IP Layer

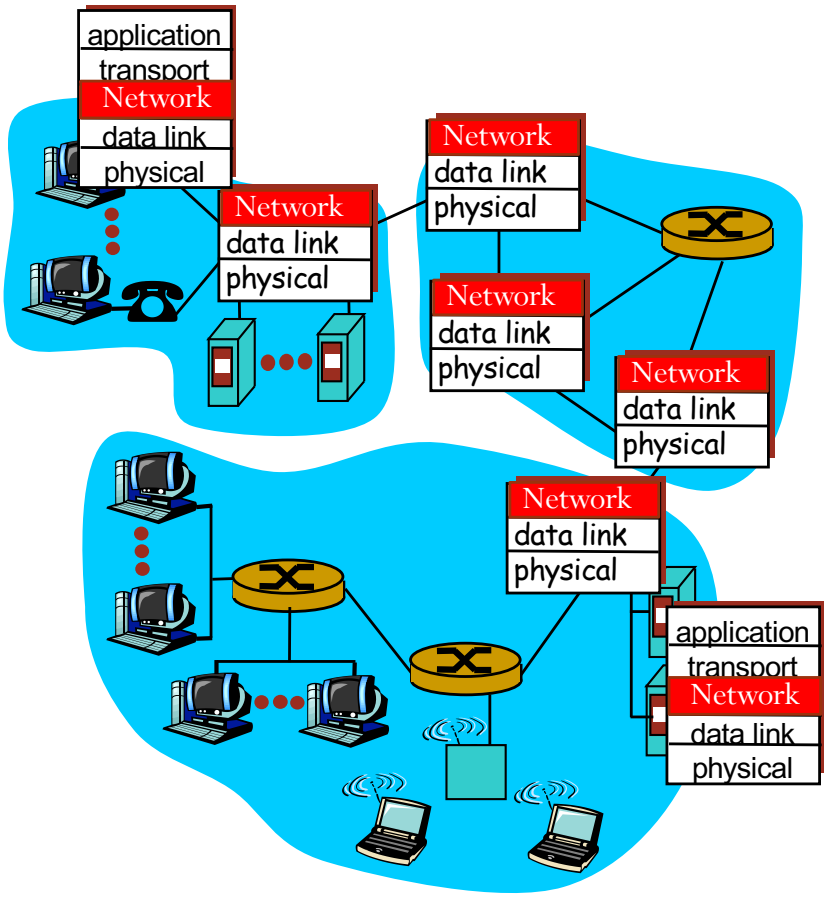
Rong Zheng

Outline

- Introduction
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - IPv6
  - ICMP
- Routing algorithms

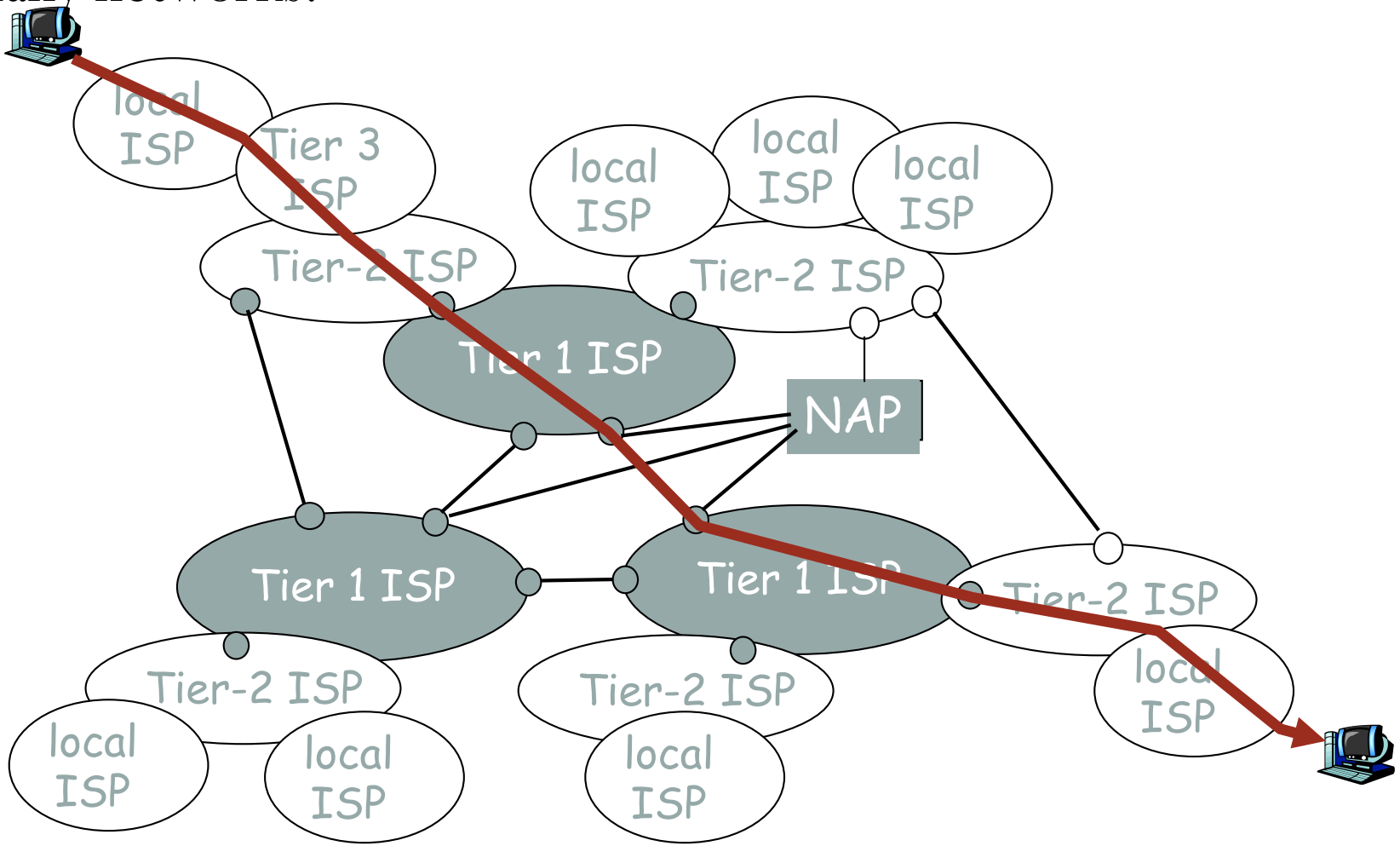
Network layer

- transport segment from sending to receiving hosts
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in every host, router
- router examines header fields in all IP datagrams passing through it



Internet structure: network of networks

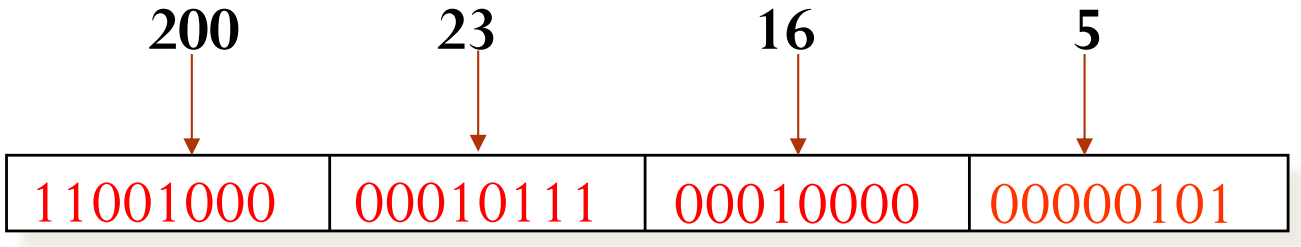
- Message passes through many networks!



IPv4 Address

- IPv4 Address: Unique 32-bit number associated with a host, router interface
- Represented with the dotted-quad notation

  e.g.: **200.23.16.5**



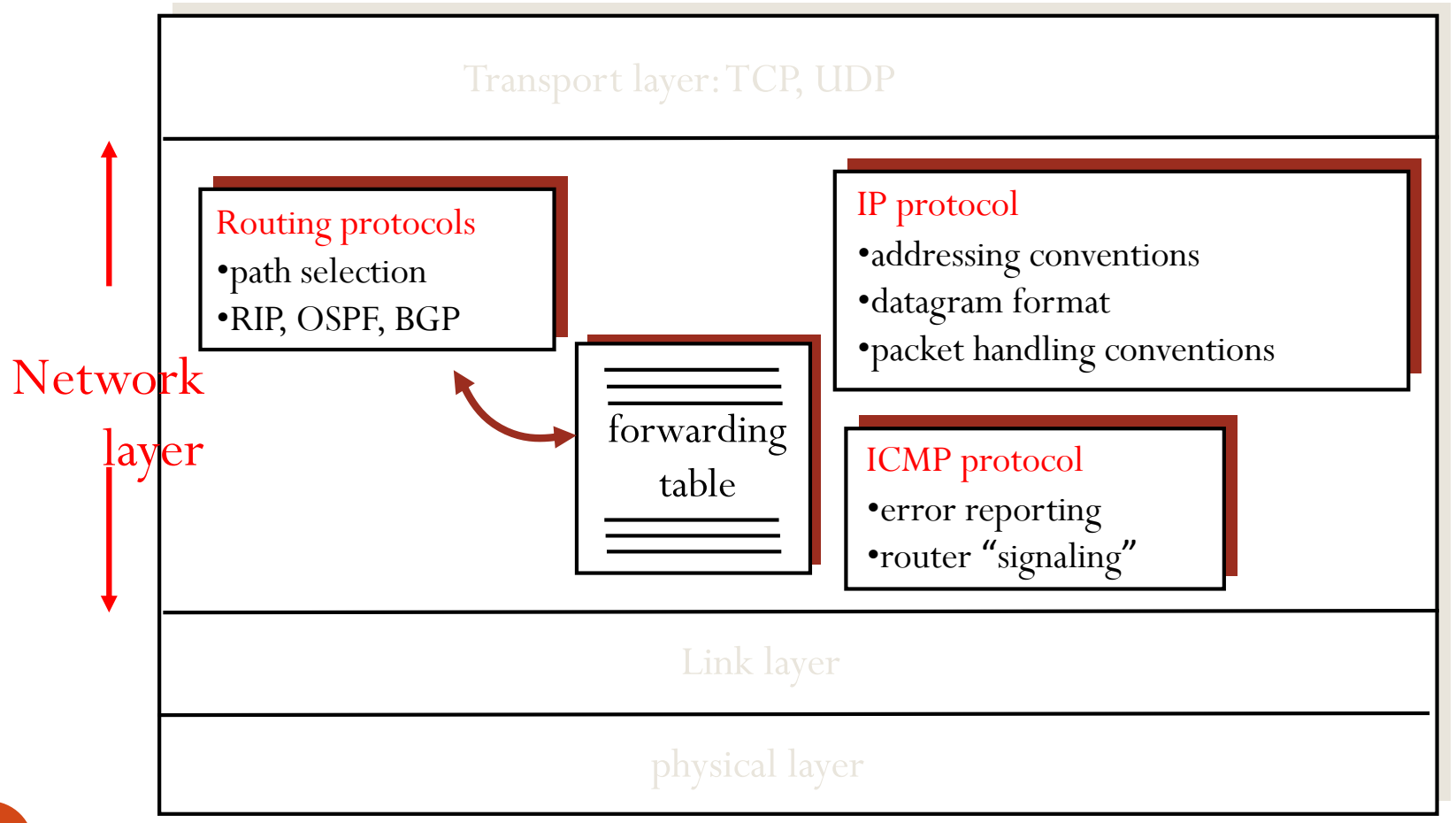
- interface: connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one physical interface

Outline

- Introduction
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing/IPv6
  - DHCP, NAT
  - ICMP
- Routing algorithms

The Internet Network layer

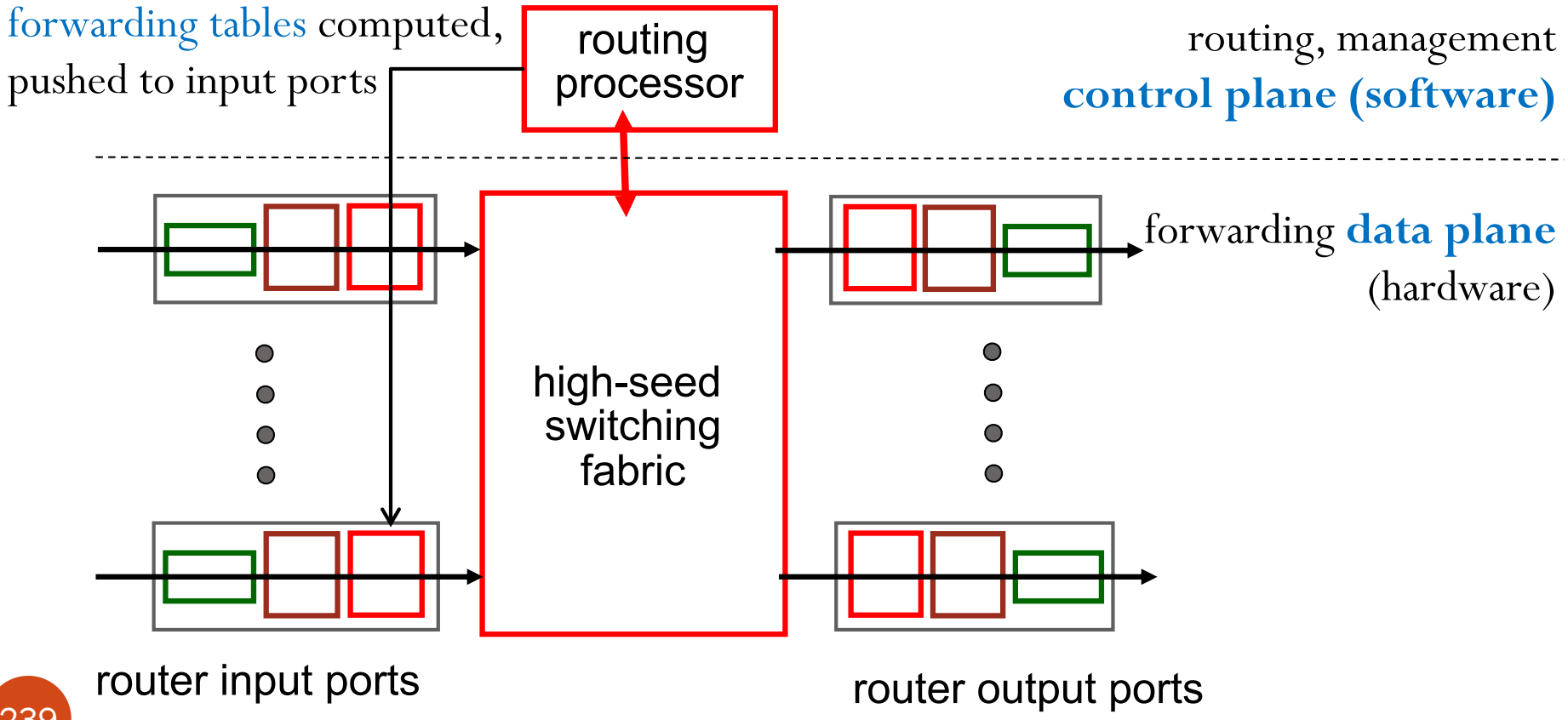
- Host, router network layer functions:



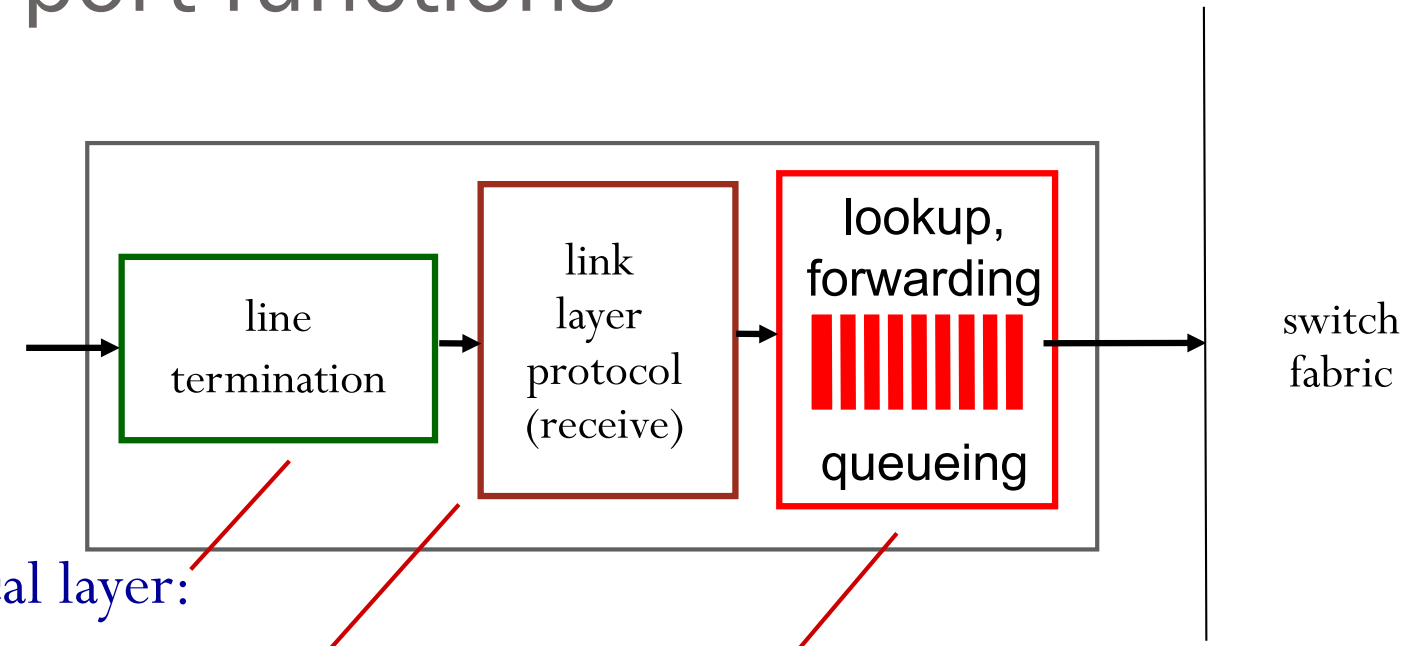
Router architecture overview

two key router functions:

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



Input port functions



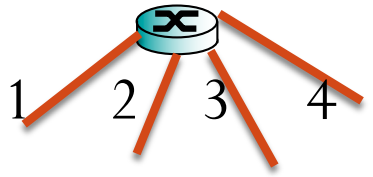
physical layer:
bit-level reception

data link layer:
e.g., Ethernet
see chapter 5

decentralized switching:

- given datagram dest., lookup output port using forwarding table in input port memory *("match plus action")*
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Forward table



Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

32 bits Address →4 billion possible entries

Forward table: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 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

