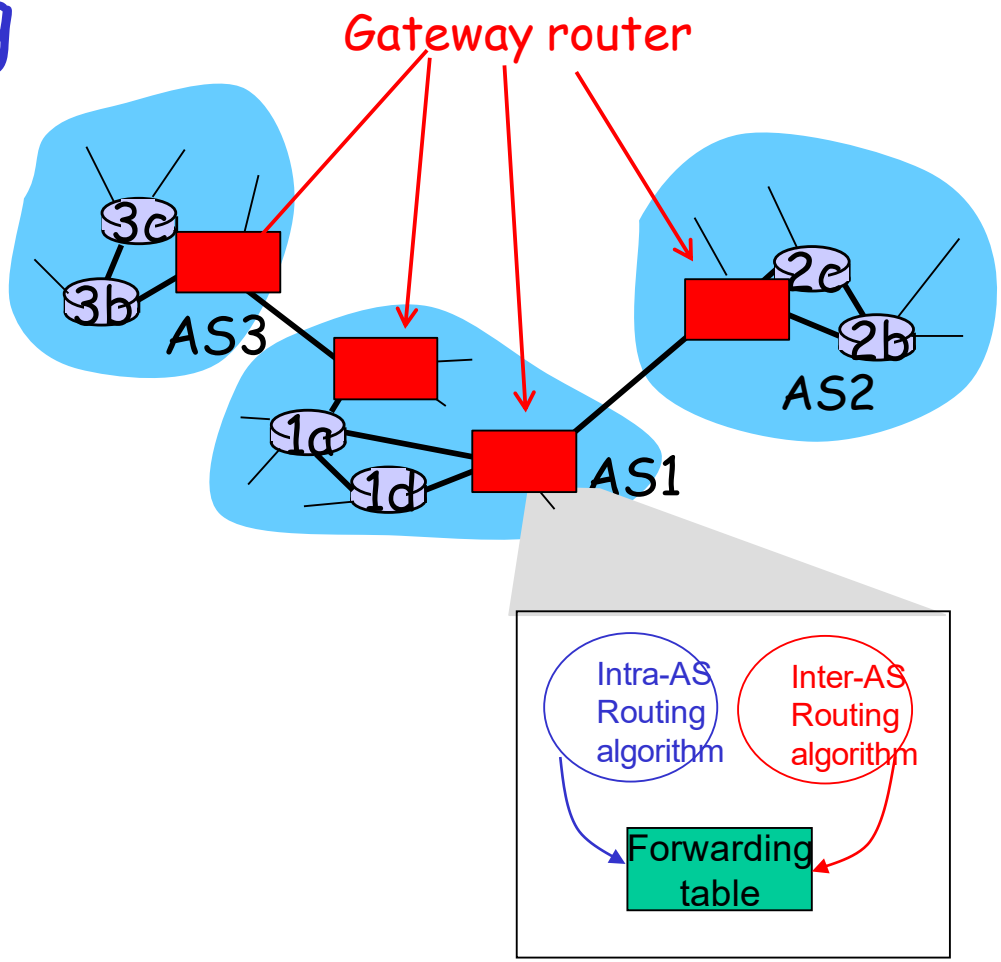


Hierarchical Routing

- Gateway router
  - In each AS, one or several routers take responsibility for forwarding packets to destination outside the AS, which is called Gateway router.
  - It provides direct link to router in another AS



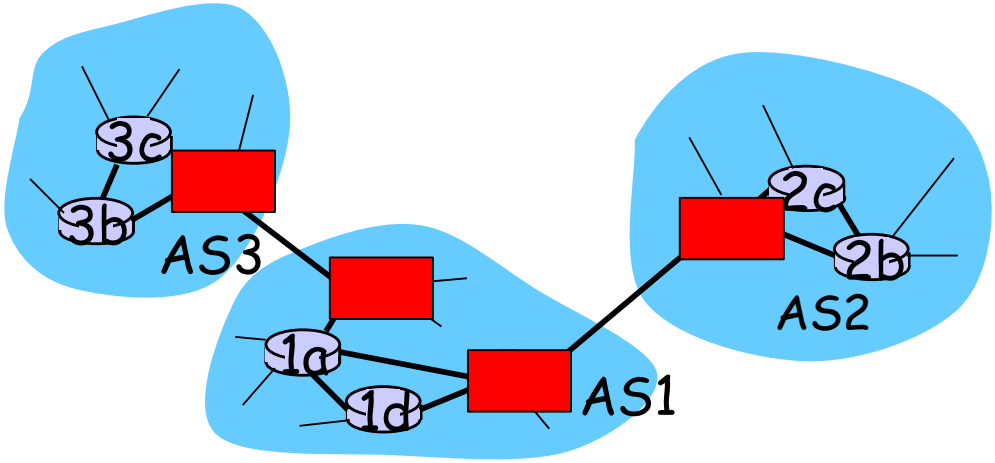
Hierarchical Routing



- Gateway router
  - Run inter-AS protocol to get the reachability information about the destinations that are reachable through neighboring AS(s).
  - Propagate this information to all routers within the AS
  - In the Internet, all ASes run the same inter-AS routing protocol, called BGP4 (Border Gateway Protocol).
  - configure its forwarding table based on the obtained information from both inter-AS routing protocol and intra-AS routing protocol.
    - Intra-AS sets entries for internal destinations
    - Inter-AS & Intra-As sets entries for external destinations

