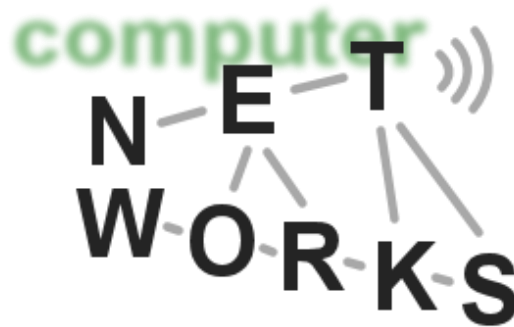# Network Layer – Part I

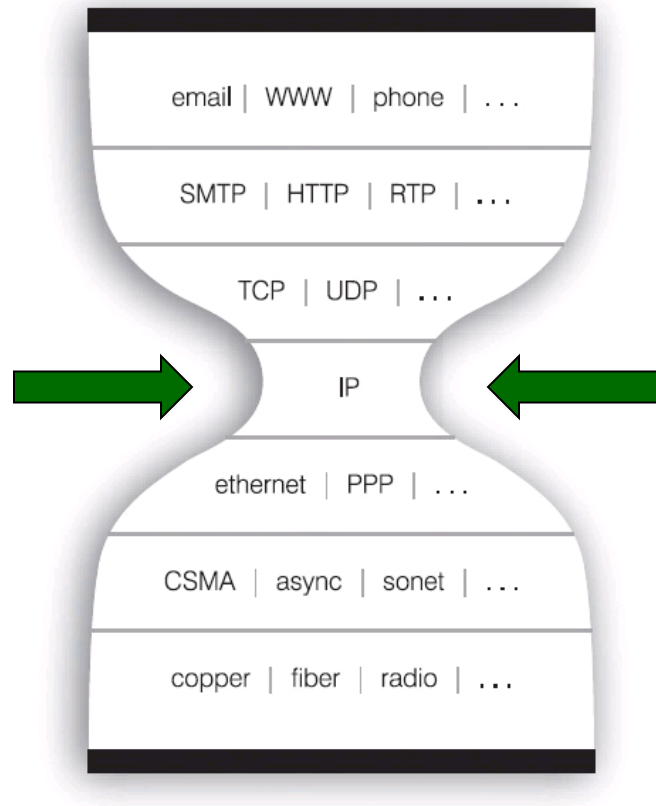## Computer Networks, Winter 2011/2012

Acknowledgements: J. Kurose & K. Ross – Computer Networking: A Top-Down Approach Featuring the Internet

Network Layer

o 4.1 Introduction

o 4.2 What's inside a router

o 4.3 IP: Internet Protocol

   o Datagram format

   o IPv4 addressing

   o ICMP

   o IPv6

"Hourglass" architecture

© Jonathan L. Zittrain (http://yupnet.org/zittrain/archives/13)

Network layer
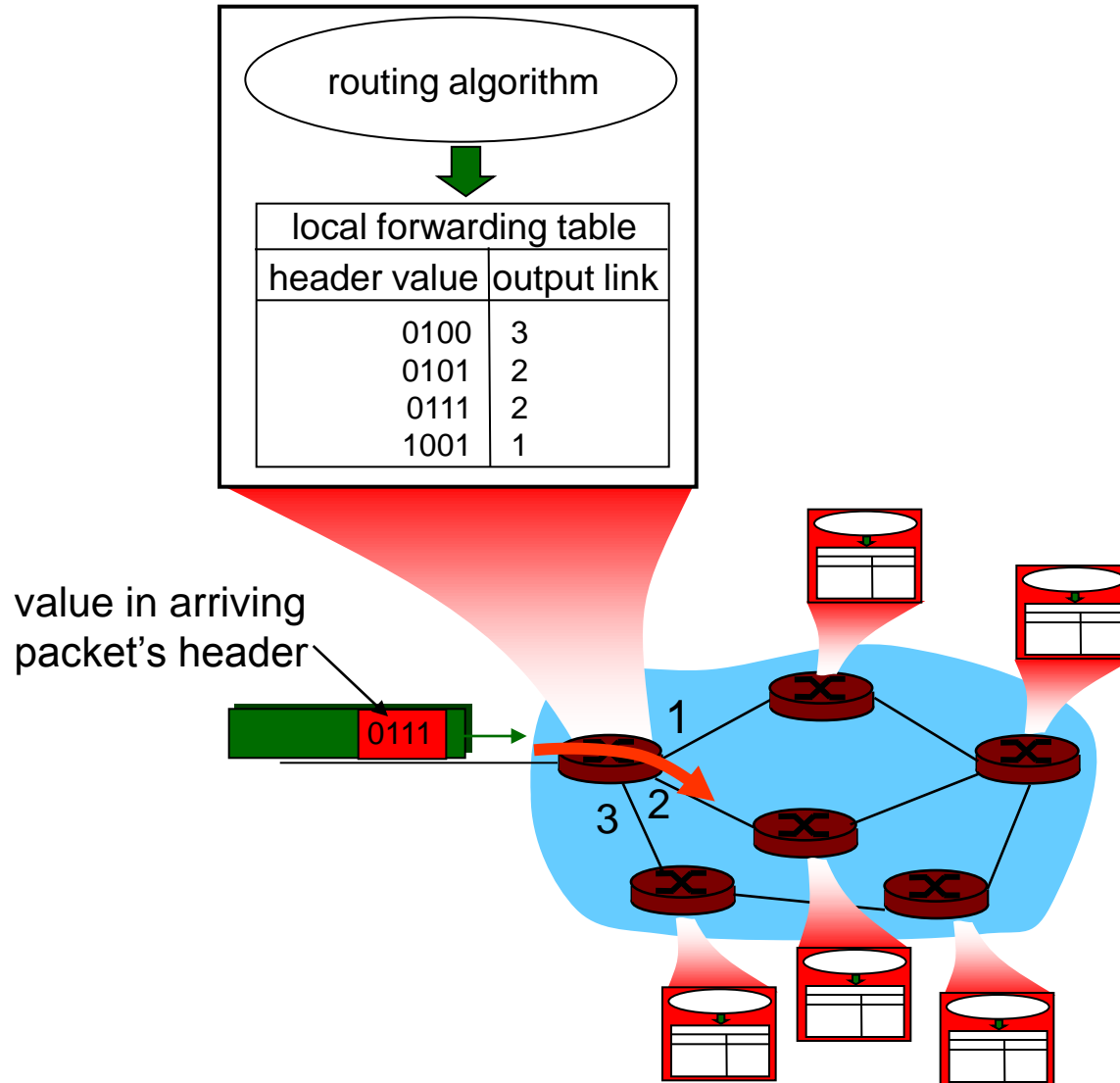
o Transports segments from sending to receiving host

o Sending side: encapsulates segments into datagrams

o Receiving side: delivers segments to transport layer

o Network layer protocols in *every* host, router

o Router examines header fields in all IP datagrams passing through it
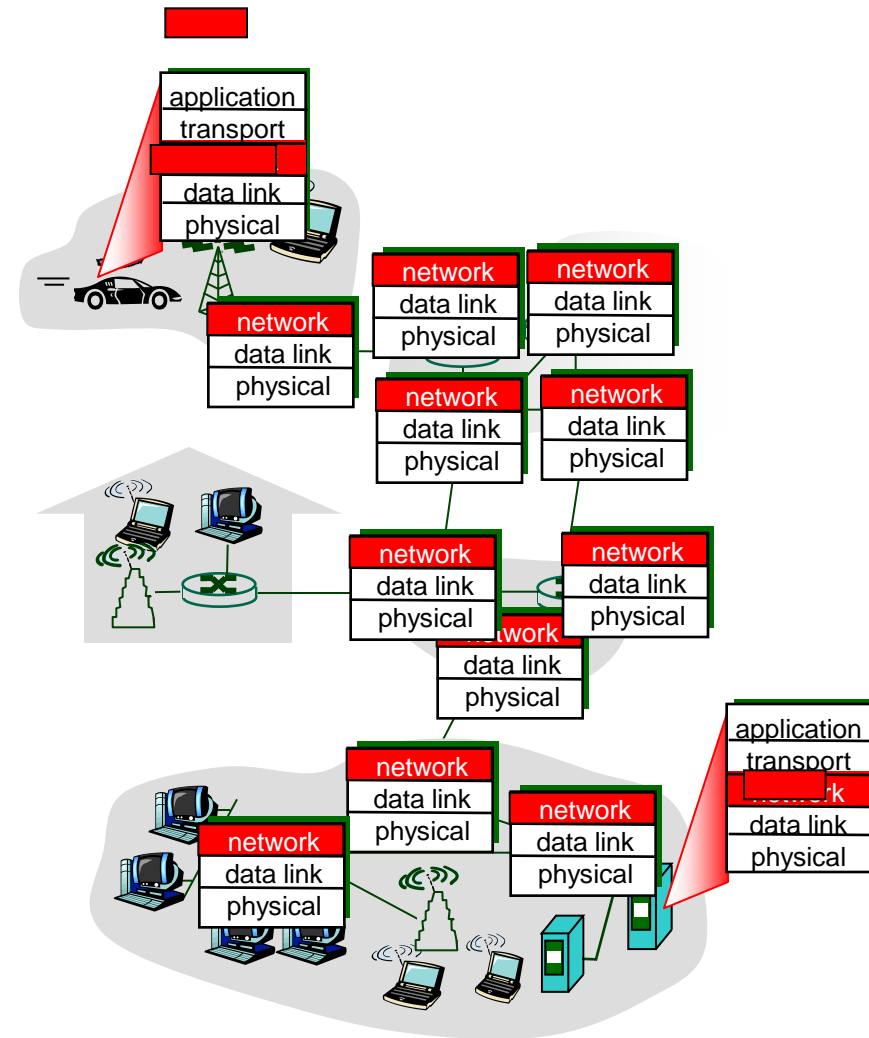
Two Key Network-Layer Functions

o Forwarding: move packets from router's input to appropriate router output

o Routing: determine route taken by packets from source to dest.