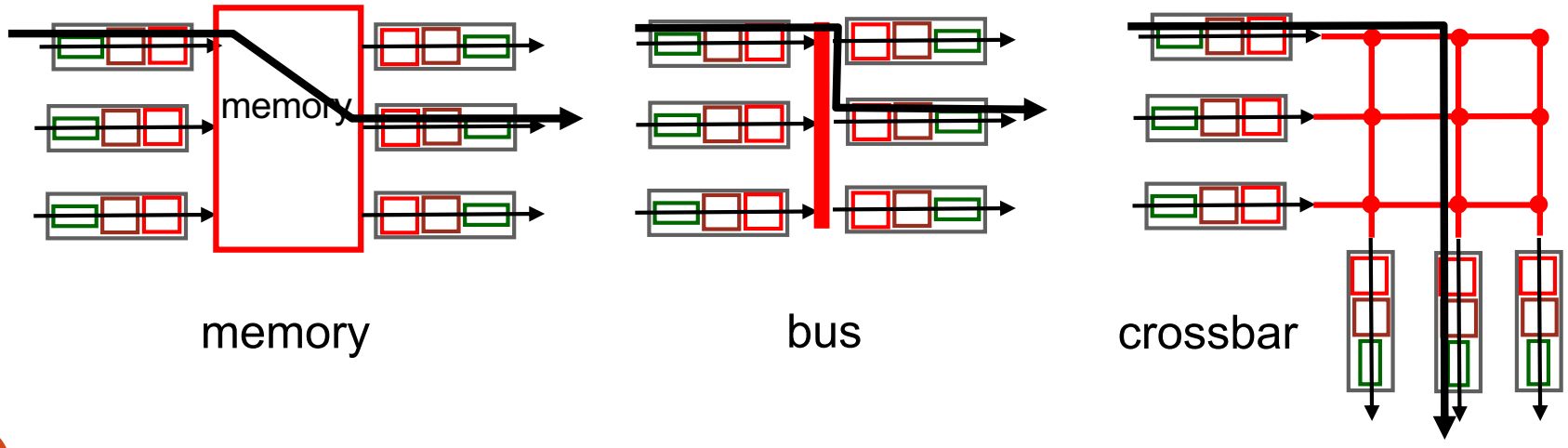
Forward table:Longest prefix matching

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

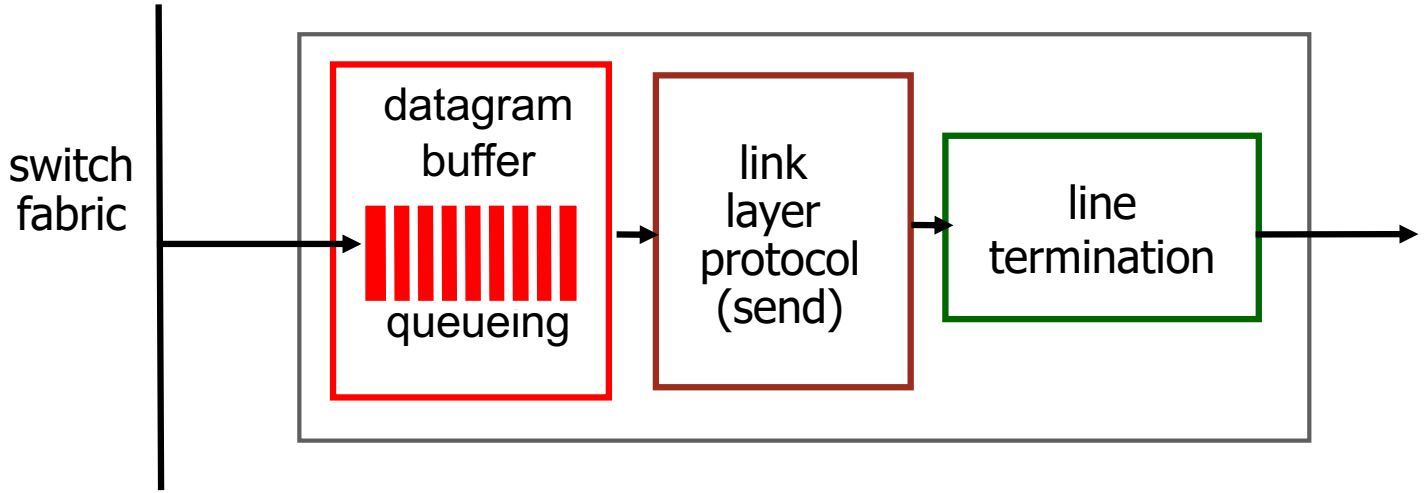
- Which interface?
- DA: 11001000 00010111 00010110 10100101 0
- DA: 11001000 00010111 00011000 11100000 1

Switching fabrics

- transfer packet from input buffer to appropriate output buffer
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - **N** inputs: switching rate **N** times line rate desirable
- three types of switching fabrics



Output ports

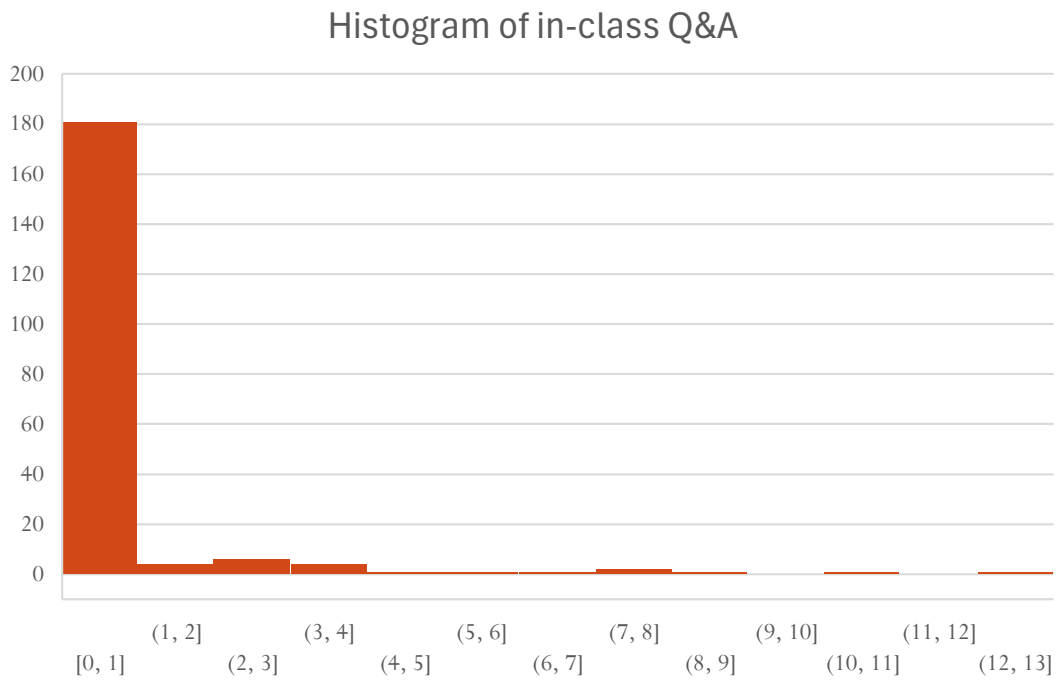


- *buffering* required when datagrams arrive from fabric faster than the transmission rate
- *scheduling discipline* chooses among queued datagrams for transmission

Datagram (packets) can be lost due to congestion, lack of buffers

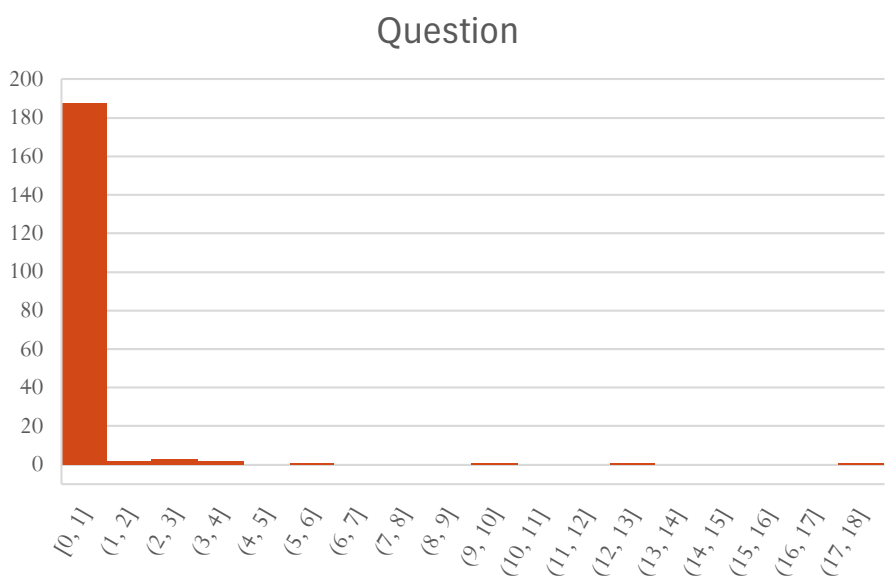
Priority scheduling – who gets best performance, network neutrality

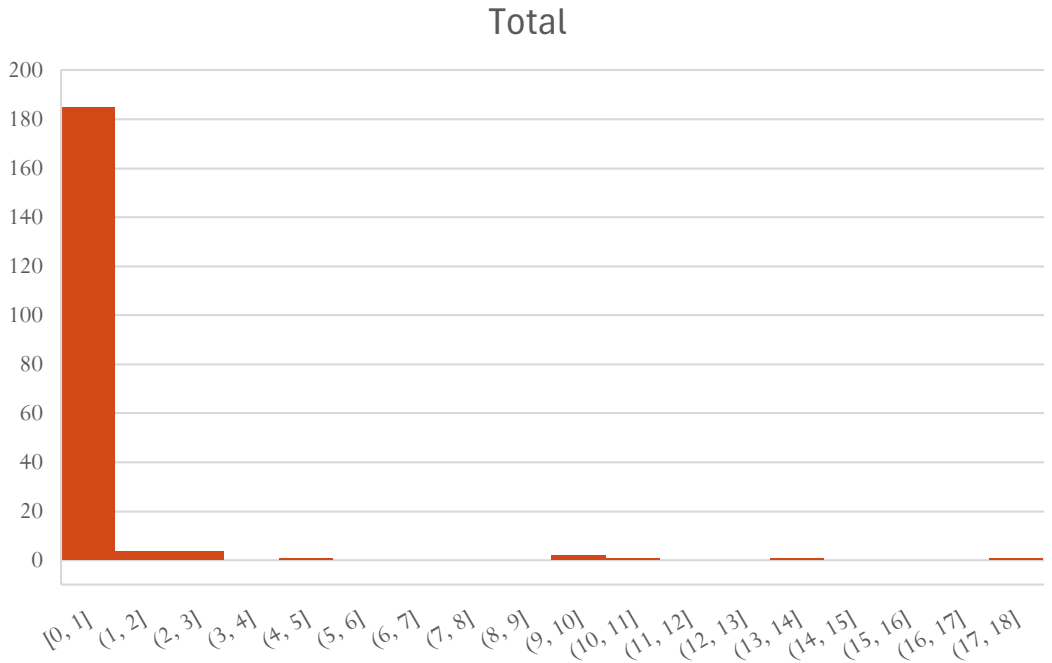
In-class Participation Stats



400297560
400300652
400343384
400317089
400071512
400314972
400226757
400330872
400411764
400379495
400319506
400320263
400321546
400393678
400361322
400318027
400373097
400306555
400396965
400331004

Avenue Stats

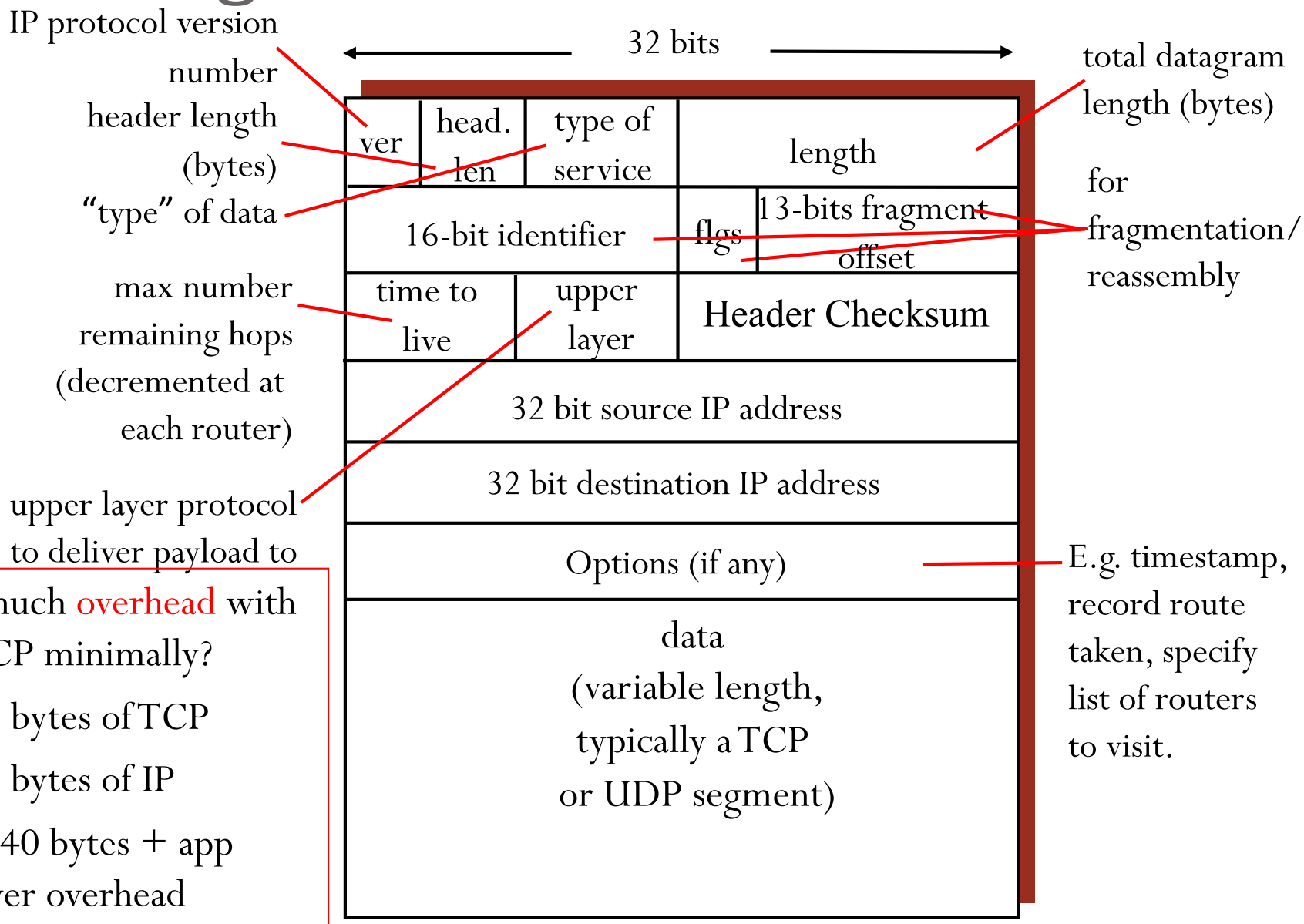




Outline

- Introduction
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing/IPv6
  - DHCP, NAT
  - ICMP
- Routing algorithms

IP datagram format



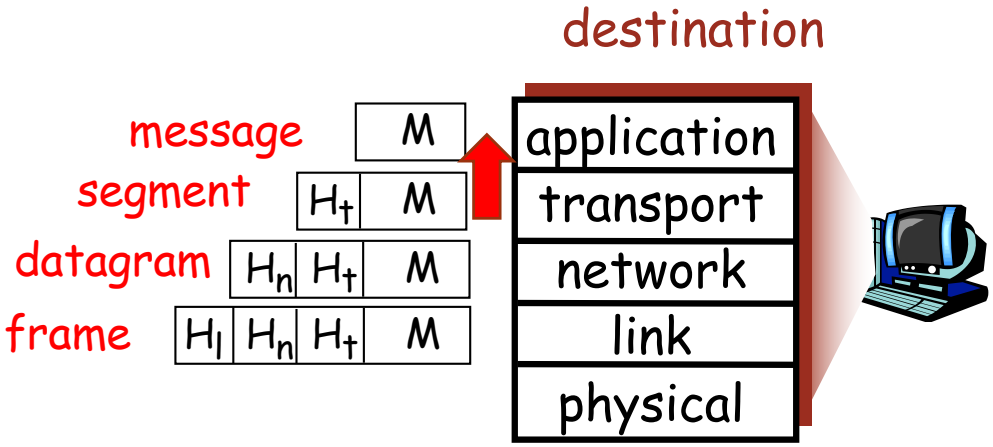
how much overhead with TCP minimally?

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

IP: How to Handle Packet

- Protocol (8 bits)
  - Identifies the higher-level protocol
  - Important for demultiplexing at receiving host
- Most common examples

  6 for TCP, 17 for UDP



IP Header: Checksum, TTL

- Checksum (16 bits)
  - Particular form of checksum over packet header
  - If not correct, router discards packets
  - Checksum recalculated at every router
- Time-to-Live (TTL) Field (8 bits)
  - Decremented 1 at each hop, packet discarded if reaches 0

Source : http://www.potaroo.net/tools/ipv4/

IPv4 Address Exhaustion

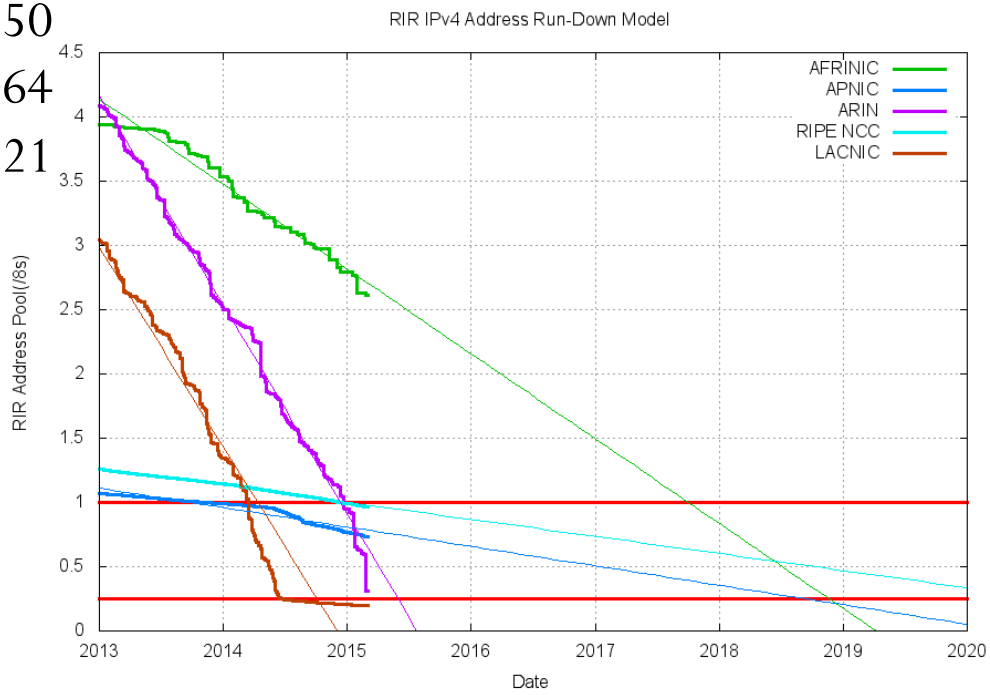
IANA Unallocated Address Pool Exhaustion:

03-Feb-2011

Projected RIR Address Pool Exhaustion Dates:

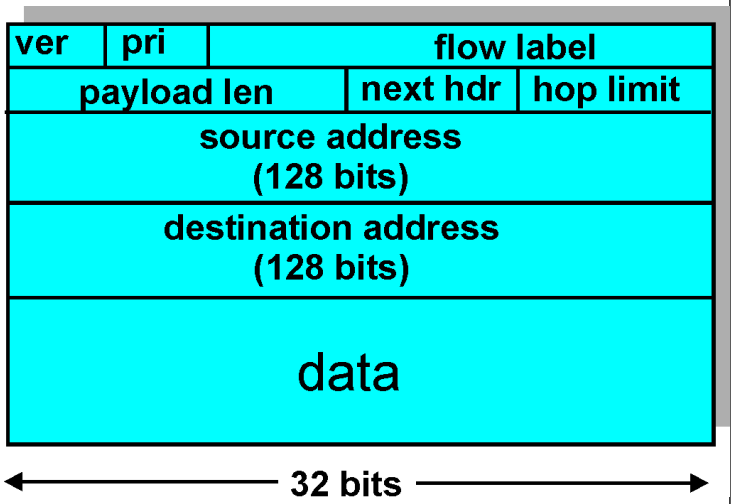
RIR	Projected Exhaustion Date	Remaining Addresses in RIR Pool (/8s)
APNIC:	**19-Apr-2011** (actual)	0.7331
RIPE NCC:	**14-Sep-2012** (actual)	0.9634
LACNIC:	**10-Jun-2014** (actual)	0.1950
ARIN:	**16-May-2015**	0.3064
AFRINIC:	**09-Jan-2019**	2.6121

32-bit address space soon to be completely allocated.