Hierarchical Routing

- Each other router in an AS
  - Run an intra-AS routing protocol to obtain the information on how to handle the packets to a destination within this AS.
  - Get the reachability information from gateway router(s) about the destinations that are reachable through neighboring AS(s).



  - configure its forwarding table based on the obtained information from both intra-AS routing protocol and gateway router(s).
    - Intra-AS routing protocol is used to set entries for internal destinations
    - Both Intra-AS and information from gateway router(s) are used to set entries for external destinations

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Routing in the Internet

- Internet intra-AS routing: IGP (Interior Gateway Protocol)

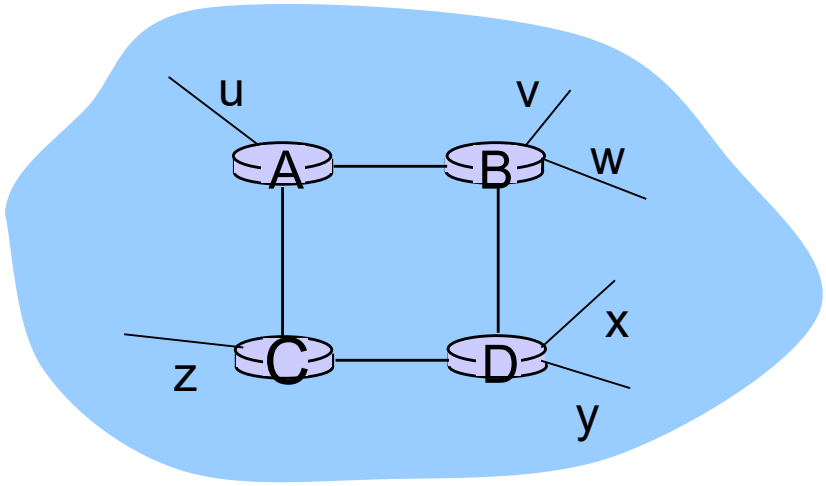
- Internet inter-AS routing: BGP (Border Gateway Protocol)

Intra-AS Routing

- Intra-AS routing in the Internet is also known as Interior Gateway Protocol (IGP) since it determines how routing is performed within an autonomous system (AS).
- Most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol

RIP (Routing Information Protocol)

- distance vector algorithm
  - distance metric: # hops (max = 15 hops), each link has cost 1
  - DVs exchanged with neighbors every 30 sec in response message (aka advertisement)
  - each advertisement: list of up to 25 destination *subnets (in IP addressing sense)*



from router A to destination *subnets:*

subnet	hops
u	1
v	2
w	2
x	3
y	3
z	2

OSPF (Open Shortest Path First)

- "open": publicly available
- 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
- advertisements flooded to *entire* AS
  - carried in OSPF messages directly over IP (rather than TCP or UDP
- *IS-IS routing* protocol: nearly identical to OSPF

OSPF (Open Shortest Path First)