    o routing algorithms

o Analogy:

o Routing: process of planning trip from source to dest

o Forwarding: process of getting through single interchange

Interplay between routing and forwarding

routing algorithm

local forwarding table

header value	output link
0100	3
0101	2
0111	2
1001	1

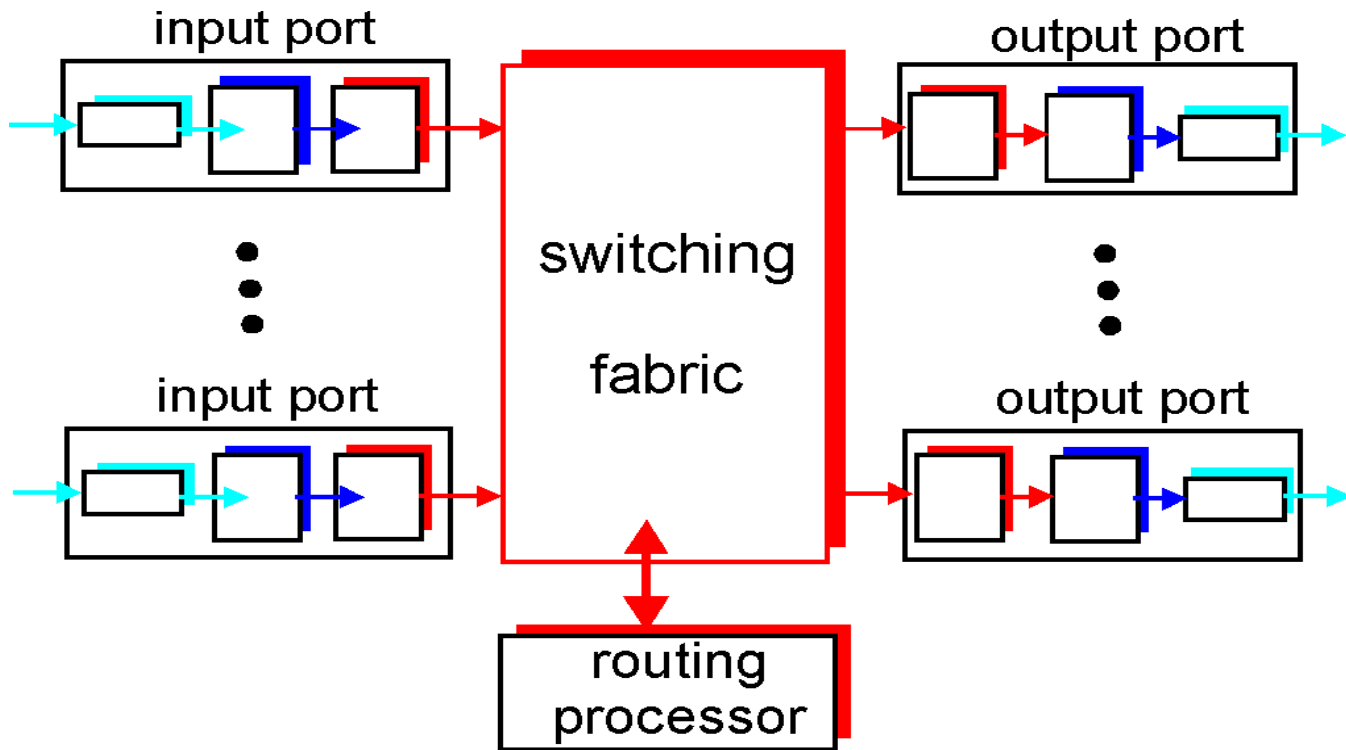value in arriving packet's header

0111

1
2
3

Network Layer

- 4.1 Introduction
- 4.2 What's inside a router
- 4.3 IP: Internet Protocol
  - Datagram format
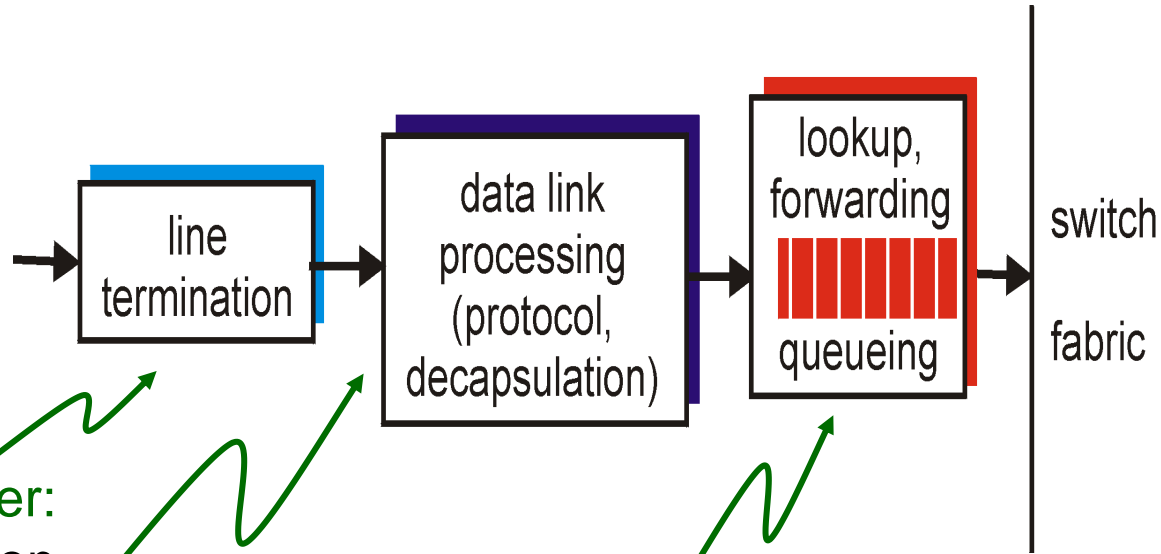 - IPv4 addressing
  - ICMP
  - IPv6

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

○ goal: complete input port processing at 'line speed'

○ queuing: if datagrams arrive faster than forwarding rate into switch fabric

Three types of switching fabrics

memory

bus

crossbar
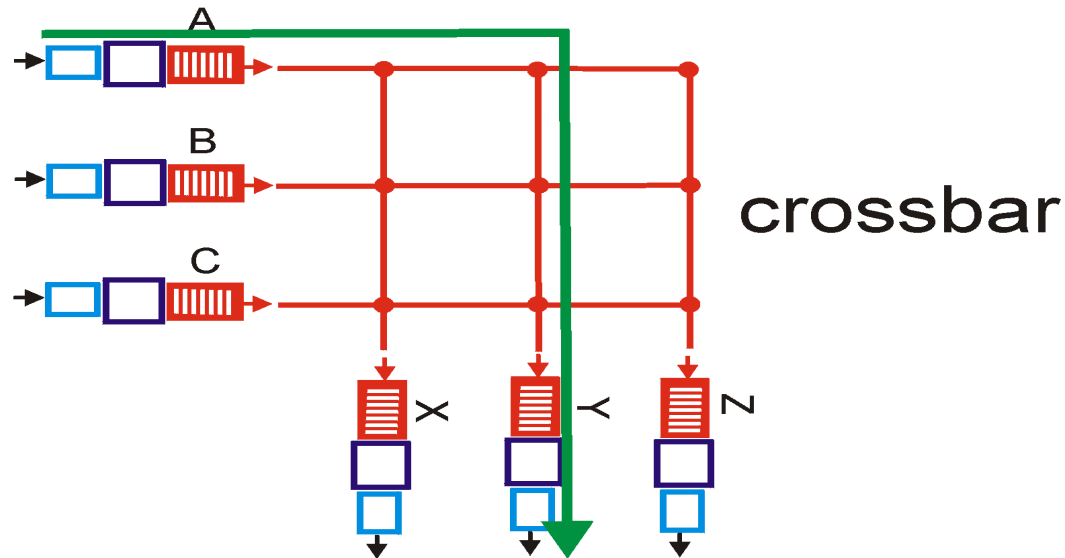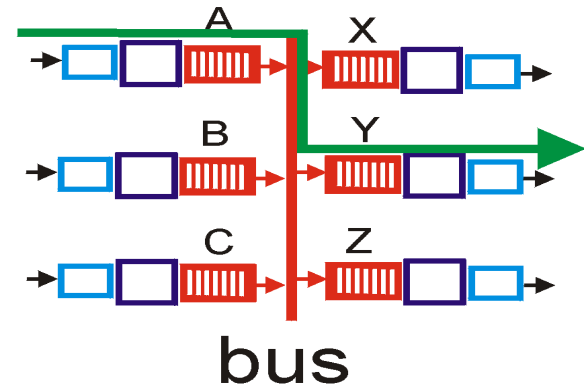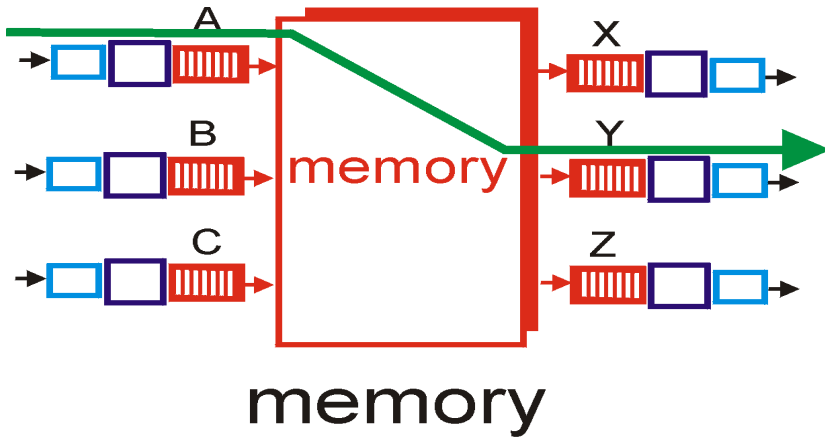
Switching Via Memory