IPv6

- Initially motivated by address exhaustion
  - Address length four times as big (128 bits vs. 32 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
    - no checksum



Priority: identify priority among datagrams in flow

Flow Label: identify datagrams in same "flow." (concept of"flow" not well defined).

Next header: identify upper layer protocol for data

Other Changes from IPv4

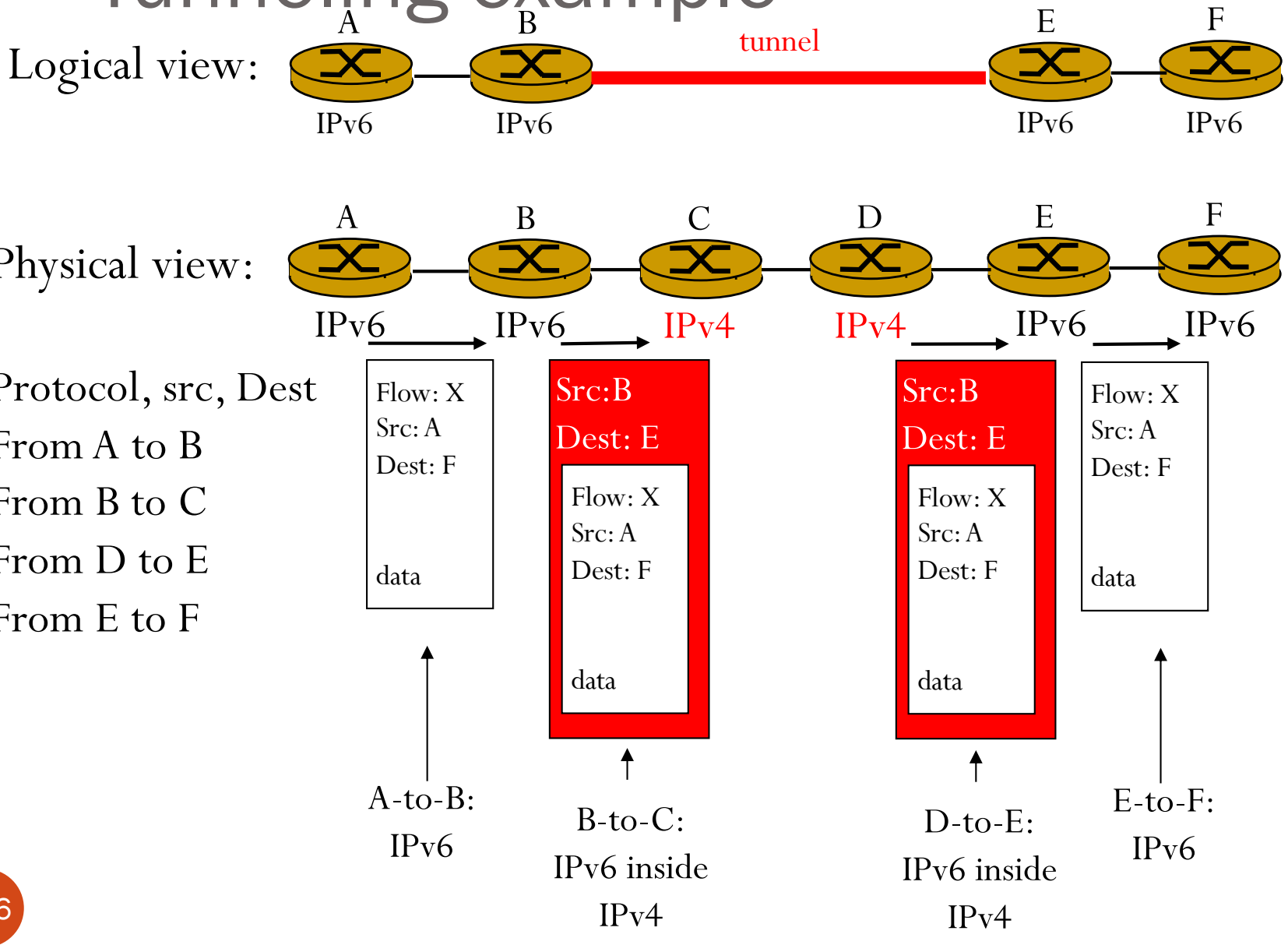
- Checksum: removed entirely to reduce processing time at each hop
- Options: allowed, but outside of header, indicated by "Next Header" field
- ICMPv6: new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneous
- How will the network operate with mixed IPv4 and IPv6 routers?

- Tunneling: IPv6 carried as payload in IPv4 datagram among IPv4 routers

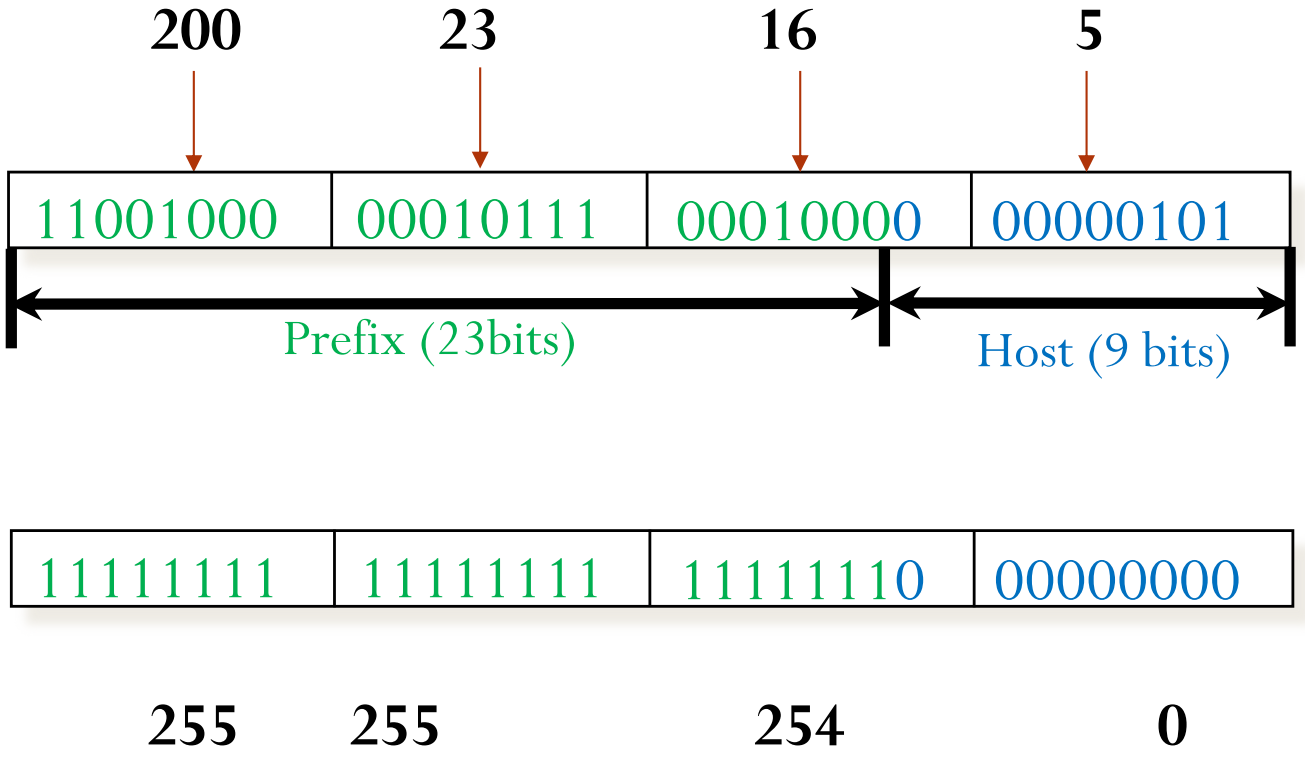
Tunneling example



IPv4 Addressing Recap

- IPv4 Address: Unique 32-bit number associated with a host, router interface
- Represented with the dotted-quad notation

  e.g.: **200.23.16.5**



IP addressing: Classful addressing

Class	Leftmost bits	Start address	Finish address	Size of *network number* bit field
A	0xxx	0.0.0.0	127.255.255.255	8
B	10xx	128.0.0.0	191.255.255.255	16
C	110x	192.0.0.0	223.255.255.255	24
D	1110	224.0.0.0	239.255.255.255	
E	1111	240.0.0.0	255.255.255.255	

- Class D for multicast
- Broadcast address 255.255.255.255
- 127.0.0.1 is the loopback address
- Problem: class C is too small, class B is too big!

IP addressing: Private addressing

- Private Addresses: free to use internally

Class	start address	finish address	blocks	Hosts
A	10.0.0.0	10.255.255.255	10.0.0.0/8	16777216
B	172.16.0.0	172.31.255.255	172.16.0.0/12	1048576
C	192.168.0.0	192.168.255.255	192.168.0.0/16	65536

e.g., 172.17.54.86; 192.168.1.25

IP addressing: CIDR

- CIDR: Classless InterDomain Routing
- Idea: Flexible division between prefix and host addresses
- Intention: offer a better tradeoff between size of the routing table and efficient use of the IP address space

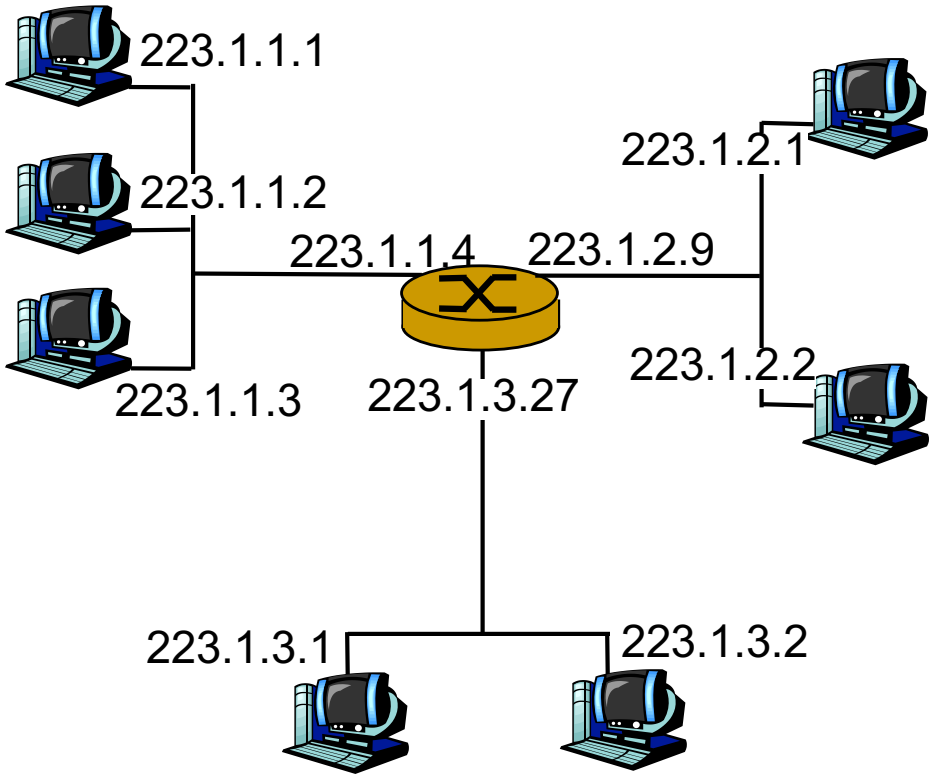
IP addressing: CIDR Example

- Suppose a network has 90 computers. What is the most efficient # of bits for prefix part and host parts?

  - allocate 7 bits for host addresses (since $2^6 < 90 < 2^7$)
  - remaining 32 - 7 = 25 bits as network prefix
  - E.g., 223.1.1.0/25

Subnets Example 1

- Subnets with 24 prefix part.
- How many subnets?
- Notation for the subnets?

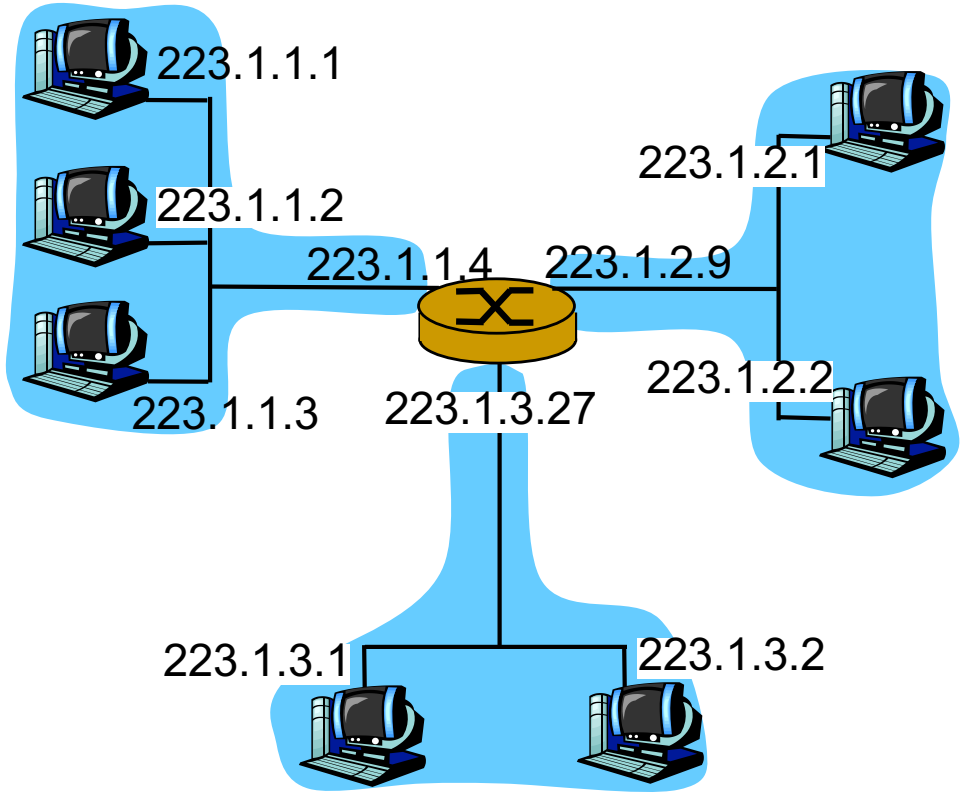


Subnets Example 1

- Subnets with 24 prefix part.
- How many subnets?

  3

- Notation for the subnets?

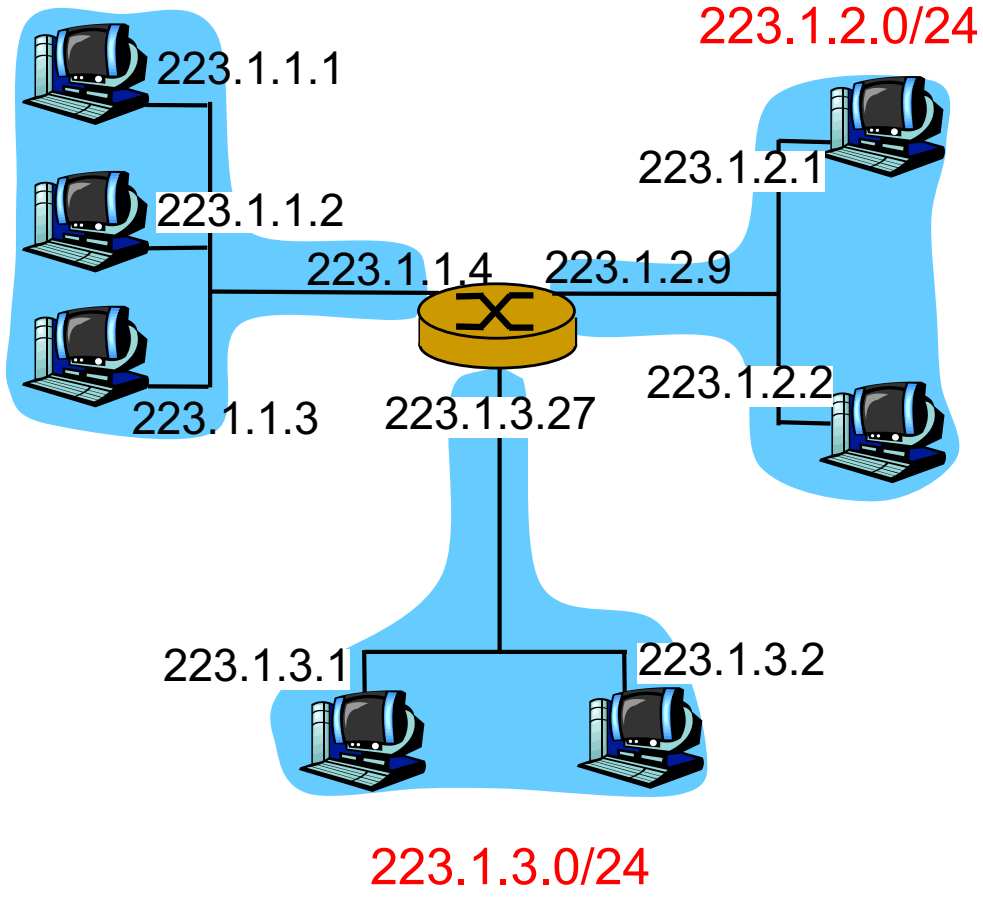


Subnets Example 1

- Subnets with 24 prefix part. 223.1.1.0/24
- How many subnets?

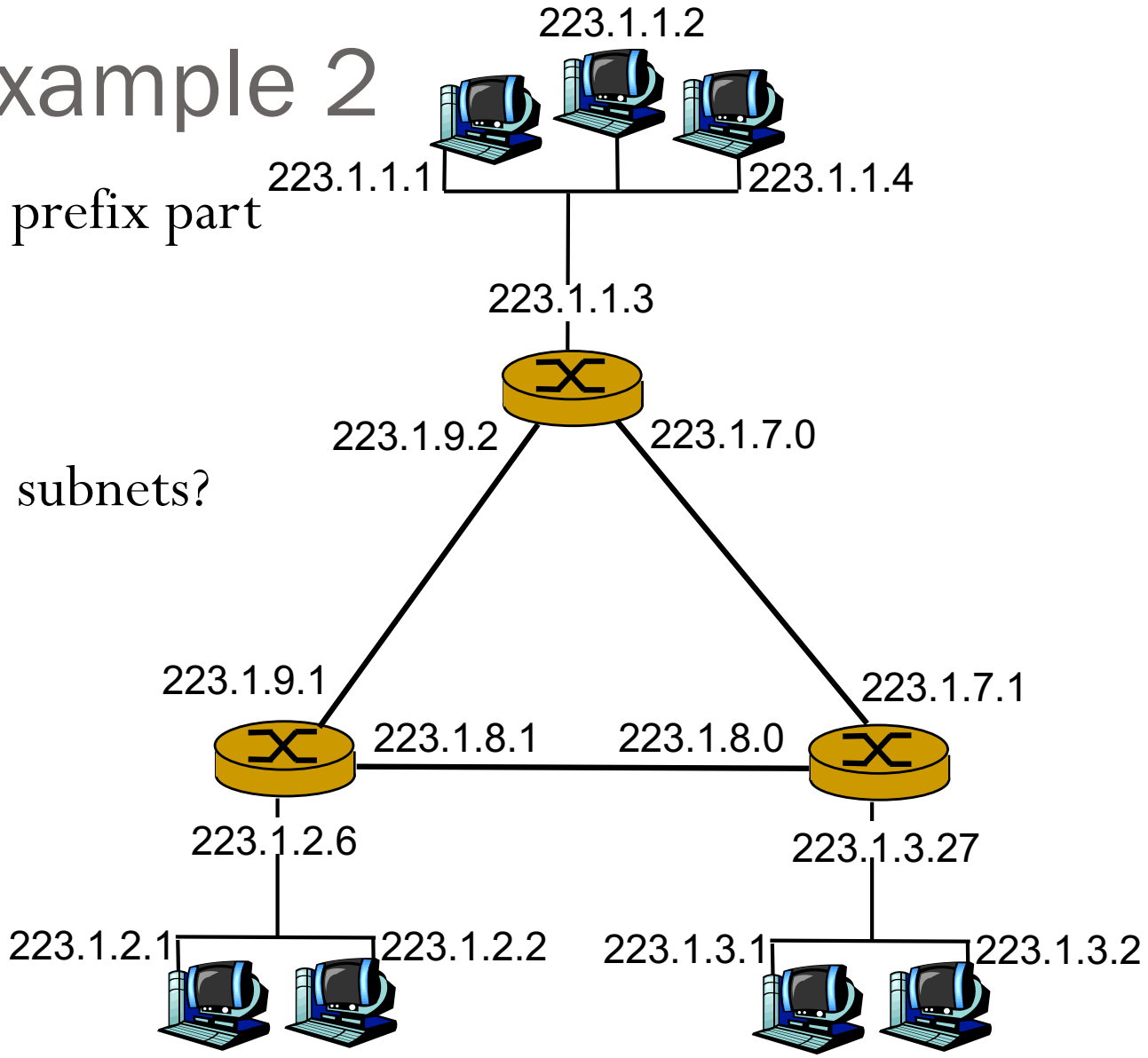
  3

- Notation for the subnets?