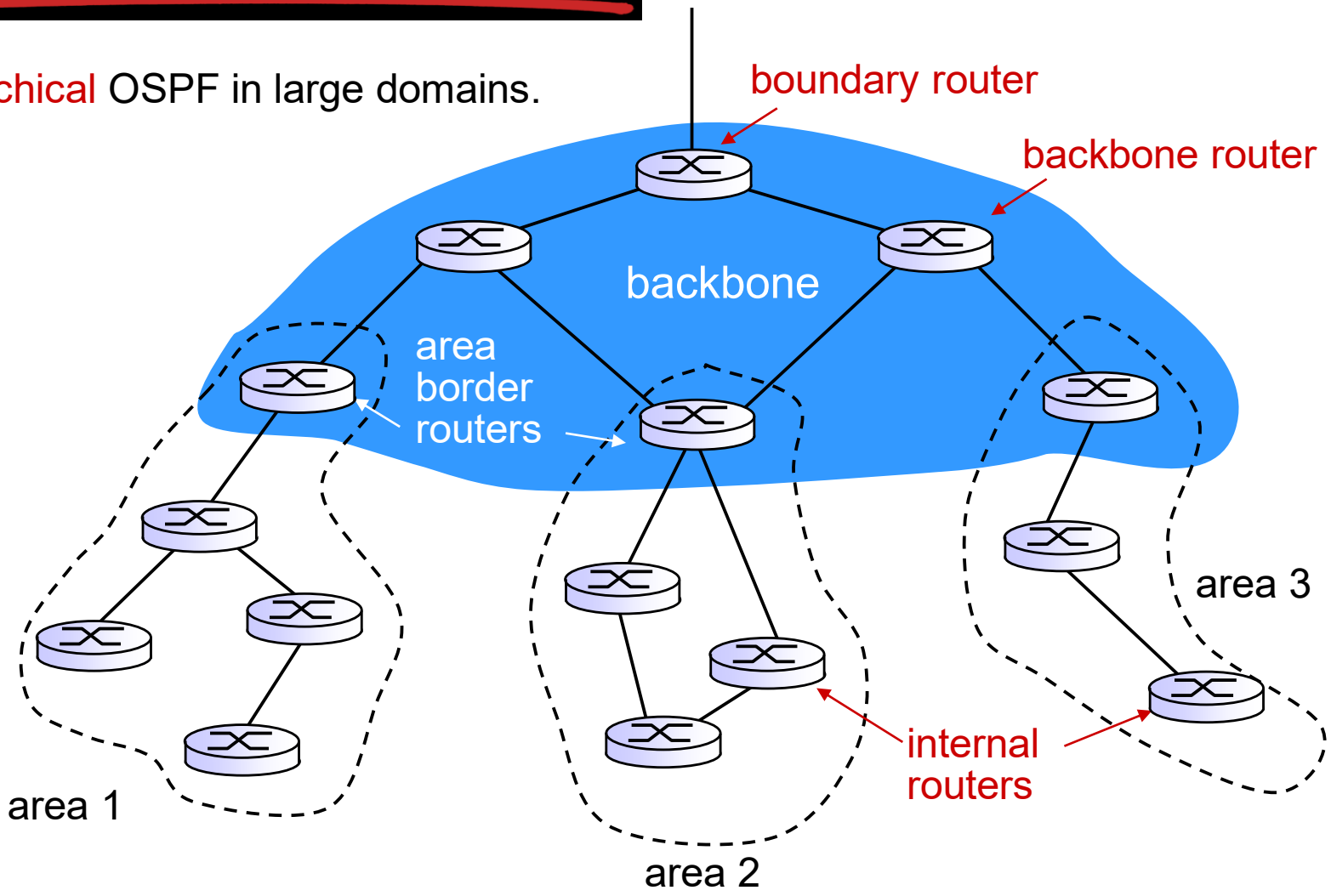
- "open": protocol specification is publicly available
- OSPF is hierarchical structure
  - Internal routers;
  - Area border routers
  - Backbone routers
  - Boundary routers
- Backbone area
  - One area in the AS is configured to be the backbone area.
  - The primary role of the backbone area is to route traffic between different areas in an AS: packets are first routed to an area border router, then routed through the backbone area to the area border router that is in the destination area, and then routed to the final destination.

OSPF (Open Shortest Path First)

- OSPF is based on the Link State (LS), but more complex
  - each AS may be divided into multiple areas
  - Each area uses link-state routing such as Dijkstra's algorithm, with all routers being equal.
  - Within each area, at least one router is connected to the backbone area. The router(s) is called area border router(s) and responsible for routing packets outside the area.

Hierarchical OSPF

hierarchical OSPF in large domains.



Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): it is the standard for inter-AS routing in Internet.
  - "glue that holds the Internet together"
- BGP Basic:
  - One or more routers in each AS use BGP to communication with other such routers in other ASs to exchange routing information.
  - These routers are called as BGP peers.
  - Obtain subnet reachability information from neighboring ASs
  - Propagate the reachability information to all routers within the AS.
  - Determine "good" routes to subnets based on reachability information and policy

how Internet has made it possible to scale to millions of users.

- Routers are aggregated into autonomous systems (ASs). Within an AS, all routers run the same intra-AS routing protocol.
- Special gateway routers in the various ASs run the inter-autonomous system routing protocol that determines the routing paths among the ASs.
- The problem of scale is solved since an intra-AS router need only know about routers within its AS and the gateway router(s) in its AS.