First generation routers:

o traditional computers with switching under direct control of CPU

o packet copied to system's memory

o speed limited by memory bandwidth (2 bus crossings per datagram)

Input Port   Memory   Output Port

System Bus
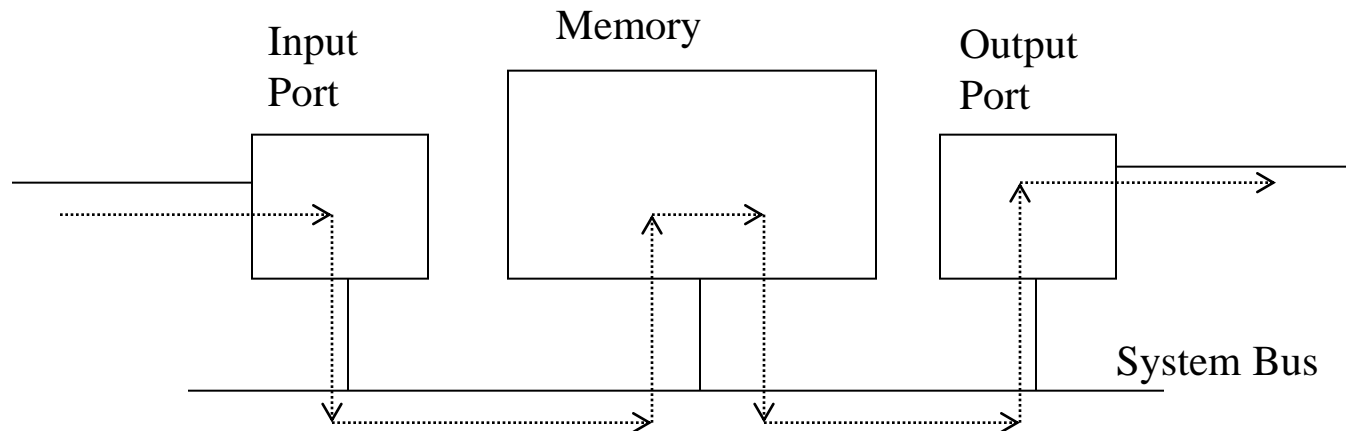
Switching Via a Bus

○ datagram from input port memory

○   to output port memory via a shared bus

○ bus contention: switching speed limited by bus bandwidth

○ 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers

Switching Via An Interconnection Network

o overcome bus bandwidth limitations

o Banyan networks, other interconnection nets initially developed to connect processors in multiprocessor

o advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.

o Cisco 12000: switches 60 Gbps through the interconnection network

Output Ports
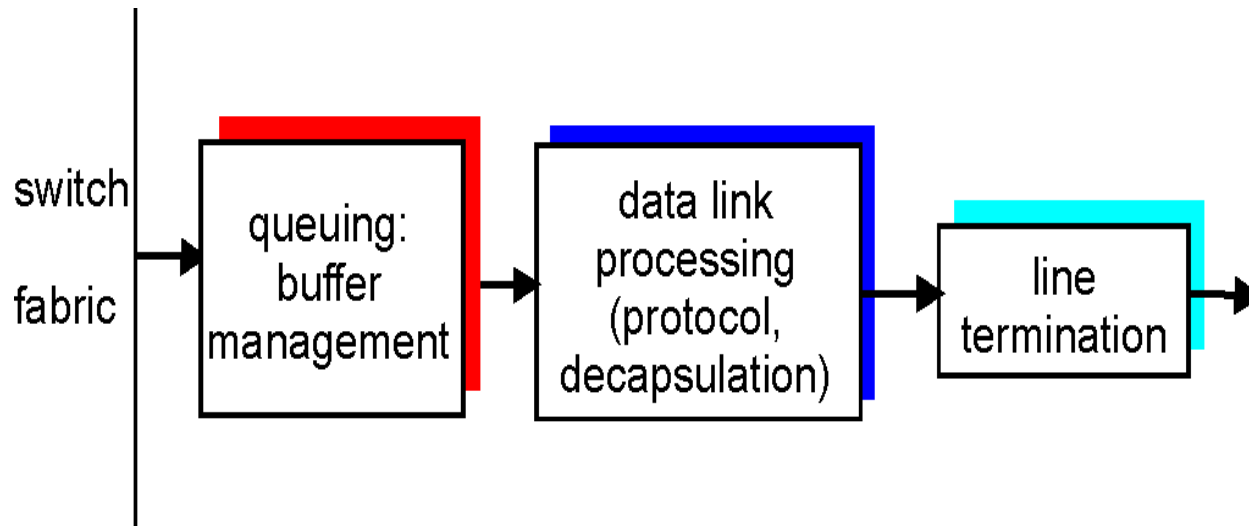
- *Buffering* required when datagrams arrive from fabric faster than the transmission rate
- *Scheduling discipline* chooses among queued datagrams for transmission

Output port queueing

Output Port Contention at Time *t*

One Packet Time Later
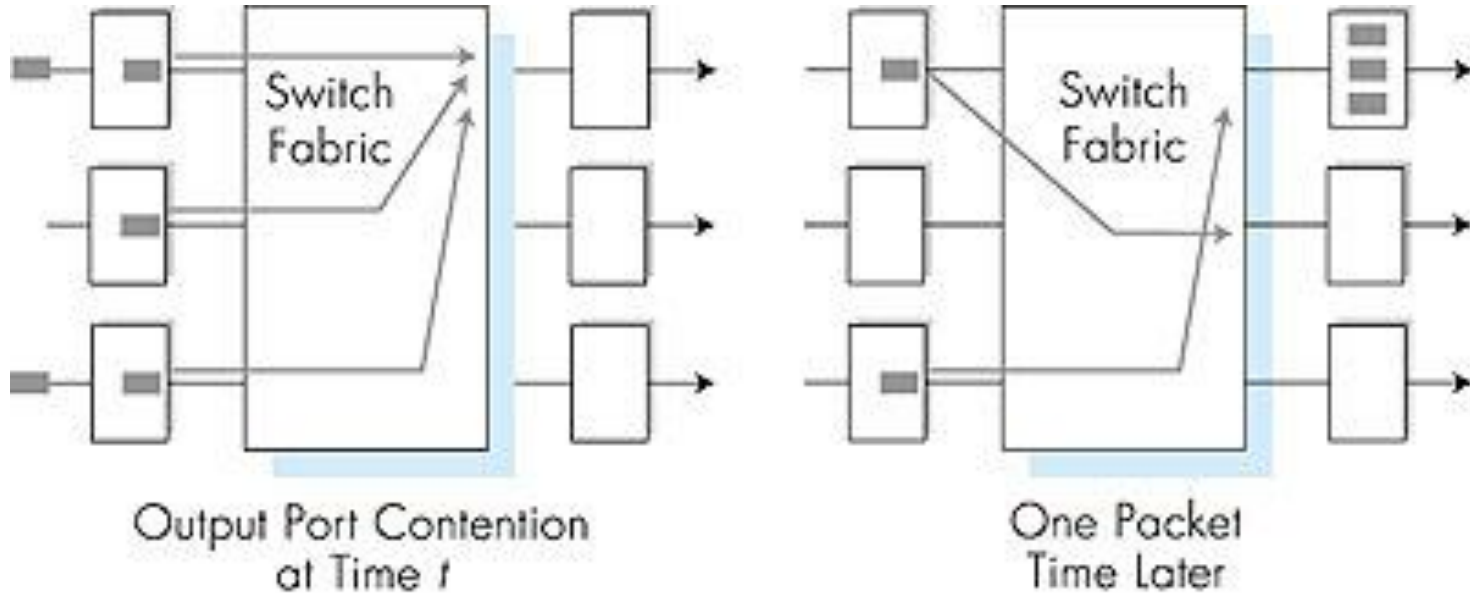
○ buffering when arrival rate via switch exceeds output line speed

○ *queueing (delay) and loss due to output port buffer overflow!*

How much buffering?