Subnets Example 2

- Subnets with 24 prefix part
- How many?
- Notation for the subnets?



Subnets Example 2

- Subnets with 24 prefix part
- How many?

  6
- Notation for the subnets?

  223.1.1.0/24

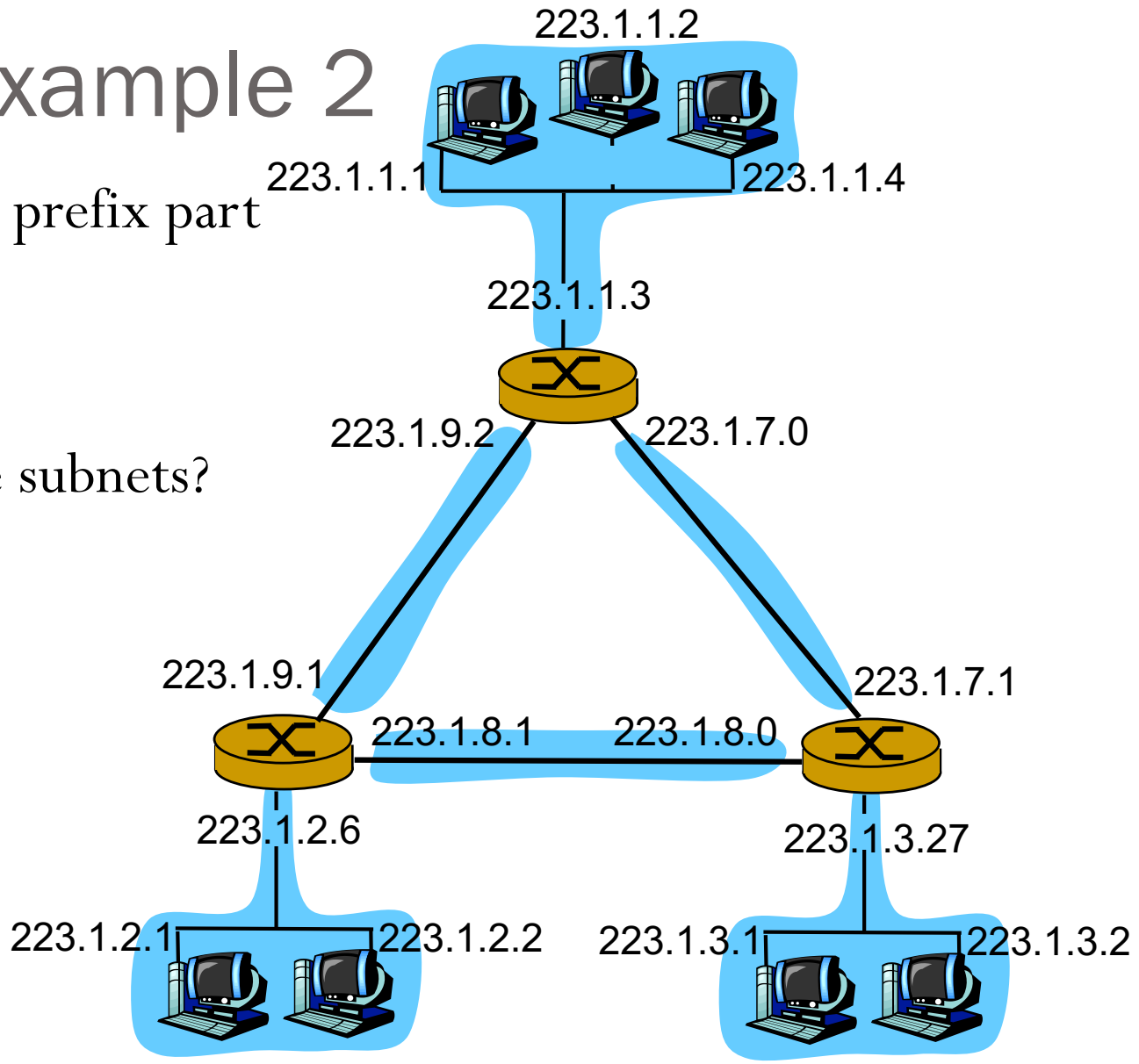
  223.1.2.0/24

  223.1.3.0/24

  223.1.7.0/24

  223.1.8.0/24

  223.1.9.0/24

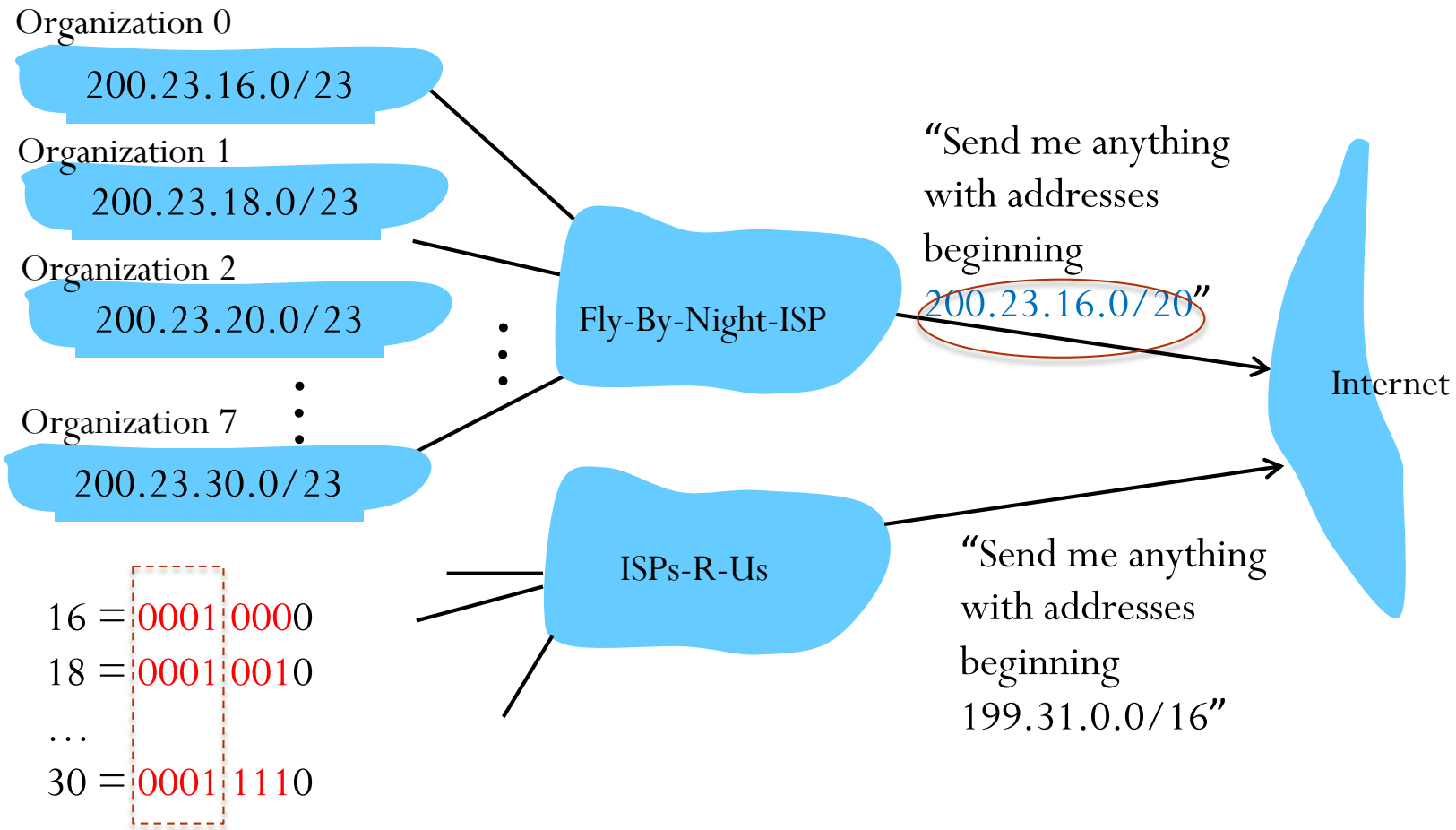


Hierarchical addressing: allocation

- Internet Corporation for Assigned Names and Numbers (ICANN) assigns IP address portion to…
  - Regional Internet Registries, which assign subnet portion to
    - Large institutions (ISPs), which assign addresses to…
      - Hosts

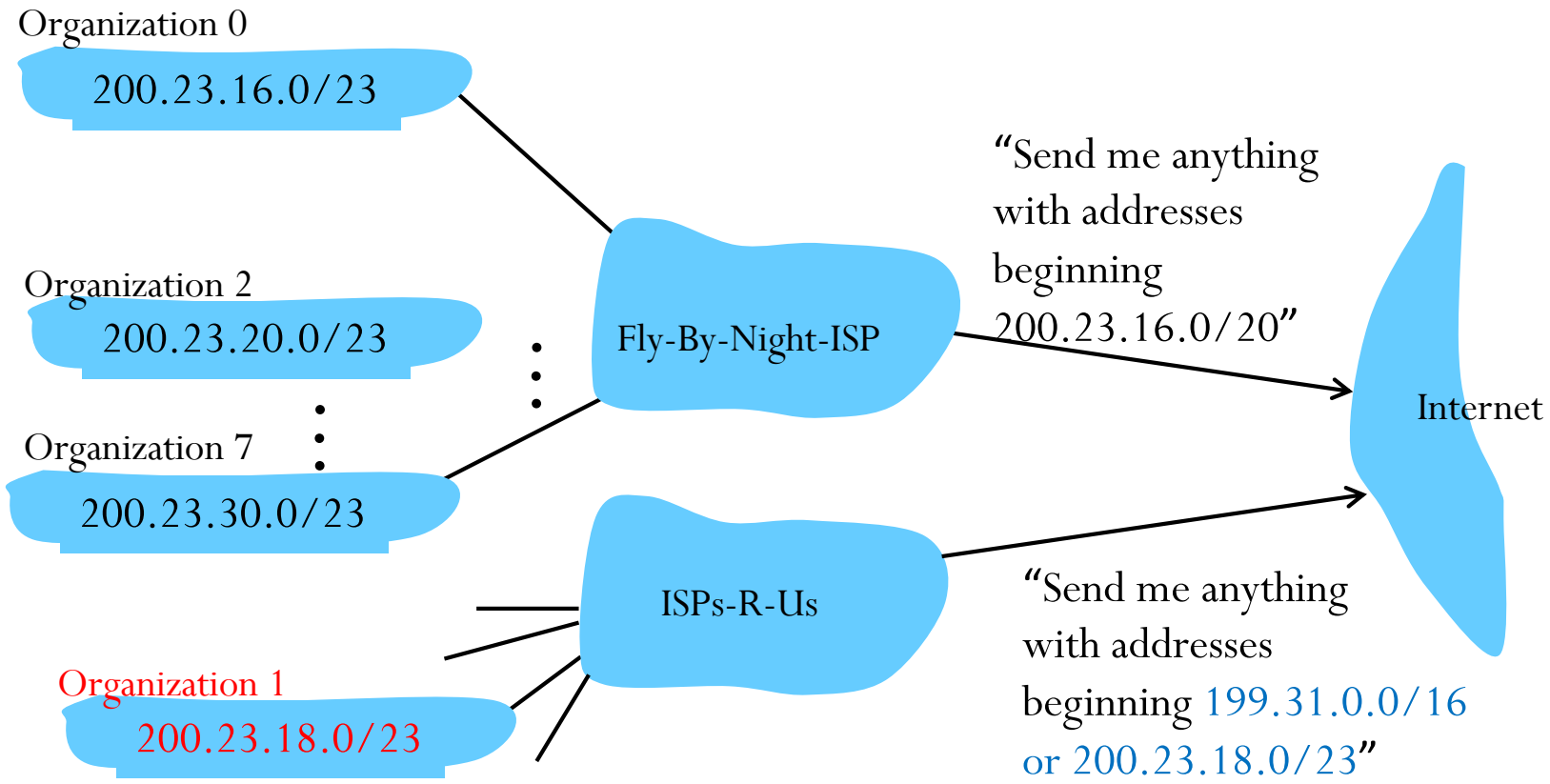
Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1



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"

DHCP: Dynamic Host Configuration Protocol

DHCP message encapsulated in UDP, encapsulated in IP

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

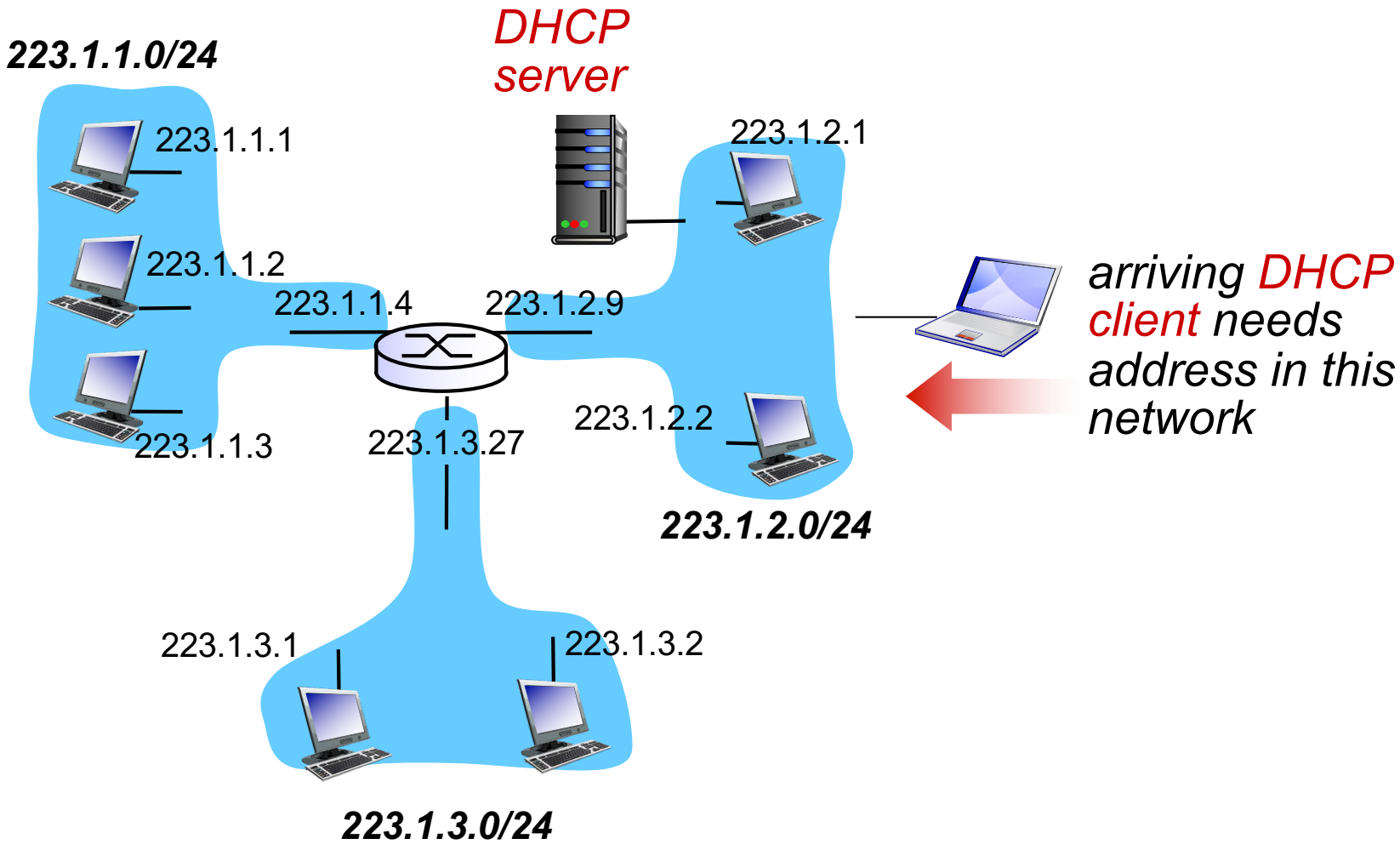
- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network

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

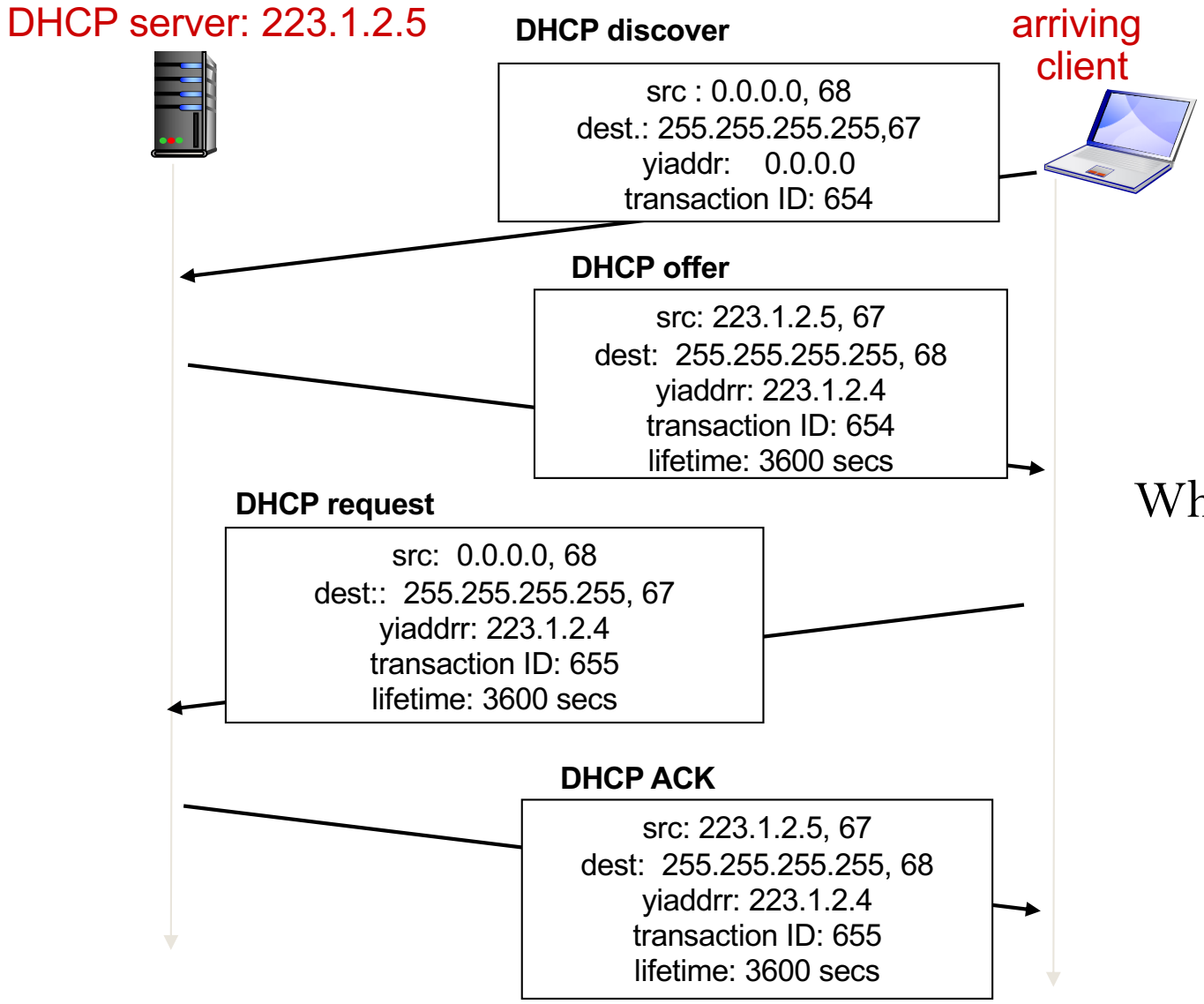
DHCP client-server scenario

what would be the dest & src IP addresses?



yiaddr: Your IP address

DHCP client-server scenario



Why not just take it?