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

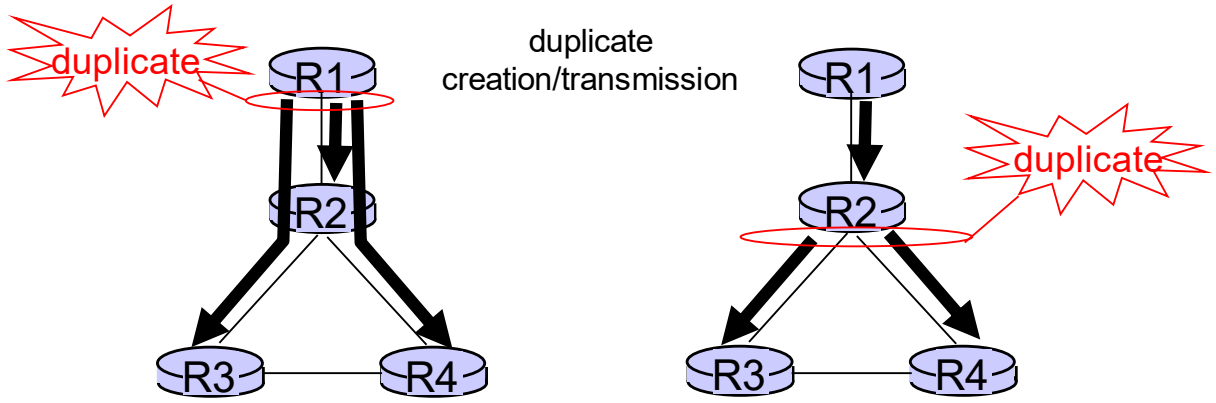
- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Three types of communications

- Unicast
  - Single source, single receiver
- Broadcast
  - Send same packet to all receivers in a LAN, subnet, or in an organization.
- Multicast
  - Send some packet to multiple receivers
  - Applications: bulk data transfer (e.g., software upgrade from the developer to users needing the upgrade), remote education (the transfer of the audio, video, and text of a live lecture to a set of students) , teleconference, etc.

Broadcast / Multicast



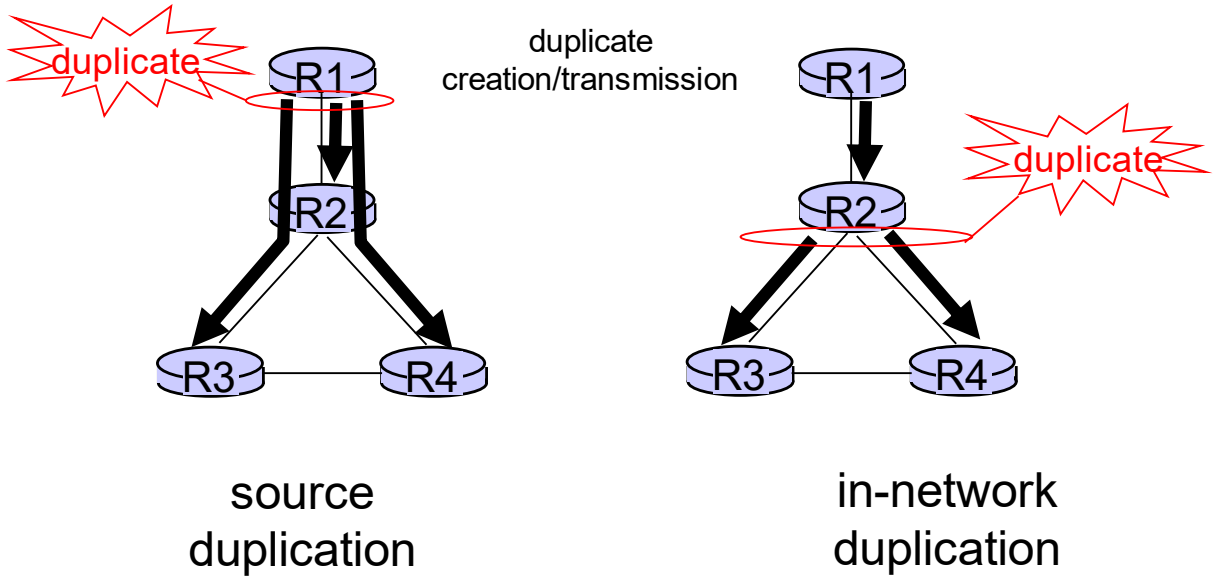
source duplication

in-network duplication

Source duplication (N-way unicast): Given N destination nodes, the source node simply makes N copies of the packet, addresses each copy to a different destination, and then transmits the N copies to the N destinations using unicast routing.

- Simple (advantage)
- Inefficient (disadvantage)
- Source may not know the addresses of all receivers (disadvantage)

Broadcast / Multicast



In-network duplication: to make copy of packet at the network nodes, rather than at the source node. It is more efficient than source duplication.

Chapter 4: *done!*

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format, IPv4 addressing, ICMP, IPv6

4.5 routing algorithms

- link state, distance vector, hierarchical routing

4.6 routing in the Internet

- RIP, OSPF, BGP

4.7 broadcast and multicast routing

- understand principles behind network layer services:
  - network layer service models, forwarding versus routing how a router works, routing (path selection), broadcast, multicast
- instantiation, implementation in the Internet