o RFC 3439 rule of thumb: average buffering equal to "typical" RTT (say 250 msec) times link capacity C

   o e.g., C = 10 Gps link: 2.5 Gbit buffer

o Recent recommendation: with *N* flows, buffering equal to

$$\frac{\text{RTT} \cdot \text{C}}{\sqrt{\text{N}}}$$

Input Port Queuing

○ Fabric slower than input ports combined -> queueing may occur at input queues

○ Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward

○ *queueing delay and loss due to input buffer overflow!*
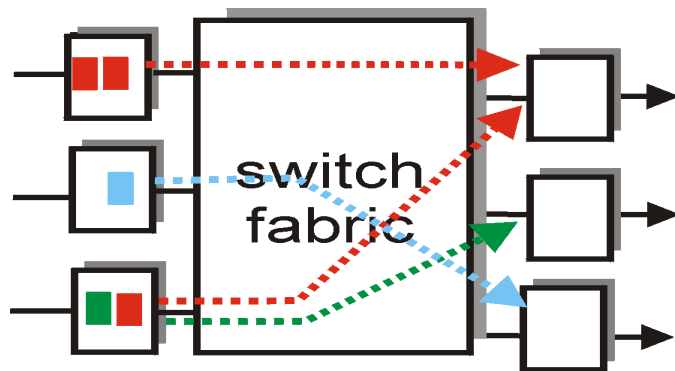
output port contention
at time t – only one red
packet can be transferred
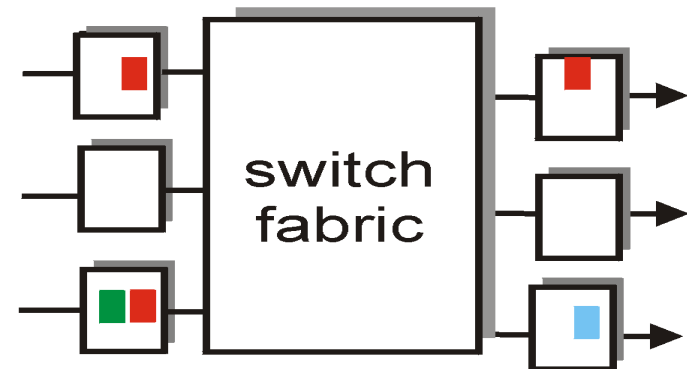
green packet
experiences HOL blocking

Network Layer

o 4.1 Introduction

o 4.2 What's inside a router

o 4.3 IP: Internet Protocol

   o Datagram format

   o IPv4 addressing

   o ICMP

   o IPv6

The Internet Network layer

Host, router network layer functions:

Network layer

Transport layer: TCP, UDP

Routing protocols
•path selection
•RIP, OSPF, BGP

IP protocol
•addressing conventions
•datagram format
•packet handling conventions

forwarding table

ICMP protocol
•error reporting
•router "signaling"

Link layer

physical layer

Network Layer

o 4.1 Introduction

o 4.2 What's inside a router

o 4.3 IP: Internet Protocol

  o Datagram format

  o IPv4 addressing

  o ICMP

  o IPv6

IP datagram format

IP protocol version number

header length (bytes)

"type" of data

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to

total datagram length (bytes)

for fragmentation/ reassembly

E.g. timestamp, record route taken, specify list of routers to visit.

32 bits

ver	head. len	type of service	length
16-bit identifier		flgs	fragment offset
time to live	upper layer	header checksum
32 bit source IP address
32 bit destination IP address
Options (if any)
data (variable length, typically a TCP or UDP segment)

how much overhead with TCP?

○ 20 bytes of TCP

○ 20 bytes of IP

○ = 40 bytes + app layer overhead

computer
N E T
W O R K S

IP Fragmentation & Reassembly

o network links have MTU (max.transfer size) - largest possible link-level frame.
   o different link types, different MTUs

o large IP datagram divided ("fragmented") within net
   o one datagram becomes several datagrams
   o "reassembled" only at final destination
   o IP header bits used to identify, order related fragments

fragmentation:
in: one large datagram
out: 3 smaller datagrams
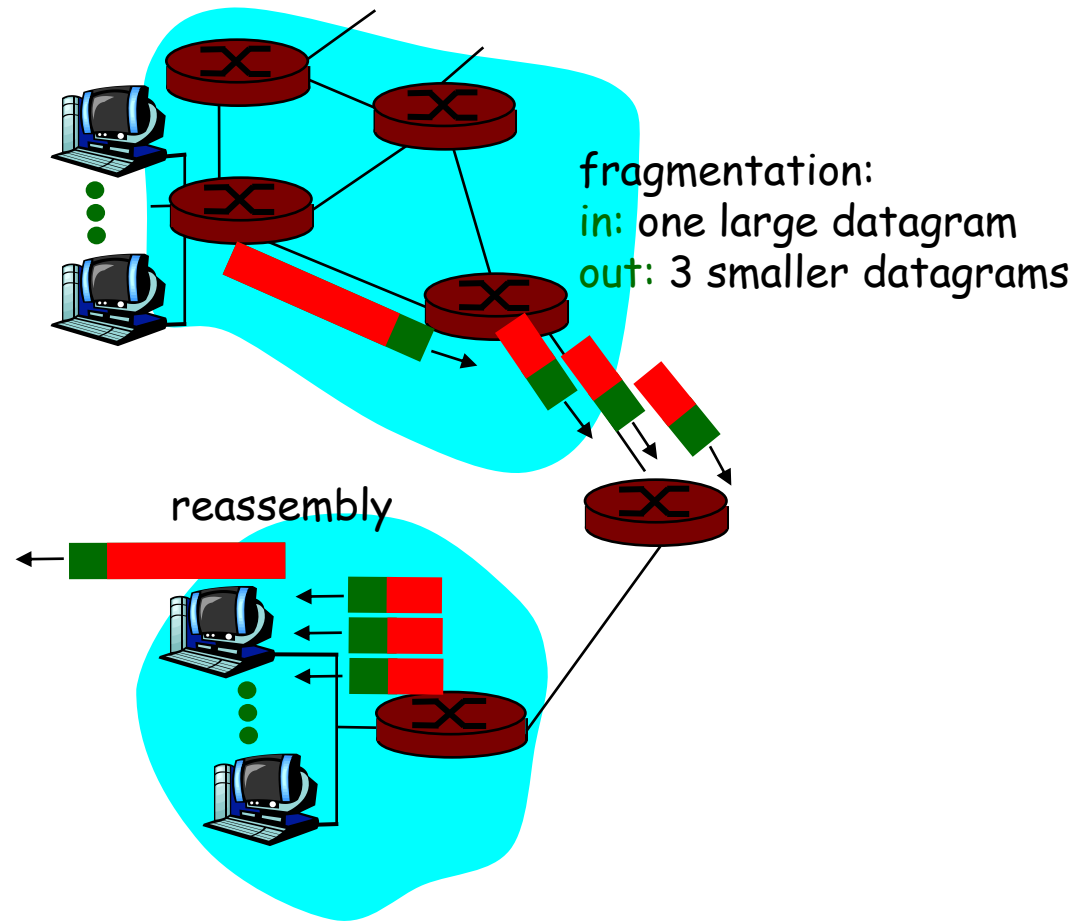
reassembly

IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes

	length =4000	ID =x	fragflag =0	offset =0		

One large datagram becomes several smaller datagrams

1480 bytes in data field

	length =1500	ID =x	fragflag =1	offset =0		

offset = 1480/8

	length =1500	ID =x	fragflag =1	offset =185		

	length =1040	ID =x	fragflag =0	offset =370		

Network Layer

o 4.1 Introduction

o 4.2 What's inside a router

o 4.3 IP: Internet Protocol

  o Datagram format

  o IPv4 addressing

  o ICMP

  o IPv6

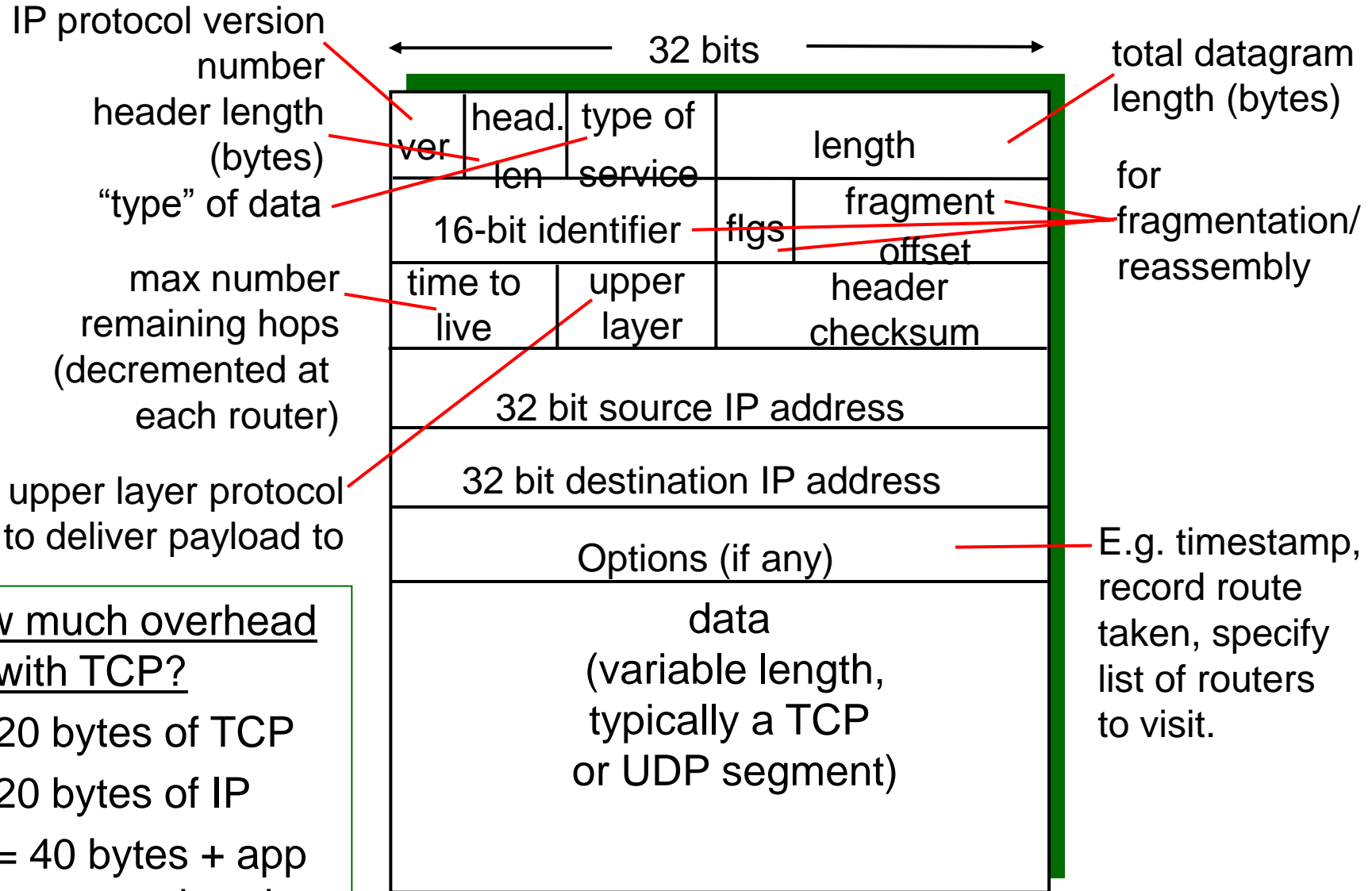# IP Addressing: introduction

o **IP address:** 32-bit identifier for host, router *interface*

o *interface:* connection between host/router and physical link

- o router's typically have multiple interfaces
- o host typically has one interface
- o IP addresses associated with each interface