DHCP: more than IP addresses

- **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)

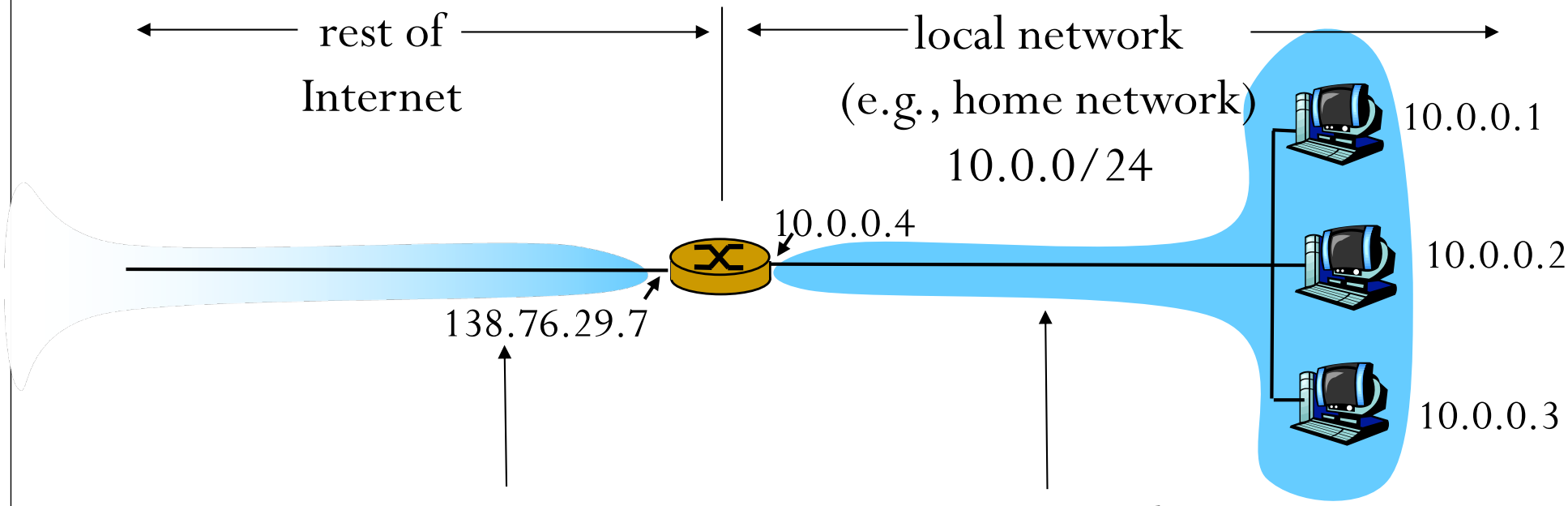
NAT: Network Address Translation

Motivation: a private (home) network uses just one IP address as far as outside world is concerned:

- no need to be allocated range of addresses from ISP
- can change addresses of devices in a private network without notifying outside world
- can change ISP without changing addresses of devices in a private network
- devices inside not explicitly addressable by or visible to outside world (a security plus).

Q: What if I have more hosts than available public IP addresses?

NAT: Network Address Translation

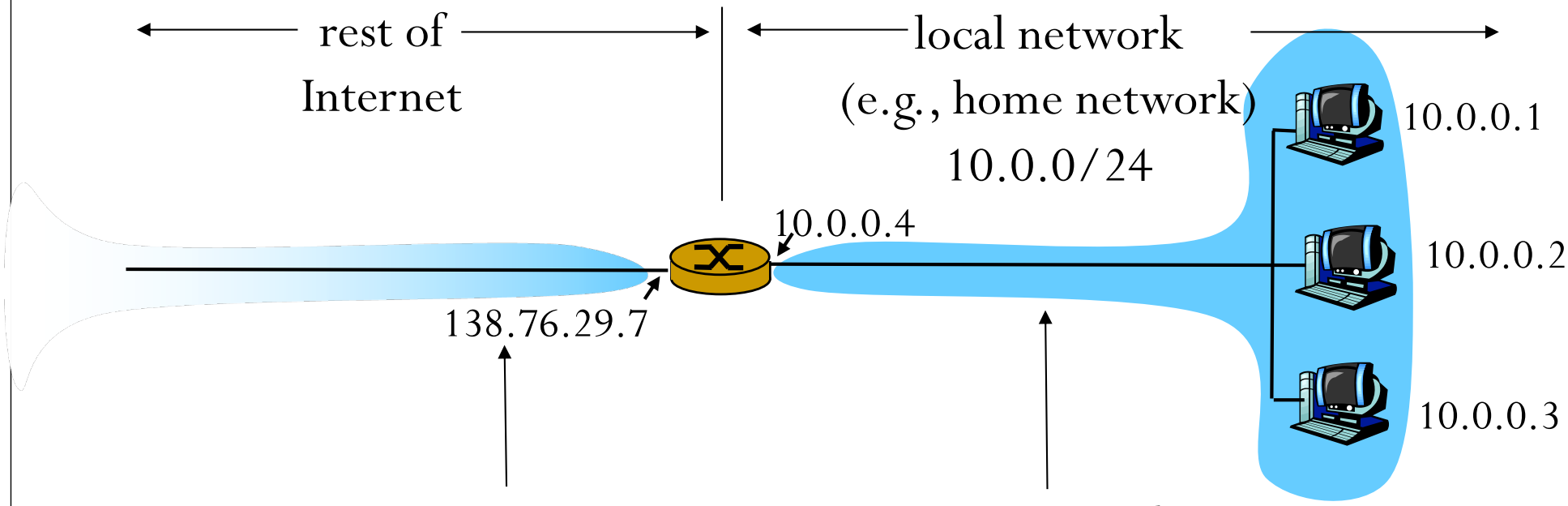


All datagrams *leaving* local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

Q: What if I have more hosts than available public IP addresses?

NAT: Network Address Translation



All datagrams *leaving* local network have same single source NAT IP address: 138.76.29.7, different source port numbers

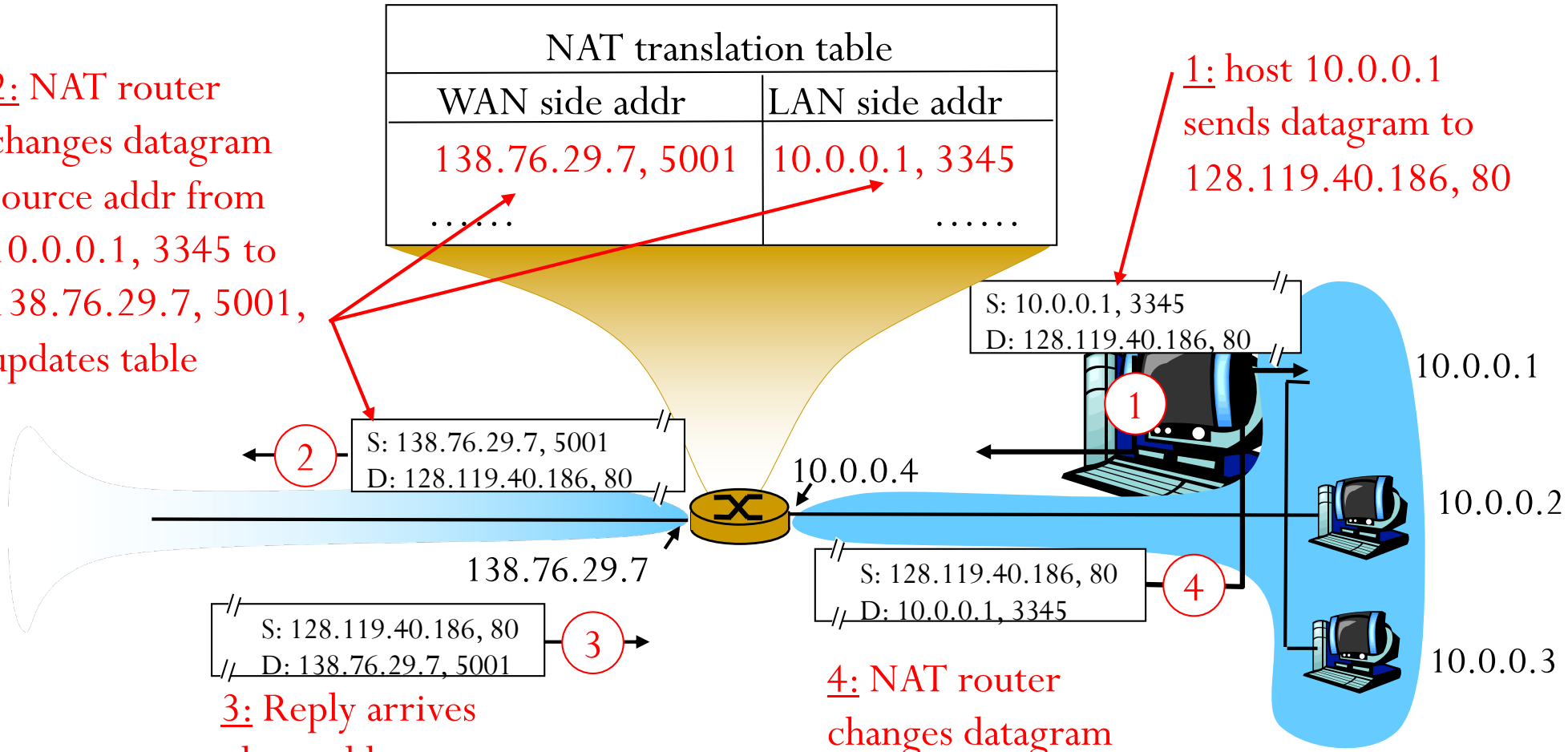
Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: Network Address Translation

Implementation: NAT router must:

- outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  - . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr.
- remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair
- incoming datagrams: replace (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: Network Address Translation



NAT: Network Address Translation

- 16-bit port-number field:
  - > 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, eg, P2P applications
  - address shortage should ideally be solved by IPv6

Outline

- Introduction
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - IPv6
  - ICMP
- Routing algorithms

ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - Status check: echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams, protocol 1
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

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

What ICMP messages do you use for traceroute?

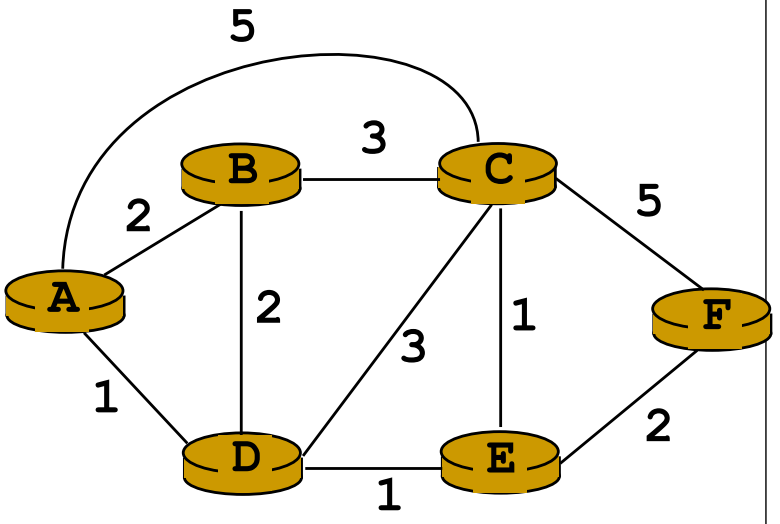
Traceroute and ICMP

- Source sends series of UDP segments to dest
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router& IP address
- When ICMP message arrives, source calculates RTT
  - Traceroute does this 3 times
- Stopping criterion
  - UDP segment eventually arrives at destination host
  - Destination returns ICMP "dest port unreachable" packet (type 3, code 3)
  - When source gets this ICMP, stops.

Routing Protocols

Routing Protocols

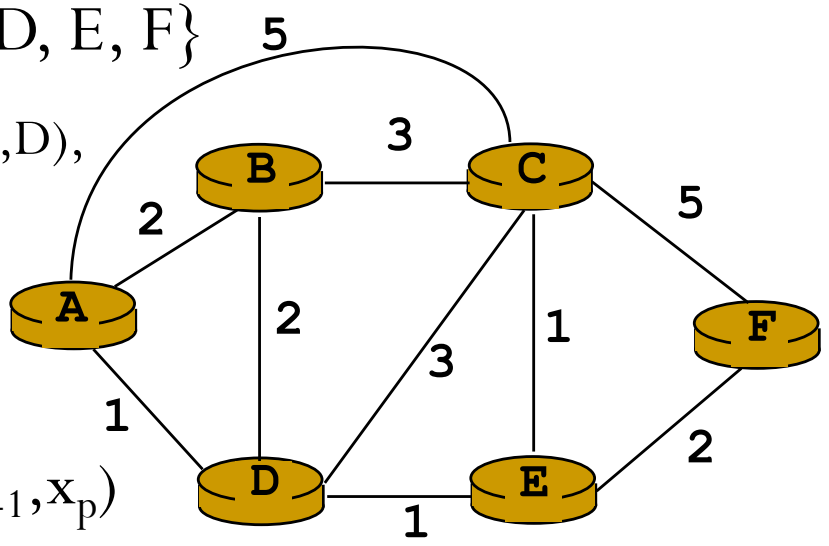
- Routing protocols implemented on the routers of a network
  - Establish paths between nodes
- Network modeled as a graph
  - Routers are graph vertices
  - Links are edges
  - Edges have an associated "cost"
    - e.g., distance, loss, latency



- Goal: compute a "good" path from source to destination
  - "good" usually means the shortest (least cost) path
  - Q: How to compute a "good" path?
    - Centralized vs distributed algorithms

Graph abstraction

- Graph: G = (N, E)
- N = set of routers = {A, B, C, D, E, F}
- E = set of links = { (A,B), (A,D), (B,D), (B,C), (D,C), (D,E), (C,E), (C,F), (E,F) }



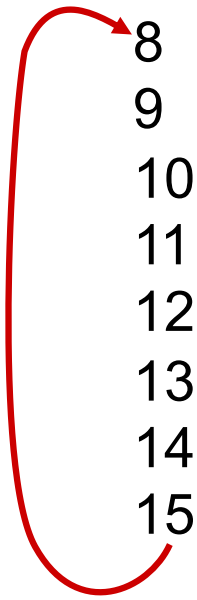
- Cost of path(x_1, x_2, x_3, …, x_p) = $c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$
  - e.g. cost of path (B,A,D) = c(B,A) + c(A,D) = 2 + 1 = 3 > 2