223.1.1.1

223.1.2.1

223.1.1.2

223.1.1.4   223.1.2.9

223.1.1.3   223.1.3.27   223.1.2.2

223.1.3.1   223.1.3.2

223.1.1.1 = 11011111 00000001 00000001 00000001

       223   1   1   1
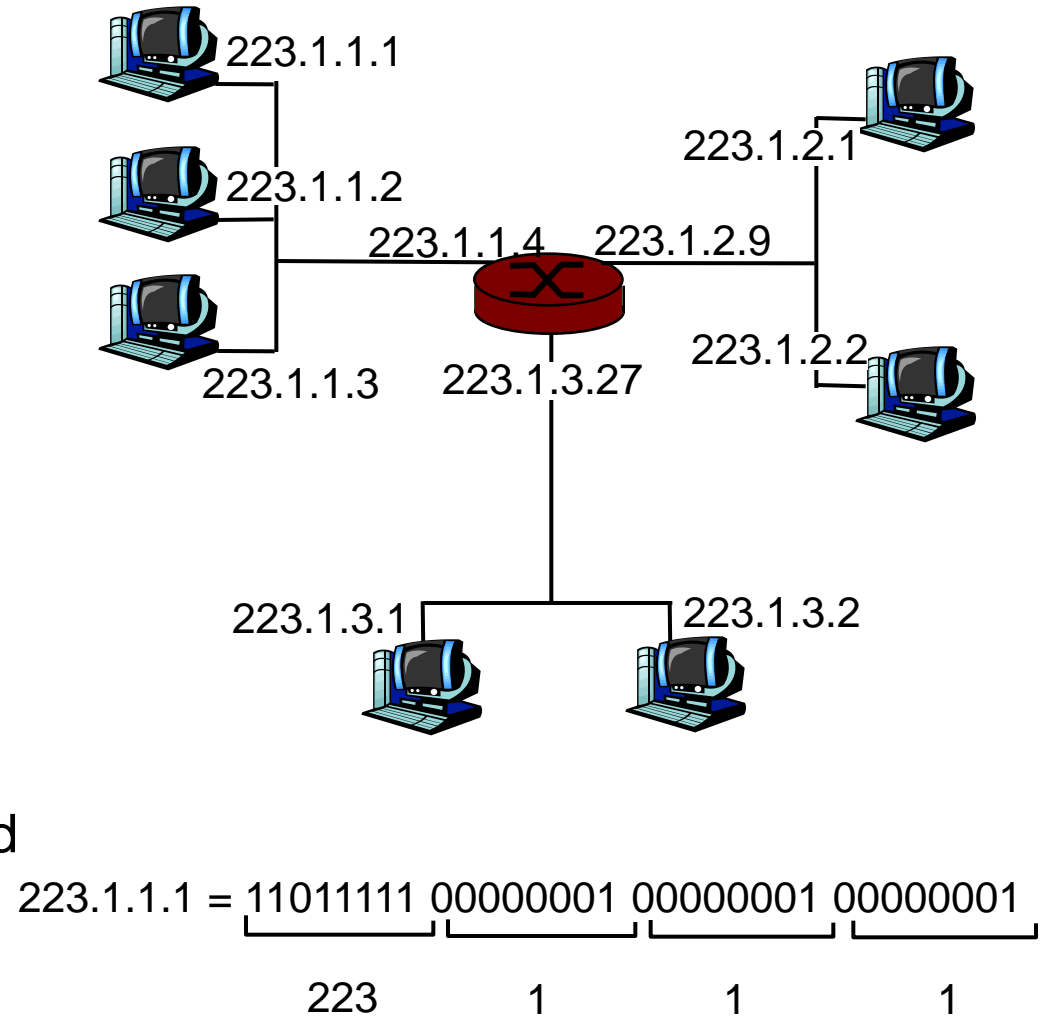
Subnets

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

223.1.1.1

223.1.1.2

223.1.1.4

223.1.1.3

223.1.2.1

223.1.2.9

223.1.2.2

223.1.3.27

subnet

223.1.3.1

223.1.3.2
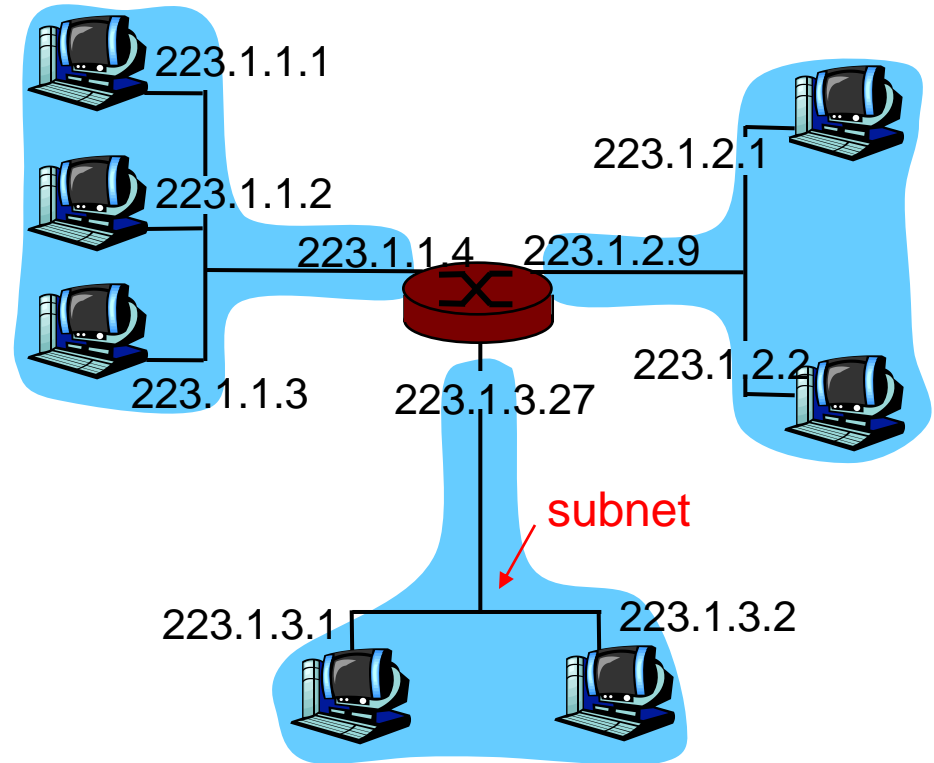
network consisting of 3 subnets

Subnets

Recipe

o To determine the subnets, detach each interface from its host or router, creating islands of isolated networks. Each isolated network is called a subnet.

223.1.1.0/24

223.1.2.0/24

223.1.3.0/24
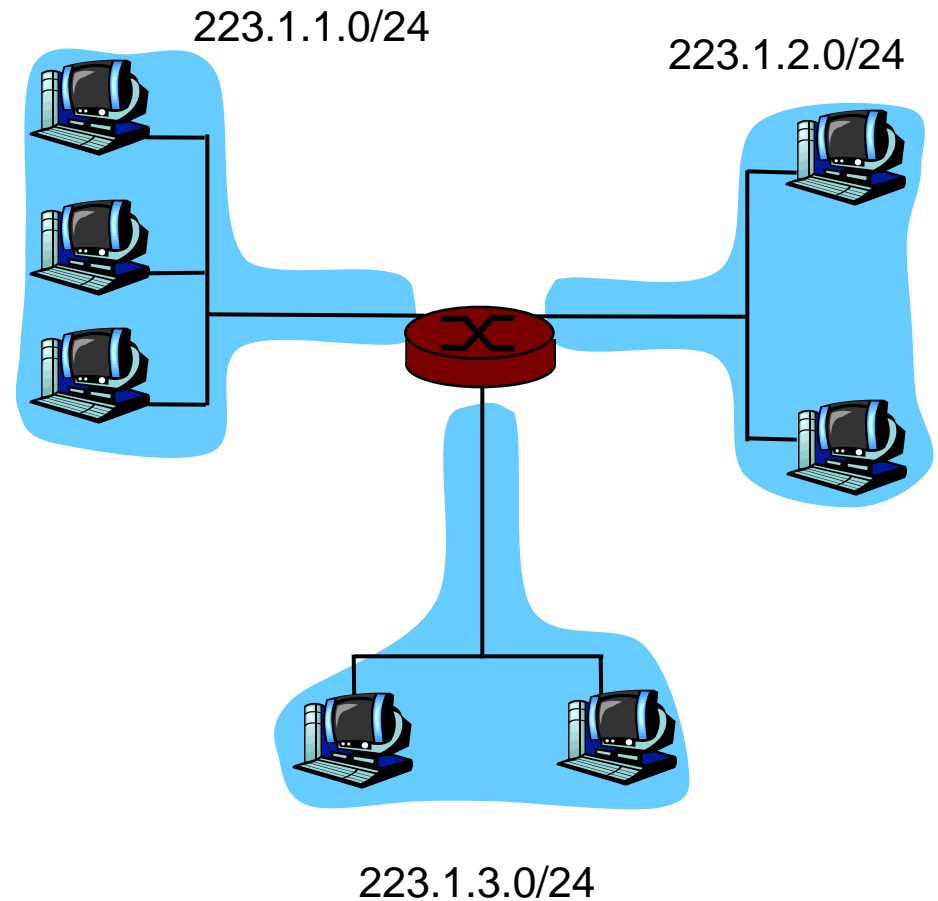
Subnet mask: /24

Subnets

How many?

223.1.1.2

223.1.1.1   223.1.1.4

223.1.1.3

223.1.9.2   223.1.7.0

223.1.9.1   223.1.7.1

223.1.8.1   223.1.8.0

223.1.2.6   223.1.3.27

223.1.2.1   223.1.2.2   223.1.3.1   223.1.3.2
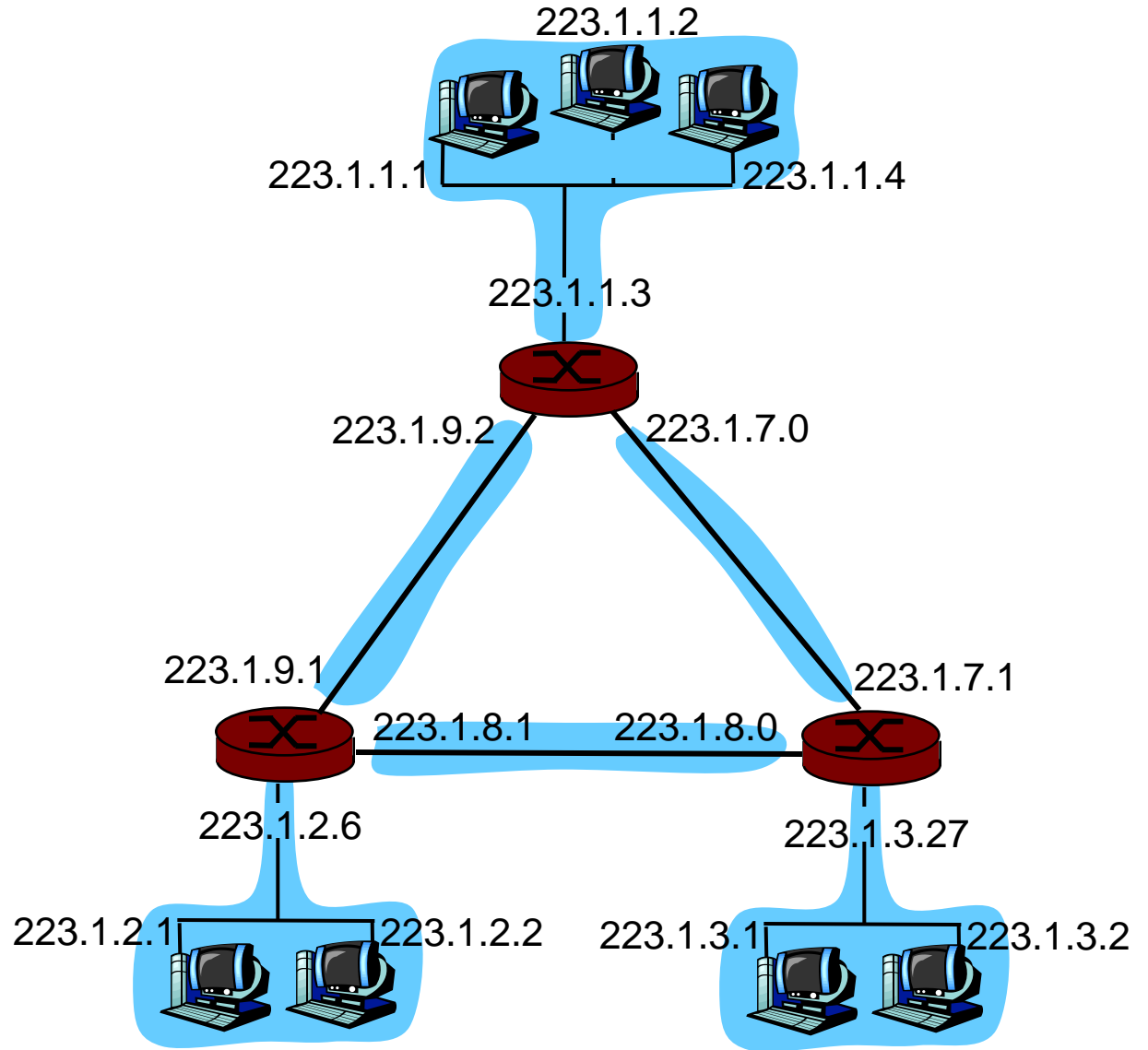
IP addressing: Classful Network

Class	Lead. Bits	Netw. Addr. Bits	No. Of Networks	No. Of Hosts	Addresses
A	0	8	128 $(= 2^7)$	16,777,216 $(= 2^{24})$	0.0.0.0 to 127.255.255.255
B	10	16	16,384 $(= 2^{14})$	65,536 $(= 2^{16})$	128.0.0.0 to 191.255.255.255
C	110	24	2,097,152 $(= 2^{21})$	256 $(= 2^8)$	192.0.0.0 to 223.255.255.255
D	1110	n/d	n/d	n/d	224.0.0.0 to 239.255.255.255
E	1111	n/d	n/d	n/d	240.0.0.0 to 255.255.255.254

IP addressing: CIDR

CIDR: **C**lassless **I**nter**D**omain **R**outing

◦ subnet portion of address of arbitrary length

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

```
   <------------ subnet ------------><-- host -->
           part                         part
11001000  00010111  00010000  00000000
```
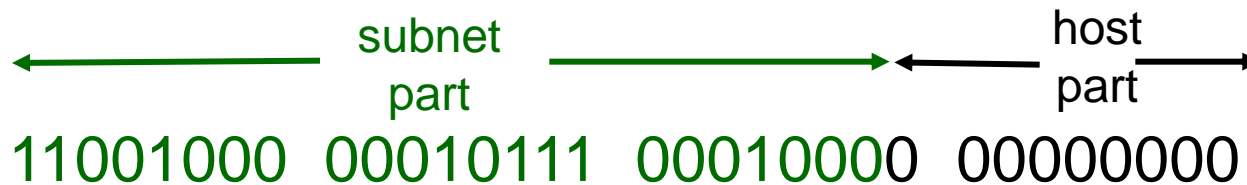
200.23.16.0/23

IP addresses: how to get one?

o Q: How does a host get IP address?

o A1: hard-coded by system admin in a file

o A2: DHCP: Dynamic Host Configuration Protocol

  o dynamically get address from as server

  o "plug-and-play"

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:

o host broadcasts "DHCP discover" msg

o DHCP server responds with "DHCP offer" msg

o host requests IP address: "DHCP request" msg

o DHCP server sends address: "DHCP ack" msg

IP addresses: how to get one?

Q: How does *network* get subnet part of IP addr?

A: gets allocated portion of its provider ISP's address space

ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	…..	….	….
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:

Organization 0

200.23.16.0/23

Organization 1

200.23.18.0/23

Organization 2

200.23.20.0/23

Organization 7

200.23.30.0/23

Fly-By-Night-ISP

"Send me anything with addresses beginning 200.23.16.0/20"