A Link-State Routing Algorithm

- Impl 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; $= \infty$ 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'
```

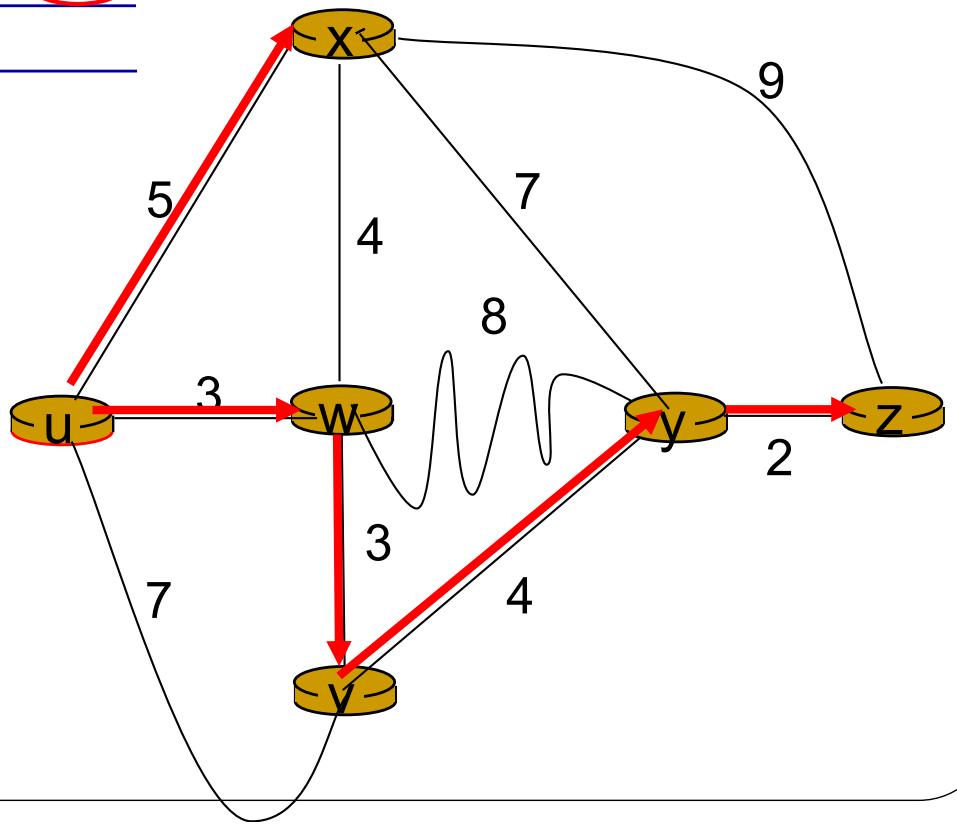


Dijsktra's Algorithm: Example

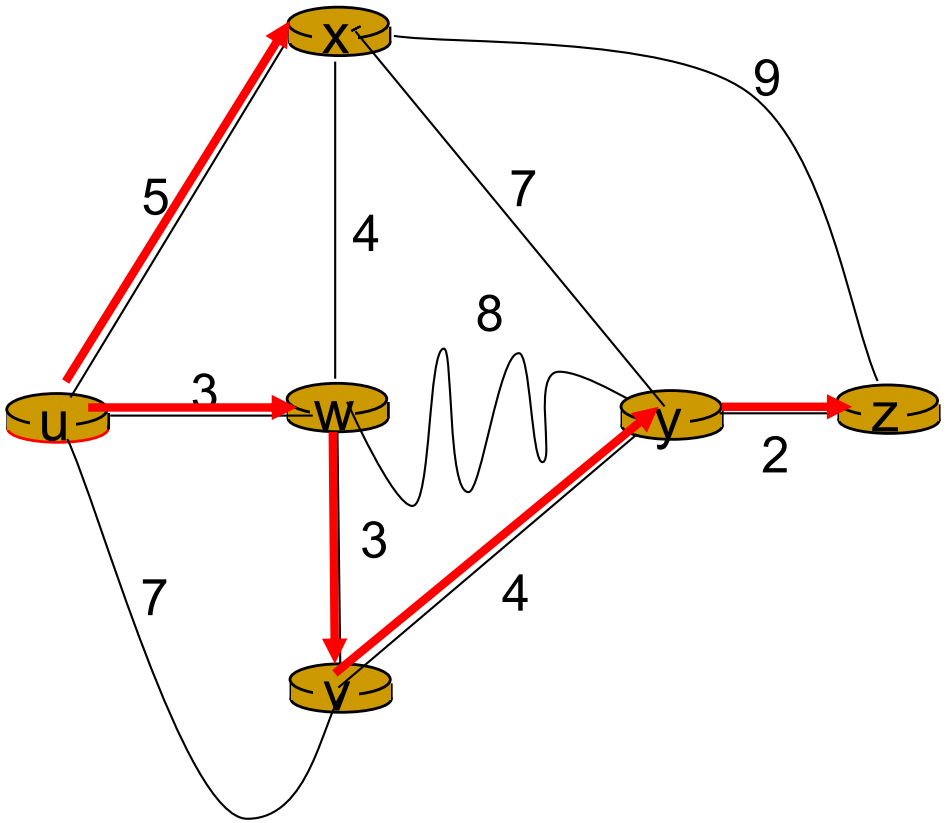
Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

notes:

- construct shortest path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)



u's forwarding Table



destination	Next hop
v	w
x	x
y	w
w	w
z	w

Algorithm Complexity

n nodes

- each iteration: need to check all nodes, w, not in N
- $n(n+1)/2$ comparisons: $O(n^2)$
- more efficient implementations possible: O(nlogn)

Bellman-Ford equation

Define

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

$$d_x(y) = \min_{v \in N(x)} \{c(x,v) + d_v(y)\}$$

$d_v(y)$: cost from neighbor v to destination y

$c(x,v)$: cost to neighbor v

$\min_{v \in N(x)}$: *min* taken over all neighbors v of x

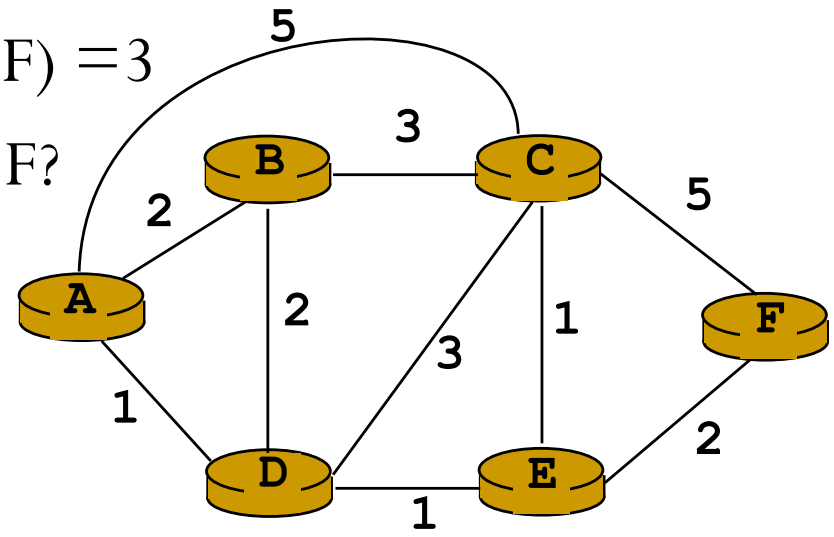
Bellman-Ford equation

Known, $d_B(F) = 5$, $d_D(F) = 3$, $d_C(F) = 3$

What is the shortest path from A to F?

Neighbors of A: B, C, D

$$d_A(F) = \min \{ c(A,B) + d_B(F),$$
$$c(A,D) + d_D(F),$$
$$c(A,C) + d_C(F),$$
$$= \min \{ 2 + 5,$$
$$1 + 3,$$
$$5 + 3\} = 4$$

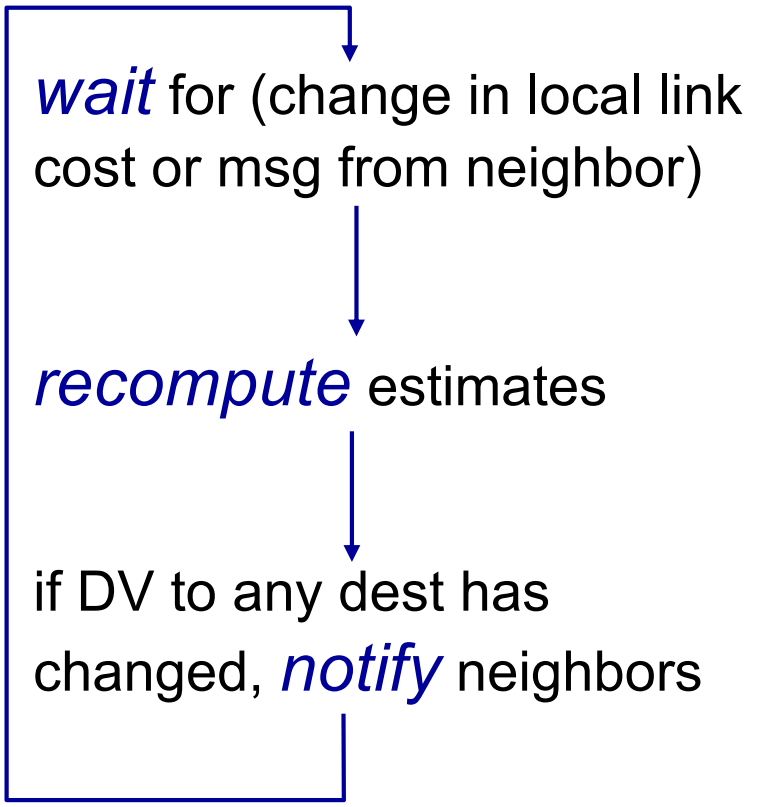


Distance Vector Algorithm

- Distance vector: a node's least-known costs to other nodes
- Each node periodically sends its own distance vector estimate to neighbors *only* when its DV changes
- when x receives a new DV estimate from a neighbor, it updates its own DV using the Bellman equation:

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

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

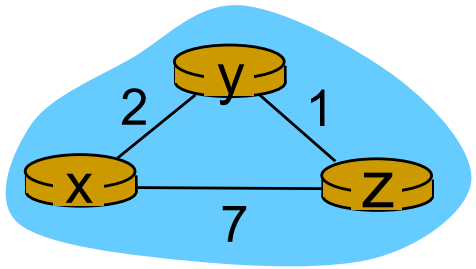
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

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

node x table

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

node y table

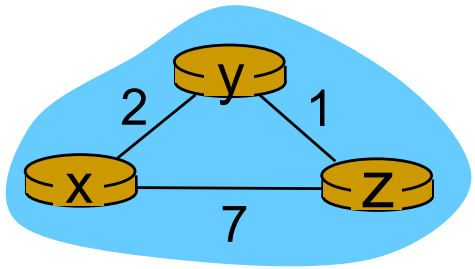
from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

node z table

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

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

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



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

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

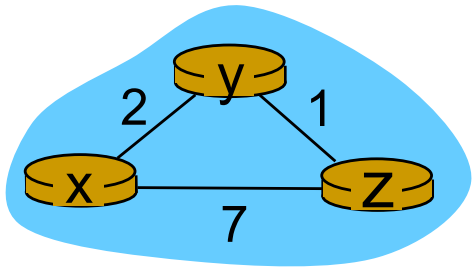
from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

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



time

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

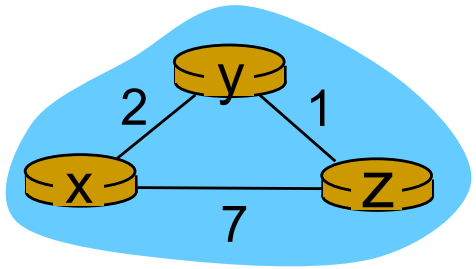
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

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

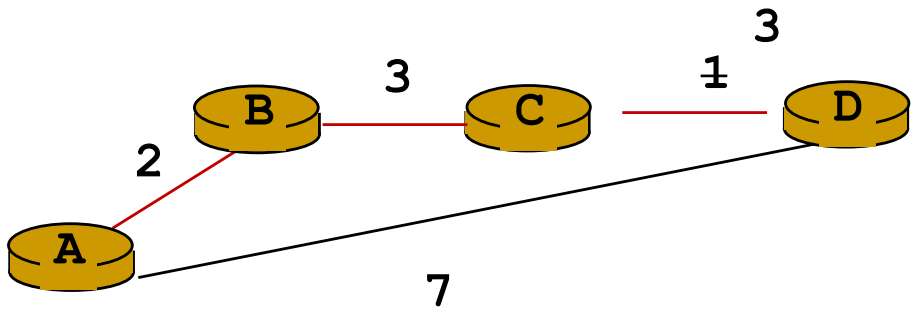
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0



time

Complexity

- How many messages per round?
  - O(|E|) with O(|V|) computation per each node
- How many rounds in the worst case?
  - O(|V|)



Routing Protocol in the Internet

Internet routing protocols

- responsible for constructing and updating the forwarding tables at routers

scale: with up to 4 billion destinations:

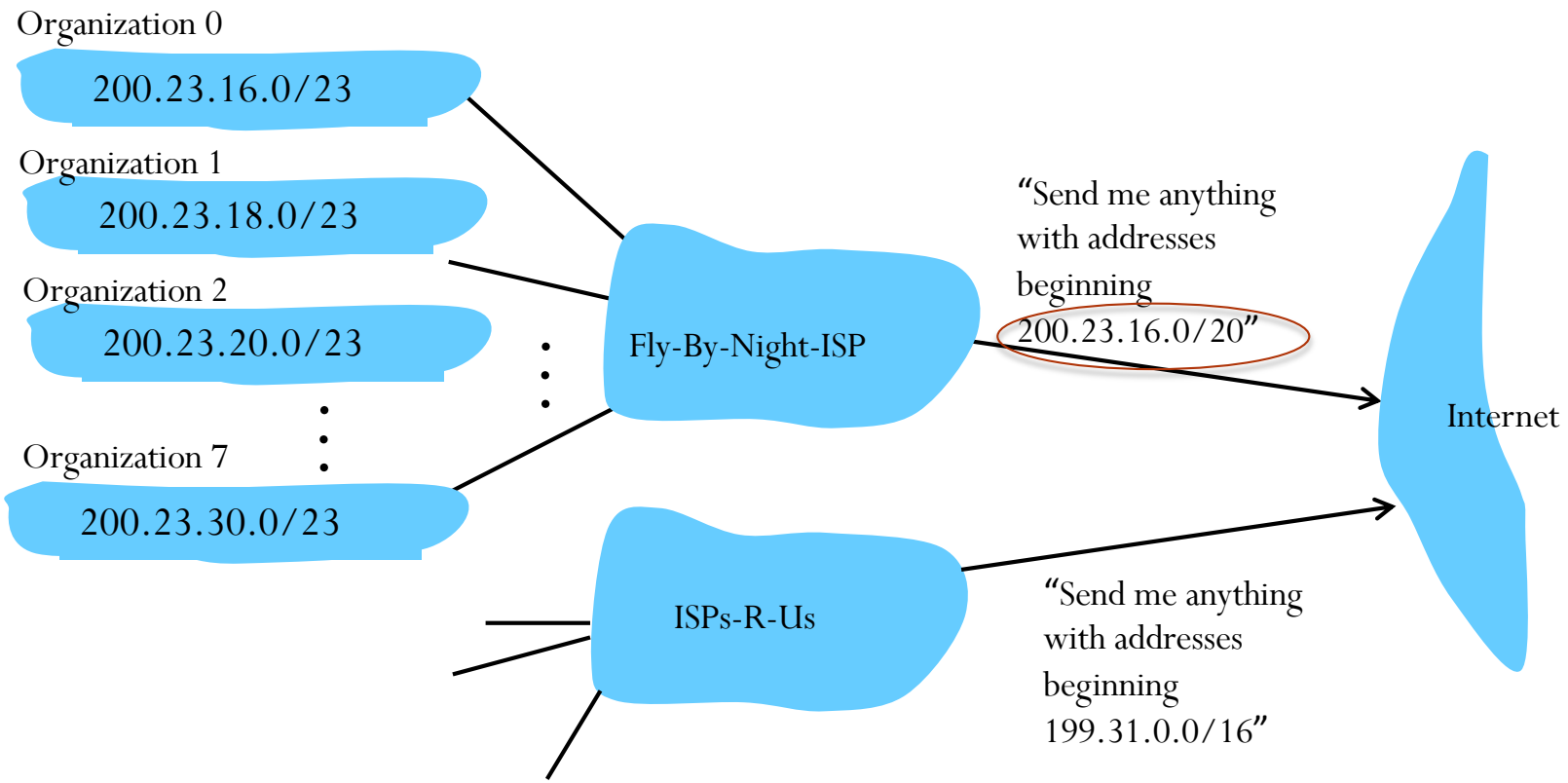