ISPs-R-Us

"Send me anything with addresses beginning 199.31.0.0/16"
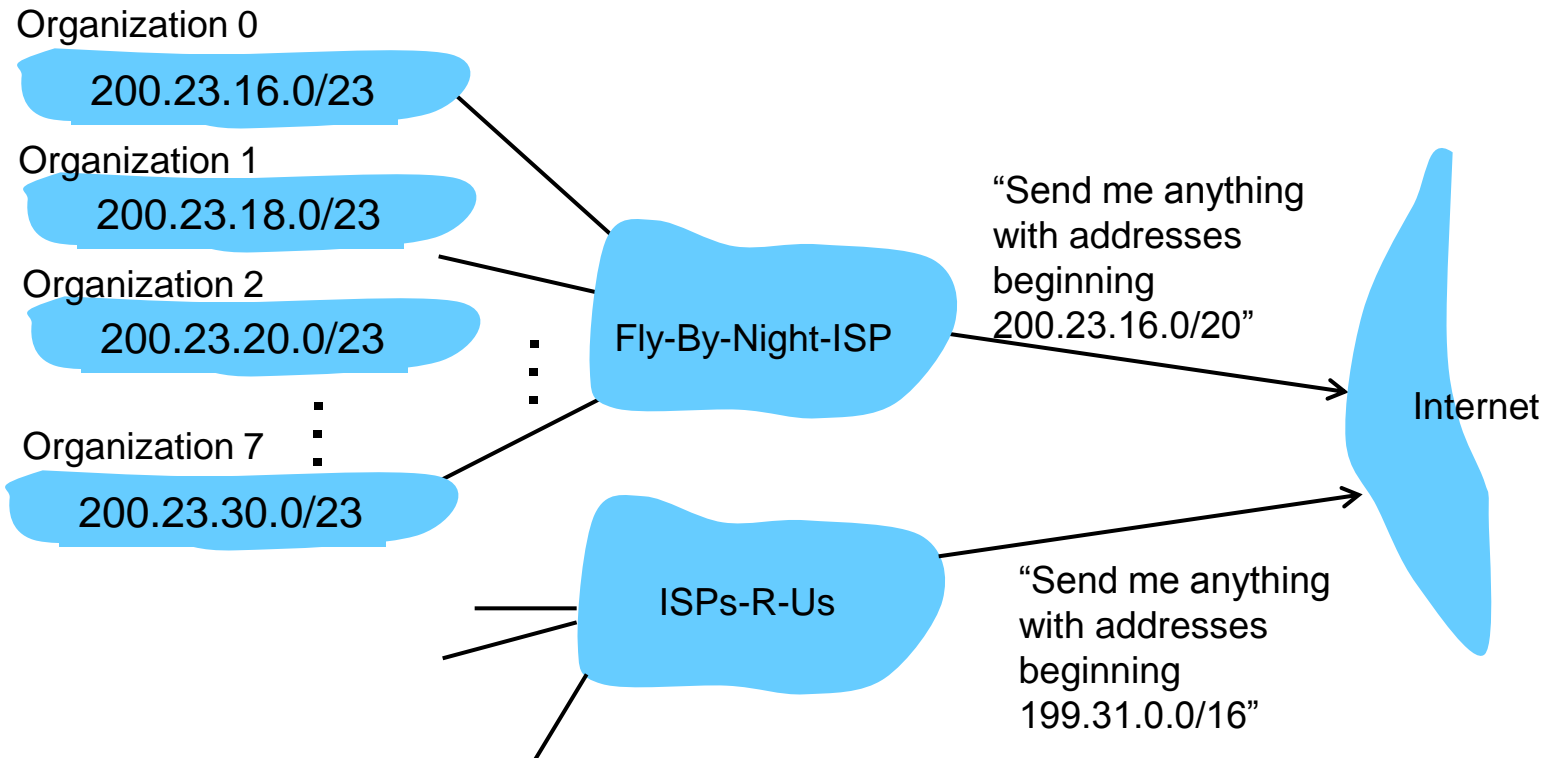
Internet

Hierarchical addressing: more specific routes

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

Organization 0
200.23.16.0/23

Organization 2
200.23.20.0/23

Organization 7
200.23.30.0/23

Fly-By-Night-ISP

"Send me anything with addresses beginning 200.23.16.0/20"

Internet

ISPs-R-Us

Organization 1
200.23.18.0/23

"Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23"
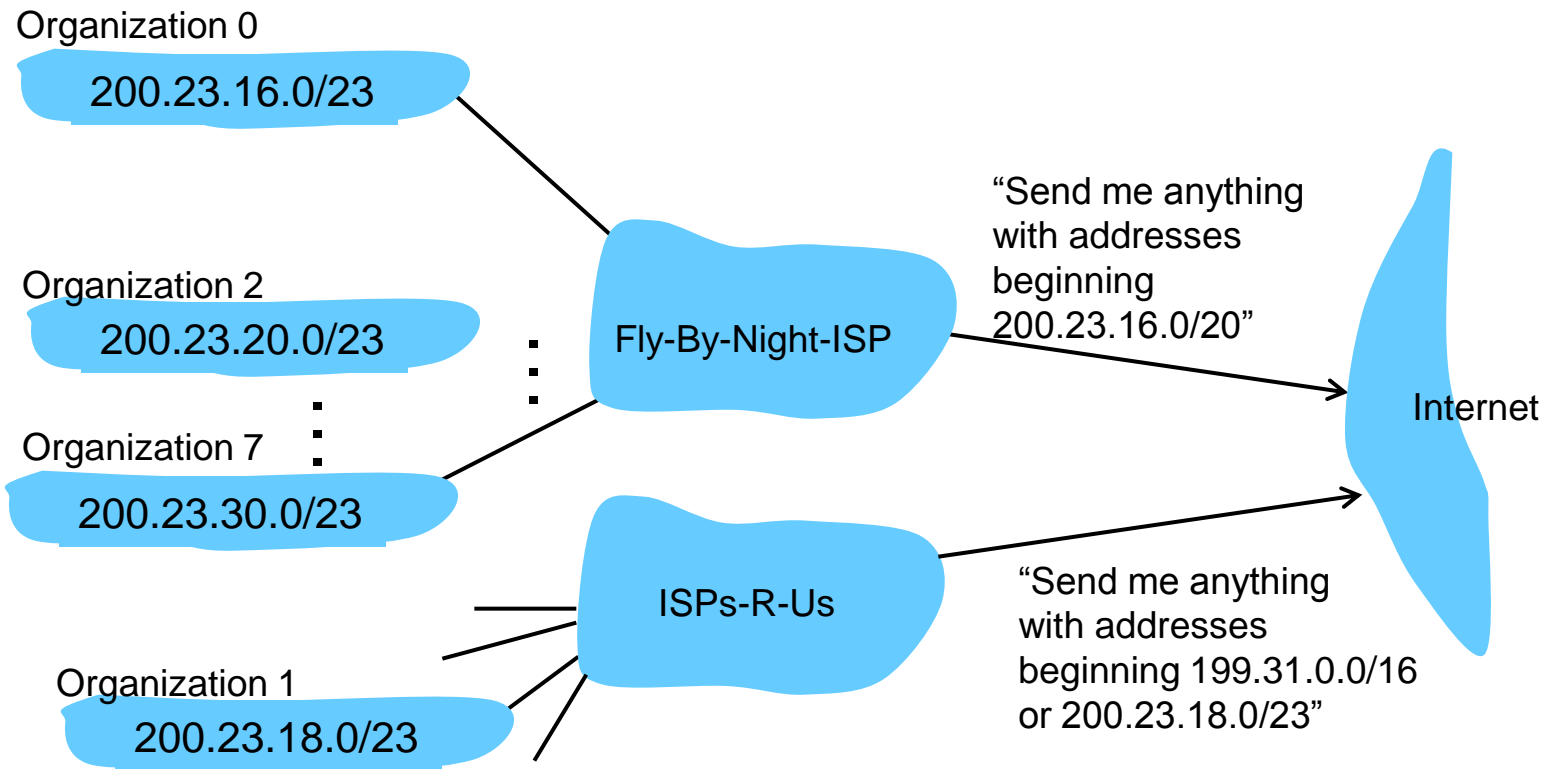
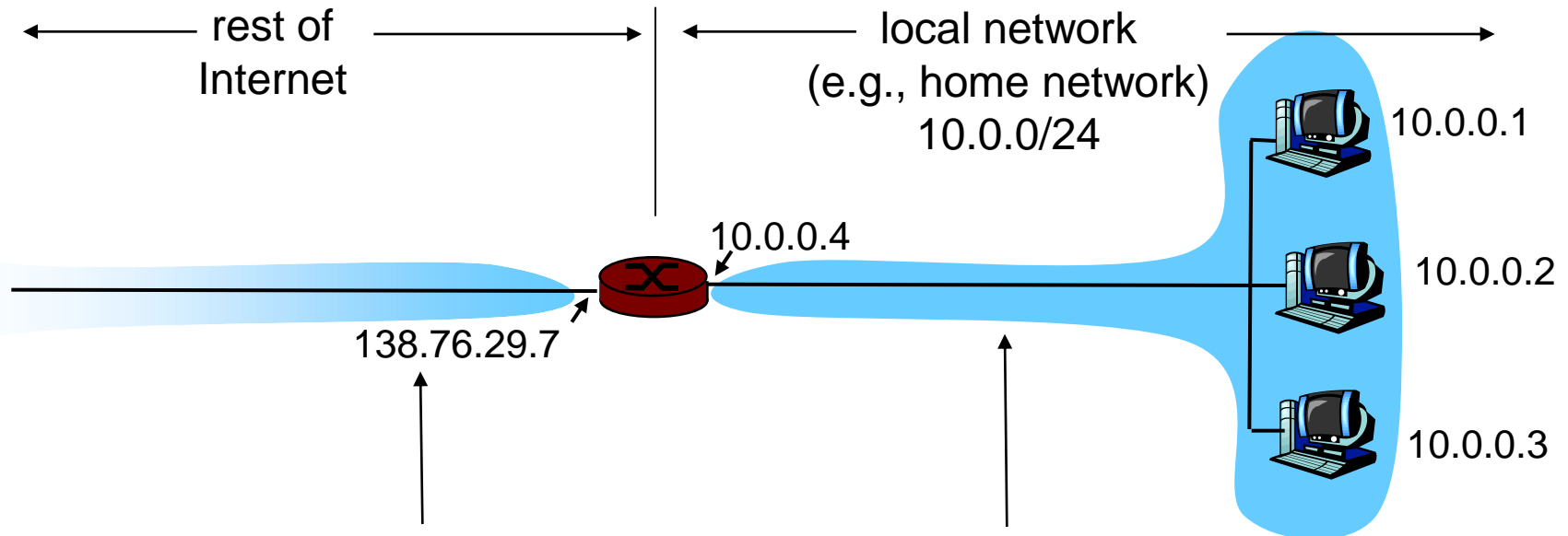# IP addressing: the last word…

o Q: How does an ISP get block of addresses?

o A: ICANN: Internet Corporation for Assigned Names and Numbers

- o allocates addresses
- o manages DNS
- o assigns domain names, resolves disputes

NAT: Network Address Translation

o Motivation: local network uses just one IP address as far as outside world is concerned:

  o range of addresses not needed from ISP: just one IP address for all devices

  o can change addresses of devices in local network without notifying outside world

  o can change ISP without changing addresses of devices in local network

  o devices inside local net not explicitly addressable, visible by outside world (a security plus).