- can't store all destination addresses in forwarding tables!
- forwarding table exchange would swamp links!

administrative autonomy

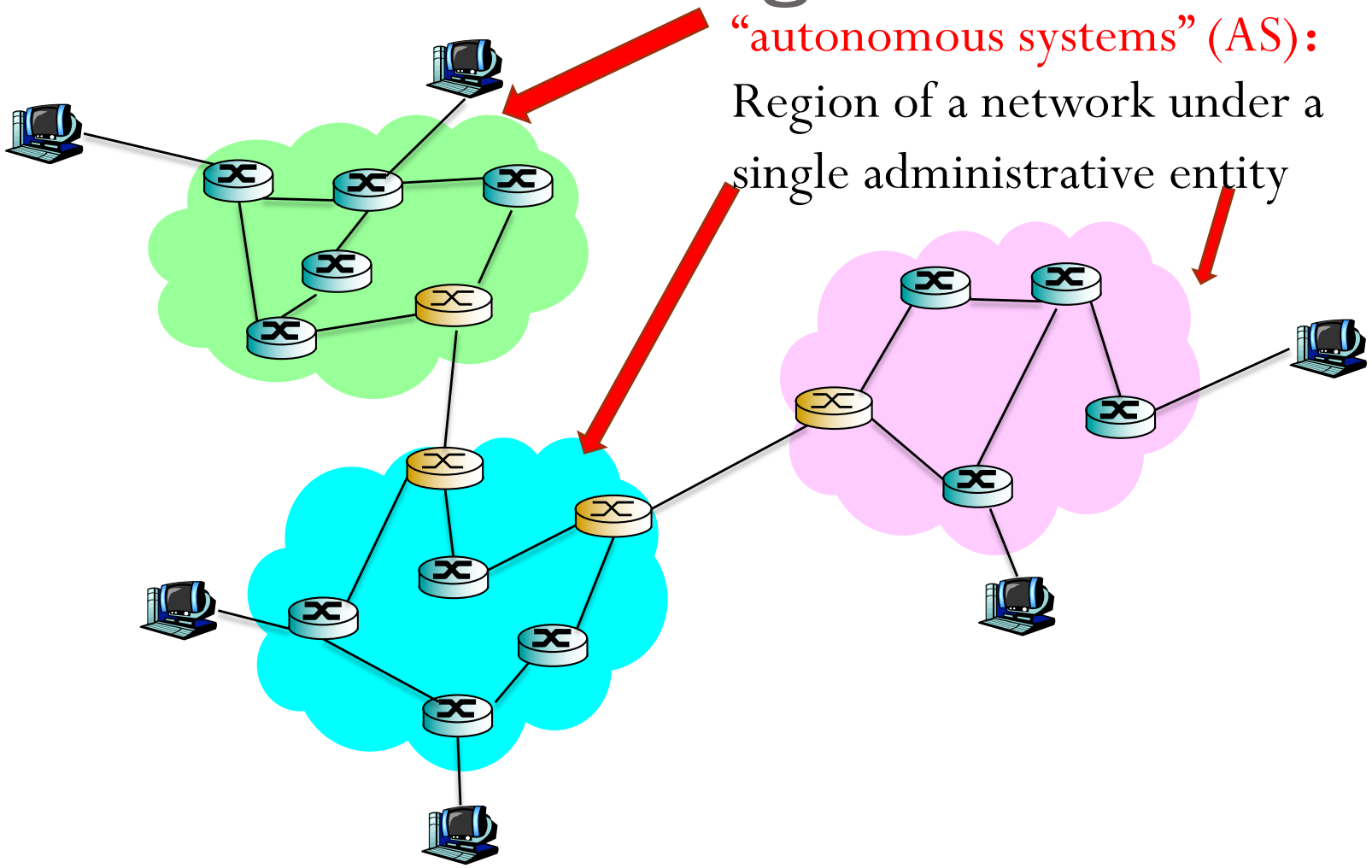
- internet = network of networks
- each network admin may want to control routing in its own network

Hierarchical addressing: route aggregation

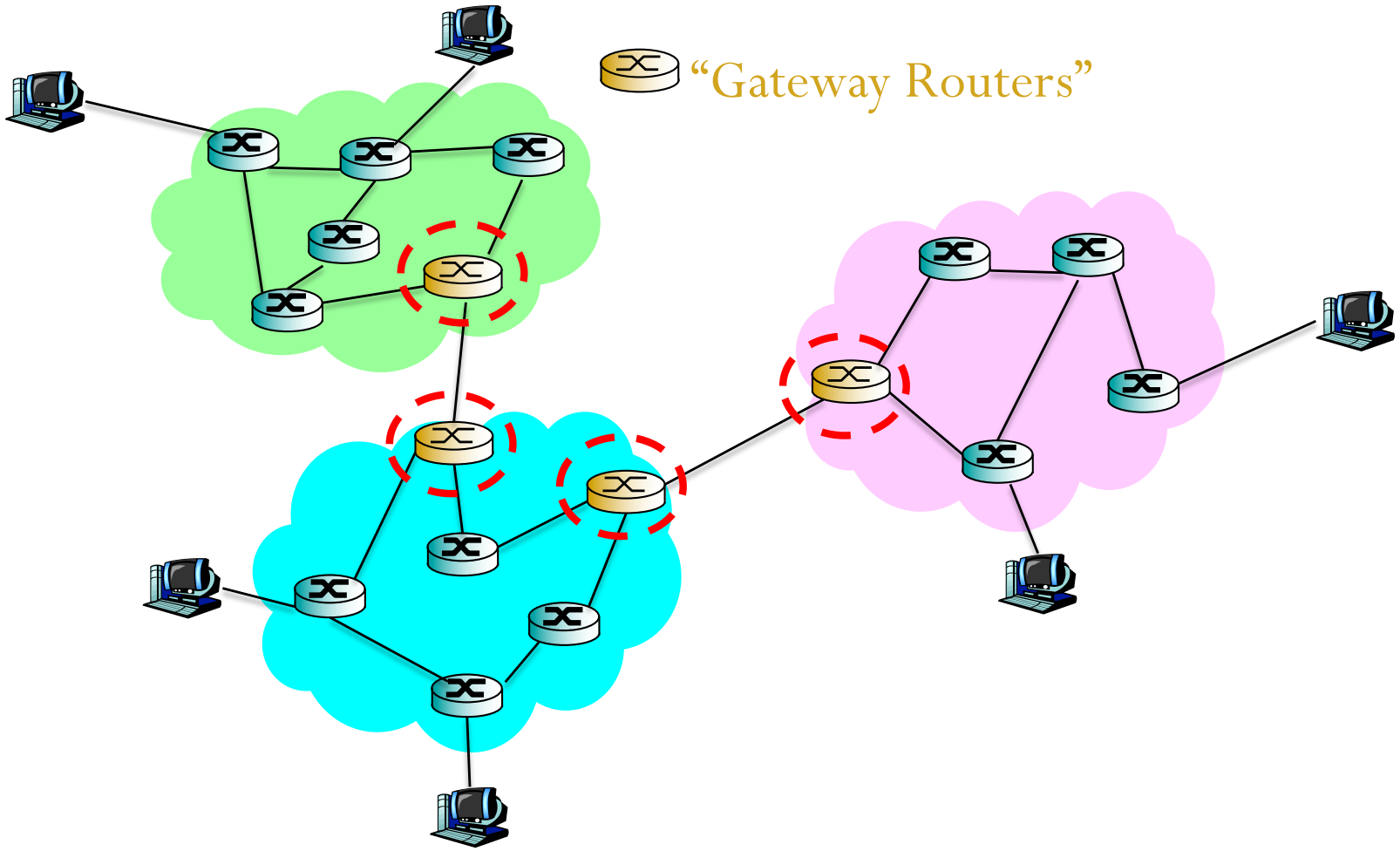
Hierarchical addressing allows efficient advertisement of routing information:



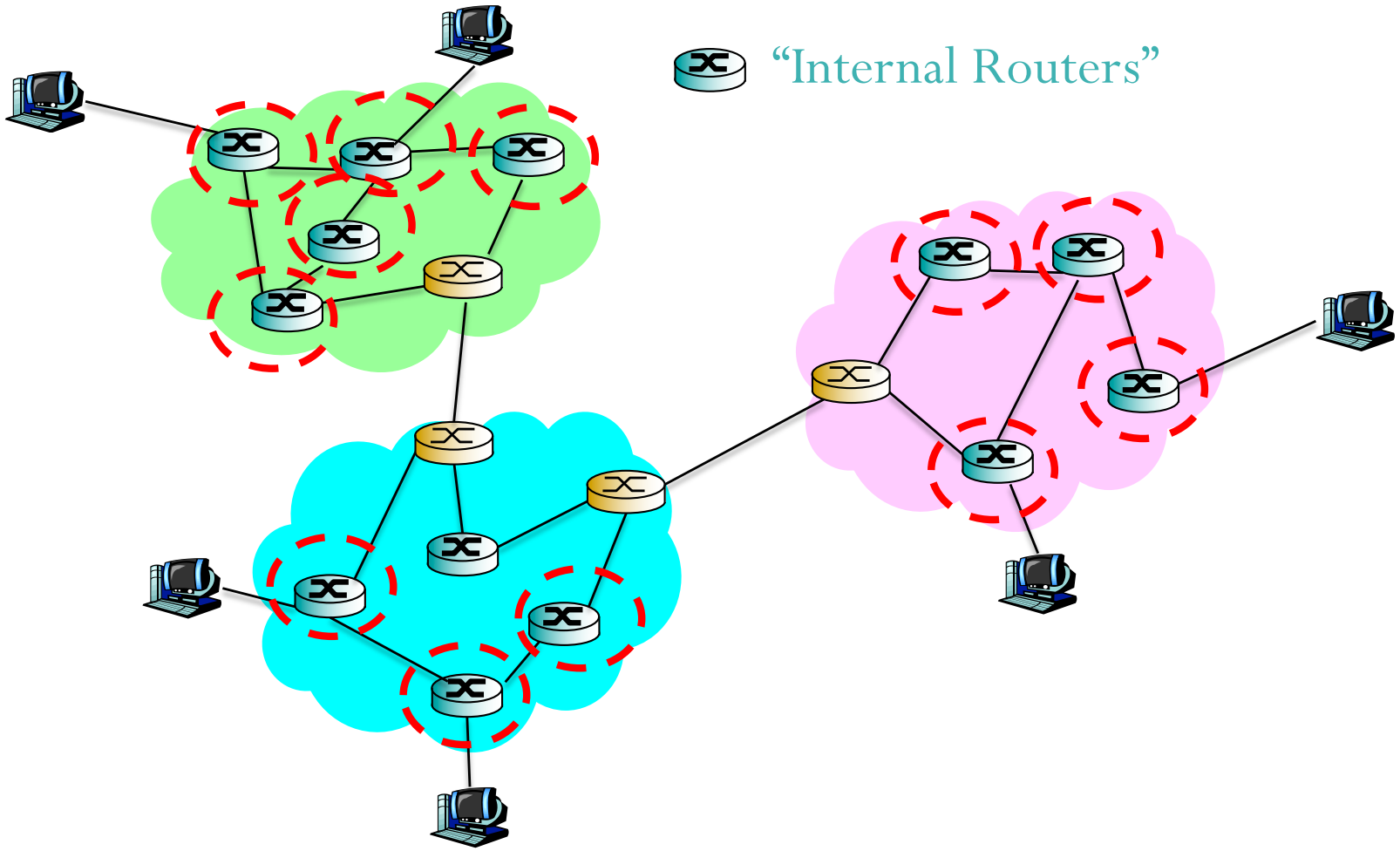
Hierarchical Routing



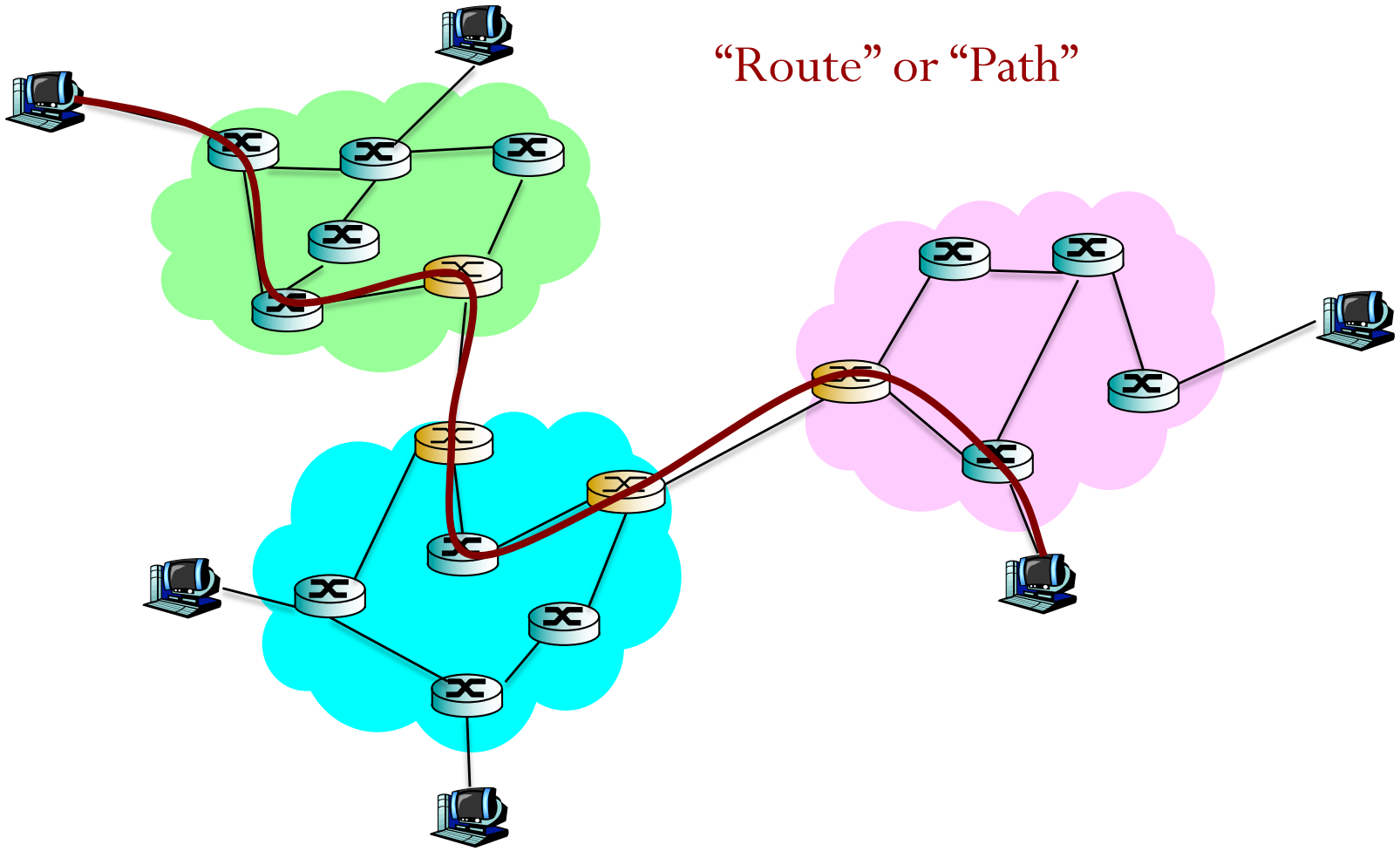
Hierarchical Routing



Hierarchical Routing



Hierarchical Routing



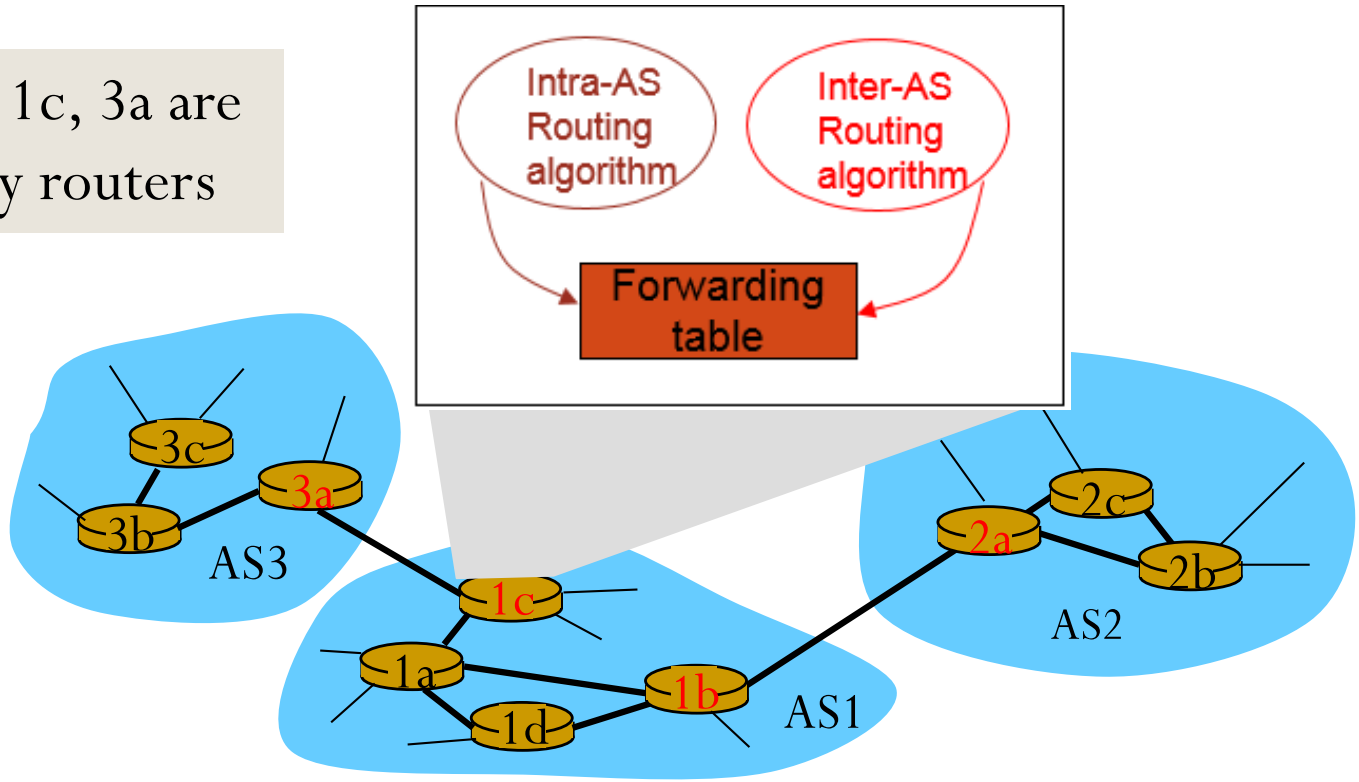
Hierarchical Routing

- Internet Routing works at two levels
- Each AS runs an intra-domain routing protocol that establishes routes within its domain
  - (AS -- region of network under a single administrative entity)
  - Link State, e.g., Open Shortest Path First (OSPF)
  - Distance Vector, e.g., Routing Information Protocol (RIP)
- ASs participate in an inter-domain routing protocol that establishes routes between domains
  - Path Vector, e.g., Border Gateway Protocol (BGP)

Interconnected ASes

- Forwarding table is 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

1b, 2a, 1c, 3a are gateway routers



Inter-AS tasks

- Suppose router 1d in AS1 receives datagram for which dest is outside of AS1 (e.g. X)
  - Router should forward packet towards one of the gateway routers, but which one?

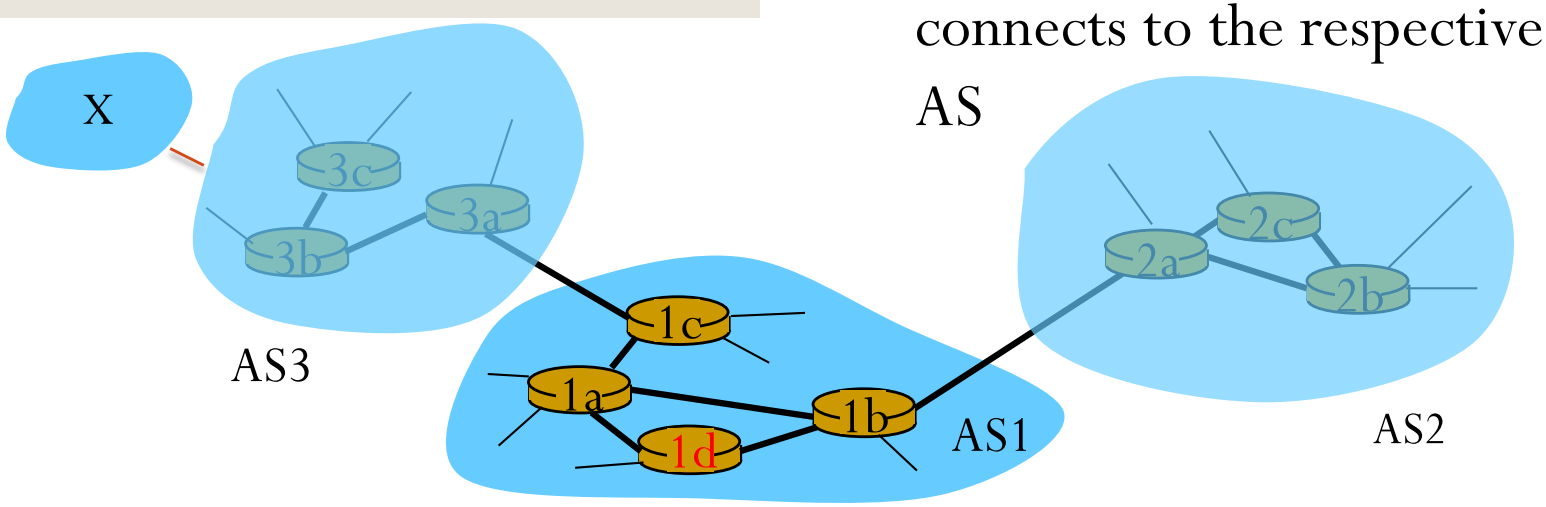
AS1 needs:

1. to learn which dests are reachable through AS2 and which through AS3
2. to propagate this reachability info to all routers in AS1

Job of inter-AS routing!

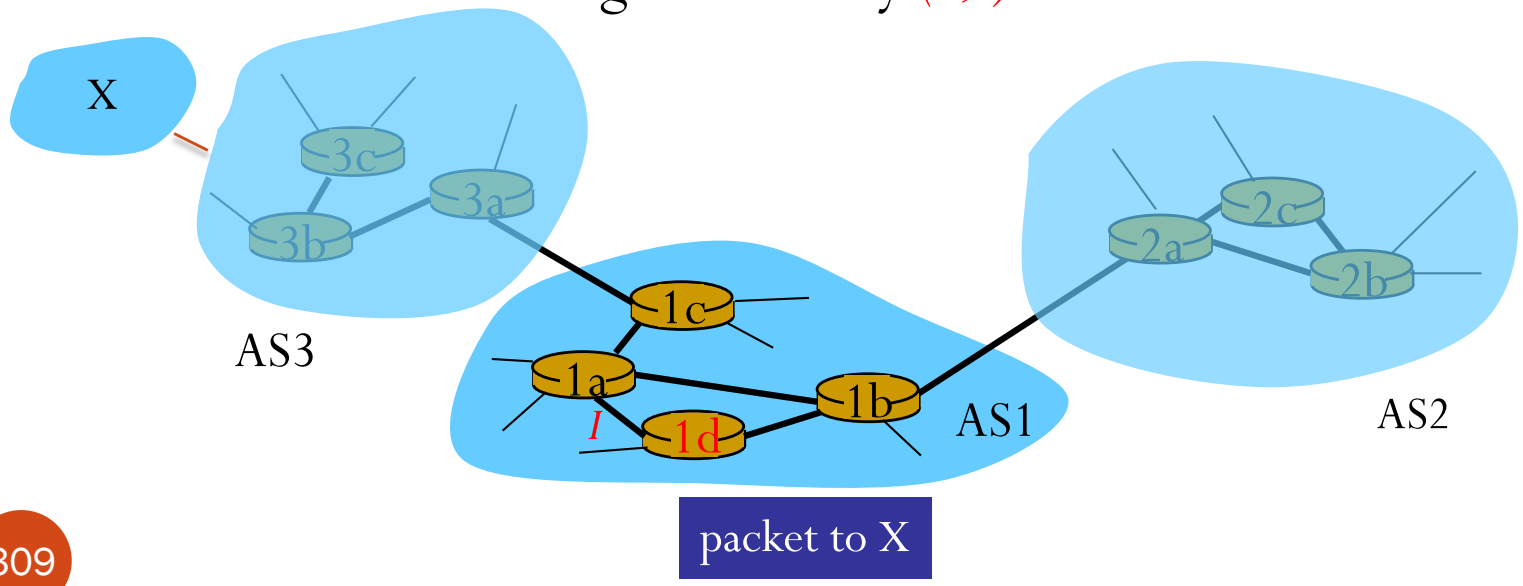
3. which gateway router connects to the respective AS

Typically static



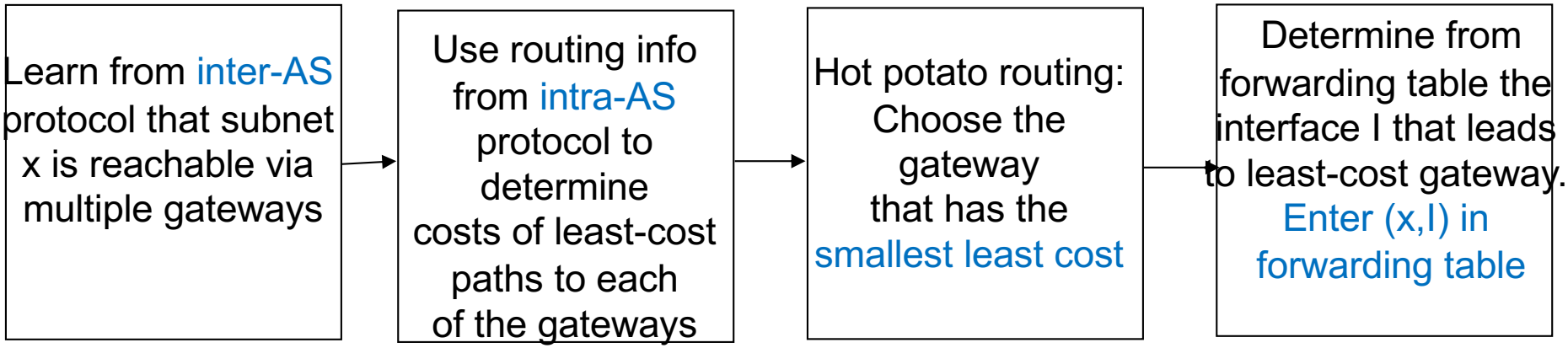
Example: Setting forwarding table in router 1d

1. Suppose AS1 learns from the inter-AS protocol that subnet *x* is reachable from AS3 (gateway 1c) but not from AS2.
2. Inter-AS protocol propagates reachability info to all internal routers.
3. Router 1d determines from intra-AS routing that its interface *I* is on the least cost path to 1c.
4. Puts in forwarding table entry *(x,I)*.



Example: Choosing among multiple ASes

- Now suppose AS1 learns from the inter-AS protocol that subnet *x* is reachable from AS3 *and* from AS2.
- To configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest x.
- This is also the job on inter-AS routing protocol!
- Hot potato routing: send packet towards closest of two routers.



Summary: all routers (NOT JUST gateway routers) need run both intra- and inter- domain routing protocols

Intra-AS Routing: IGP

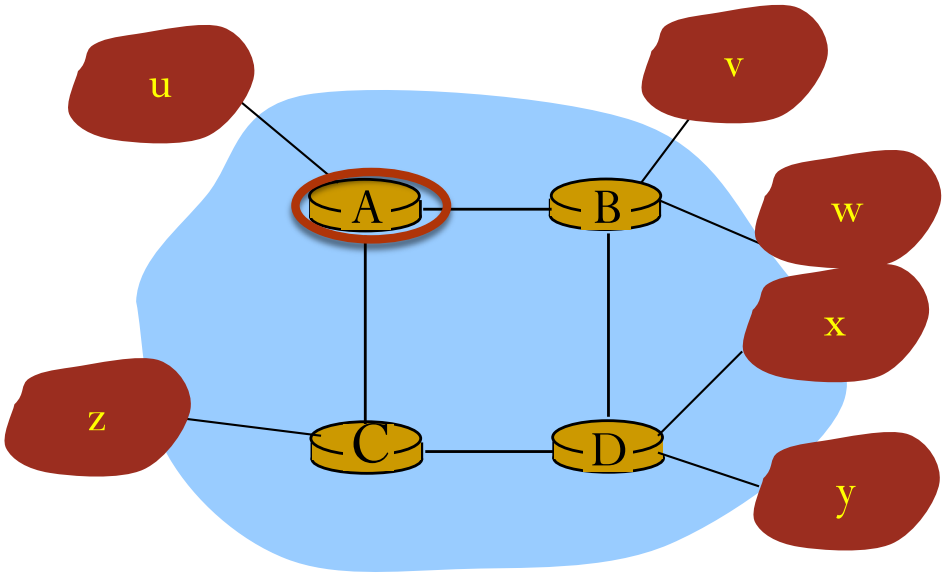
- IGP: "Interior Gateway Protocol" = Intra-domain routing protocol
  - provide internal reachability

- Most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

A hop is one link on a path between two adjacent routers

RIP (Routing Information Protocol)

- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops, unreachable/∞=16)



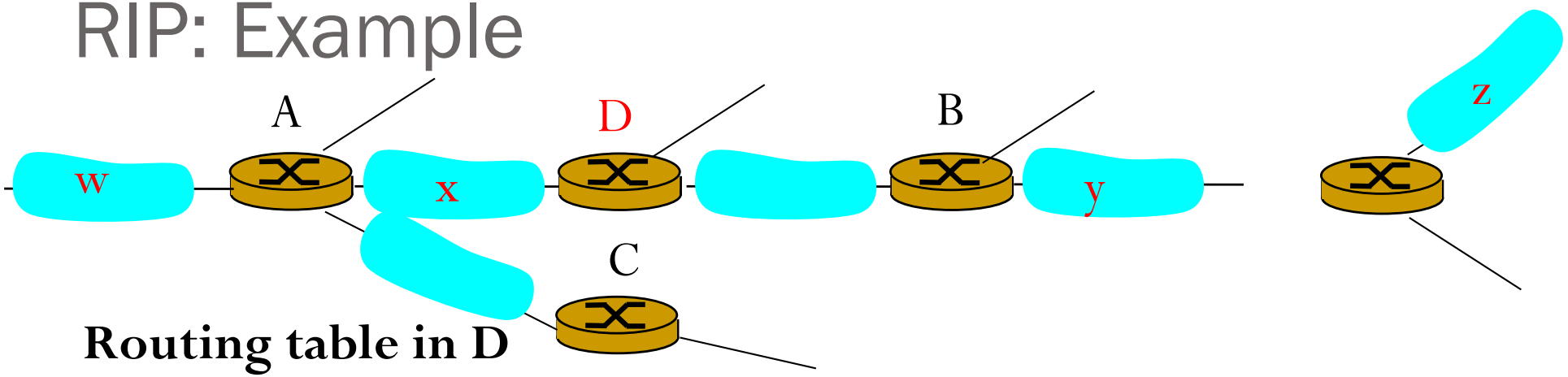
From router A to subnets:

destination	hop counts
u	1
v	2
w	2
x	3
y	3
z	2

RIP advertisements

- Distance vectors: exchanged among neighbors every 30 sec via Response Message (also called advertisement)
- Each advertisement: list of up to 25 destination nets and the hop counts within AS (no predecessor information!)

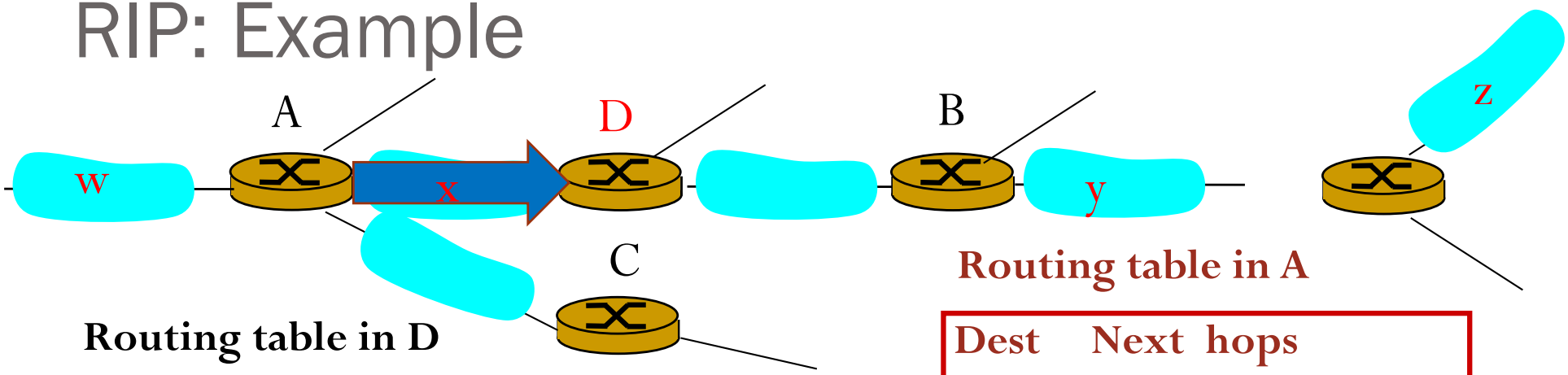
RIP: Example



Routing table in D

Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

RIP: Example



Routing table in D

Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B	7
x	--	1
....	....	....

Routing table in A

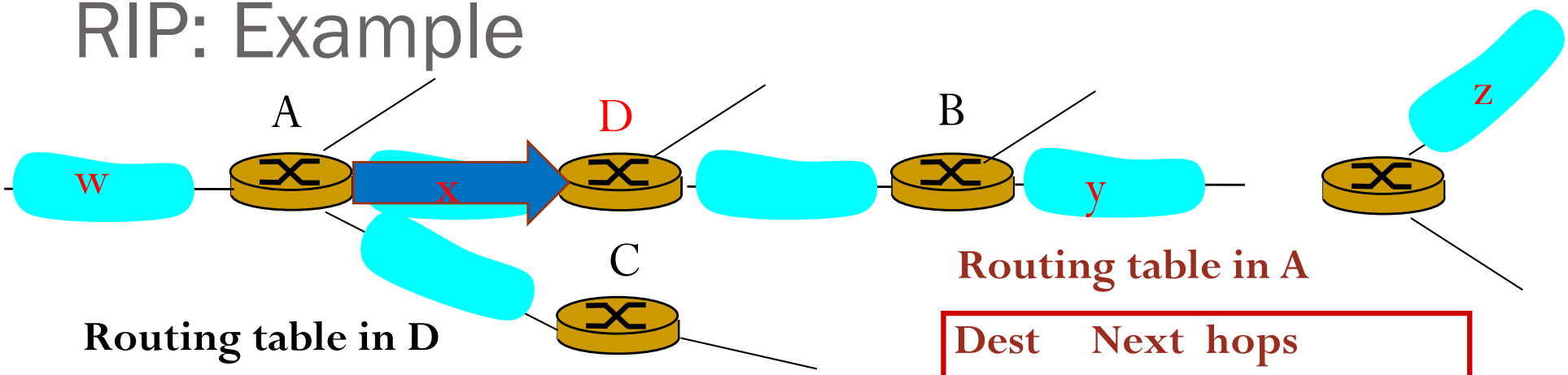
Dest	Next	hops
w	-	1
x	-	1
z	C	4
....	...	...

Advertisement from A to D

Dest	hops
w	1
x	1
z	4
....	...

Q: what changes?

RIP: Example



Routing table in D

Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

Routing table in A

Dest	Next	hops
w	-	1
x	-	1
z	C	4
….	…	…

Advertisement from A to D

Dest	hops
w	1
x	1
z	4
….	…

Q: what changes?

D to z though A: 4 + 1 = 5

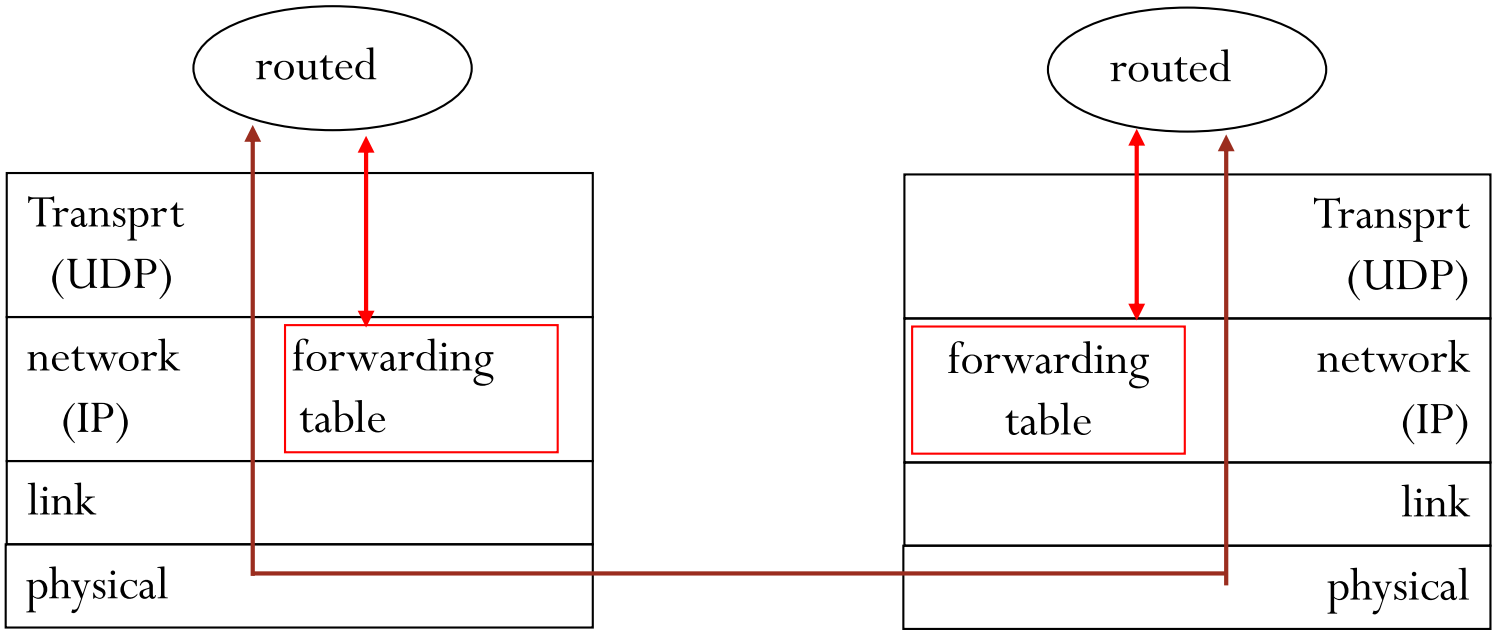
RIP: Link Failure and Recovery

If no advertisement heard after 180 sec → neighbor/link declared dead

- routes via neighbor invalidated
- new advertisements sent to neighbors
- neighbors in turn send out new advertisements (if tables changed)
- link failure info quickly propagates to entire net

RIP Table processing

- RIP routing tables managed by **application-level** process called route-d (daemon)
- advertisements sent in UDP packets, periodically repeated



OSPF (Open Shortest Path First)

- "open": publicly available
- Uses Link State algorithm
  - LS dissemination
  - Topology map at each node
  - Route computation using Dijkstra's algorithm

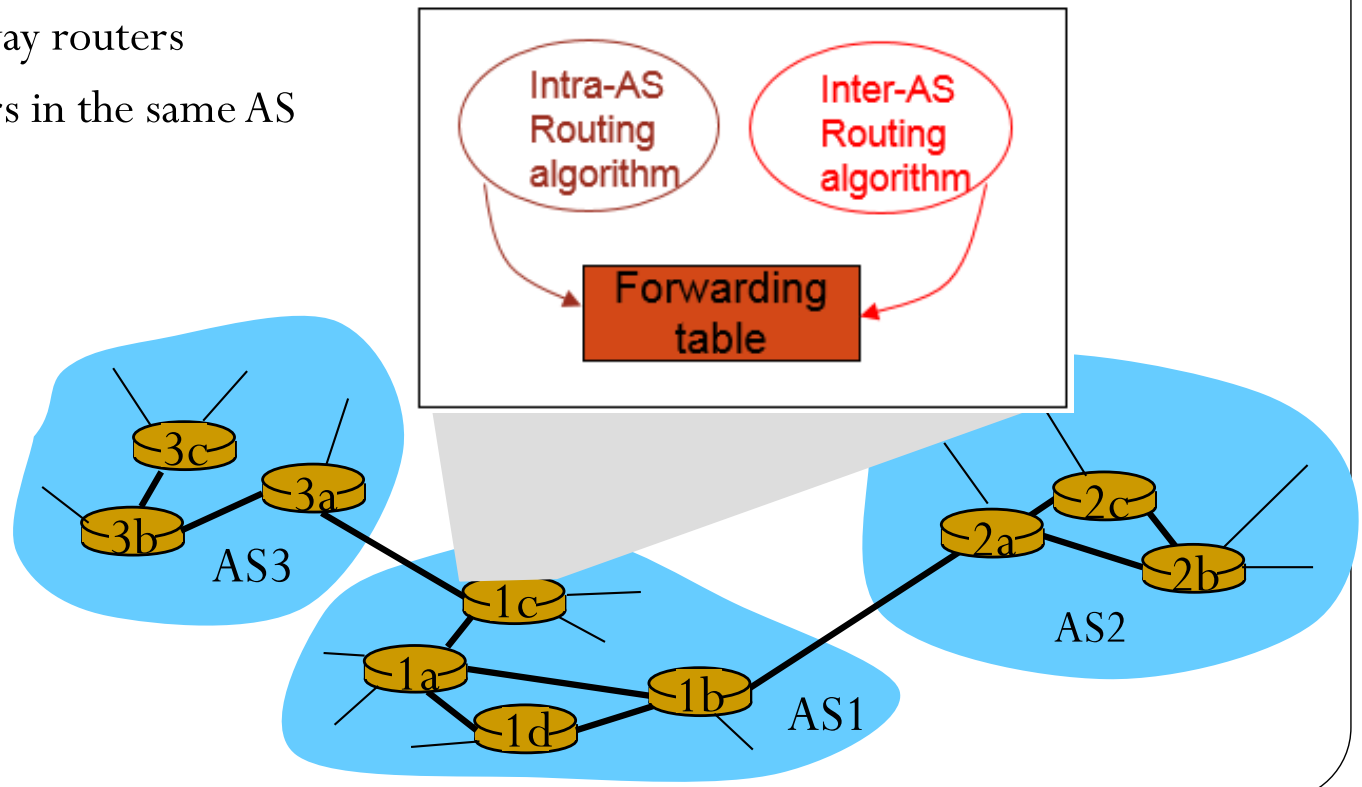
- OSPF advertisement carries one entry per neighbor router
- Advertisements disseminated to entire AS (via flooding)
  - Carried in OSPF messages directly over IP (rather than TCP or UDP

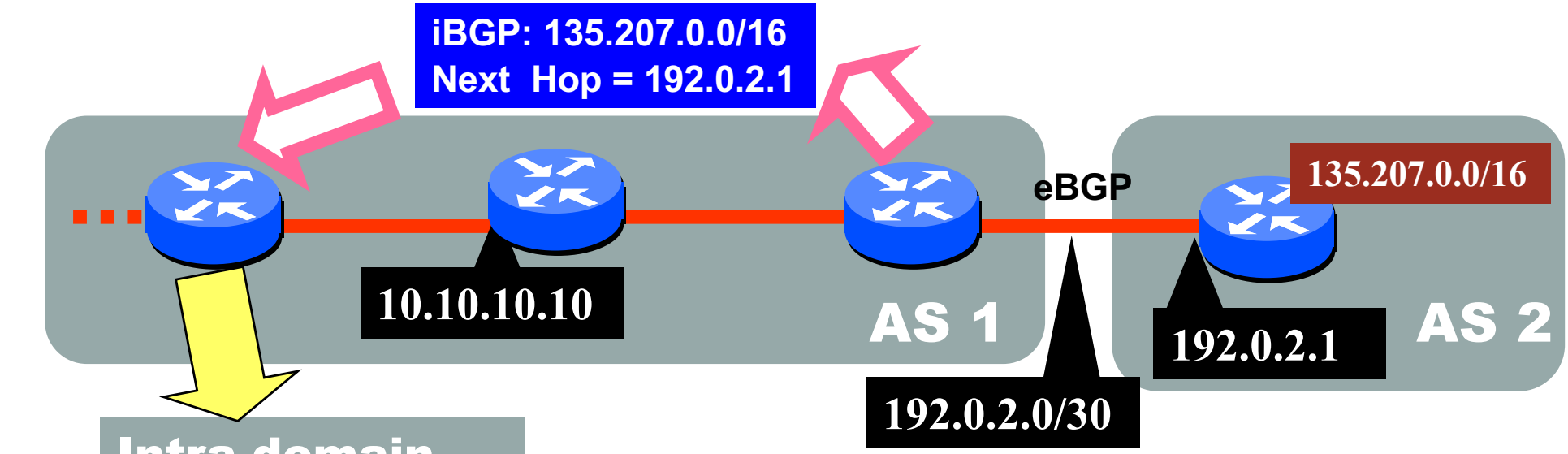
OSPF "advanced" features (not in RIP)

- Security: all OSPF messages authenticated (to prevent malicious intrusion)
- Multiple same-cost paths allowed (only one path in RIP)
- For each link, multiple cost metrics for different TOS (e.g., satellite link cost set "low" for best effort; high for real time)
- Integrated uni- and multicast support:
  - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- Hierarchical OSPF in large domains.

Inter-domain routing protocol

- Forwarding table is 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
- Border Gateway Protocol (BGP): use AS path vector
  - eBGP between gateway routers
  - iBGP between routers in the same AS
  - Policy driven





Intra-domain

destination	next hop
192.0.2.0/30	10.10.10.10

+

iBGP

destination	next hop
135.207.0.0/16	192.0.2.1

Forwarding Table

destination	next hop
135.207.0.0/16	10.10.10.10
192.0.2.0/30	10.10.10.10

Summary of difference between Intra- and Inter-AS routing

	Policy	Scale
Intra-domain routing	•Routing metric	Internal to ASs
Inter-domain routing	• control over how its traffic routed • who routes through its net.	• prefix aggregation • use AS in path attributes • iBGP to disseminate AS NEXT-HOP