NAT: Network Address Translation

rest of Internet

local network (e.g., home network) 10.0.0/24

10.0.0.4

138.76.29.7

10.0.0.1

10.0.0.2

10.0.0.3

All datagrams *leaving* local network have same single source NAT IP address: 138.76.29.7, different source port numbers
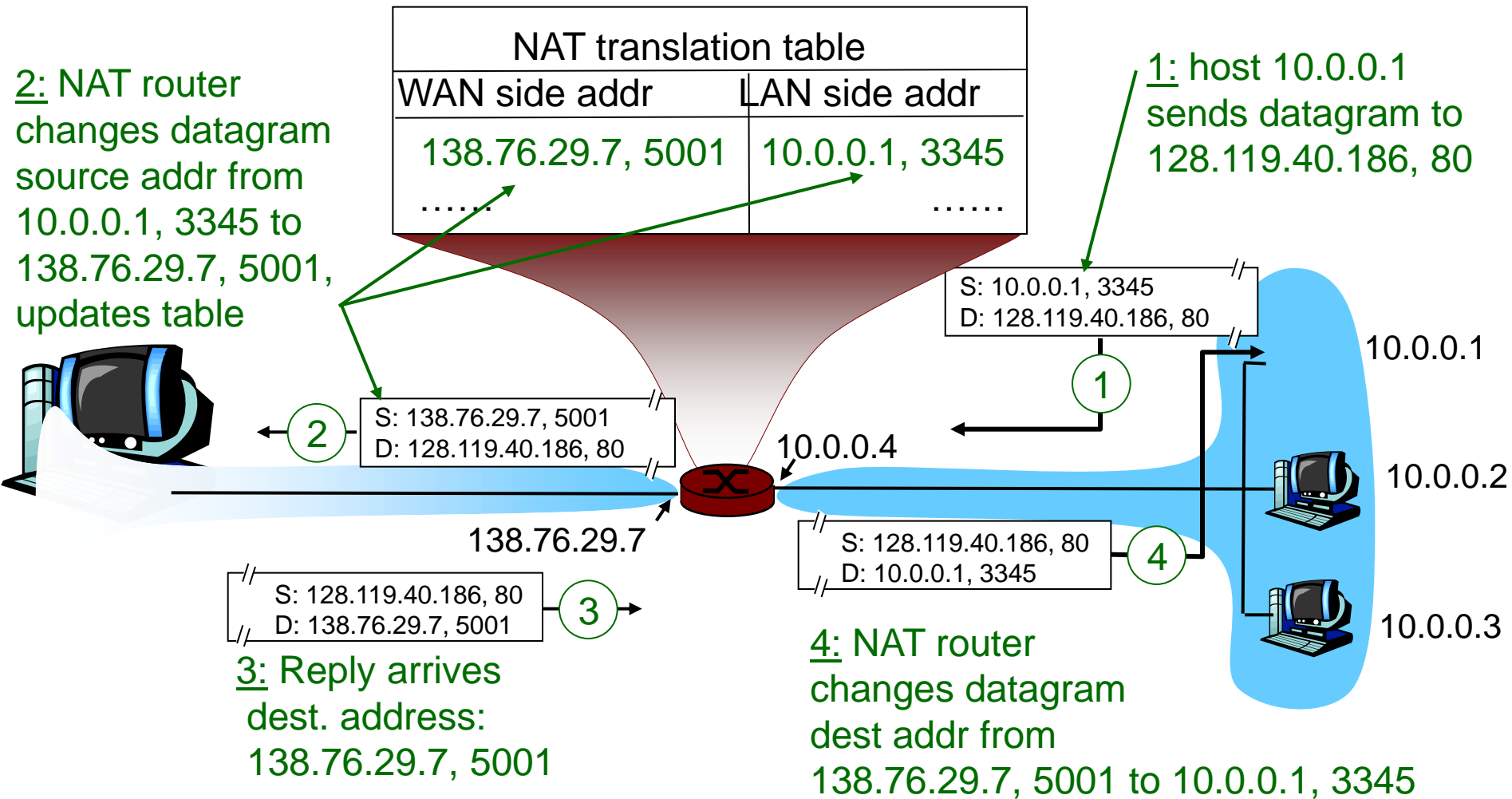
Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

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 translation table

WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

S: 10.0.0.1, 3345
D: 128.119.40.186, 80

1

10.0.0.1

S: 138.76.29.7, 5001
D: 128.119.40.186, 80

2

10.0.0.4

138.76.29.7

S: 128.119.40.186, 80
D: 10.0.0.1, 3345

4

10.0.0.2

S: 128.119.40.186, 80
D: 138.76.29.7, 5001

3

3: Reply arrives dest. address: 138.76.29.7, 5001

10.0.0.3

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

NAT: Network Address Translation

o 16-bit port-number field:

    o 60,000 simultaneous connections with a single LAN-side address!

o NAT is controversial:

    o routers should only process up to layer 3

    o violates end-to-end argument

      • NAT possibility must be taken into account by app designers, eg, P2P applications

    o address shortage should instead be solved by IPv6
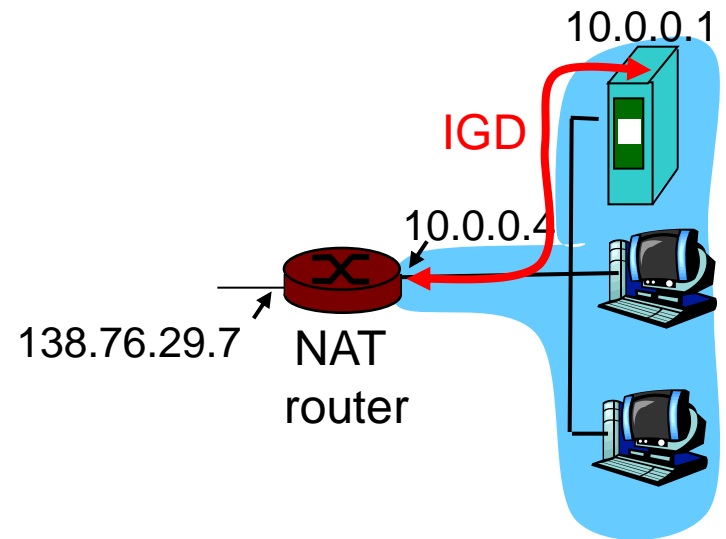
NAT traversal problem

o client wants to connect to server with address 10.0.0.1

    o server address 10.0.0.1 local to LAN (client can't use it as destination addr)

    o only one externally visible NATted address: 138.76.29.7

o solution 1: statically configure NAT to forward incoming connection requests at given port to server

    o e.g., (123.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000

Client

?

10.0.0.1

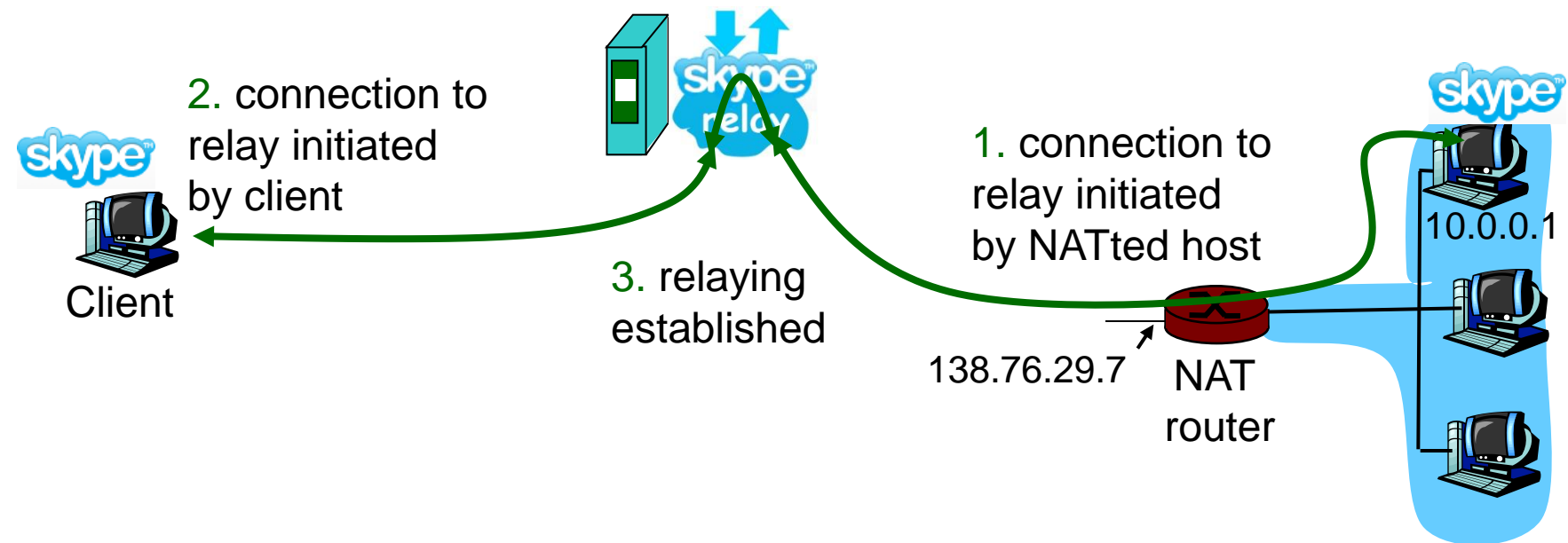10.0.0.4

138.76.29.7

NAT router
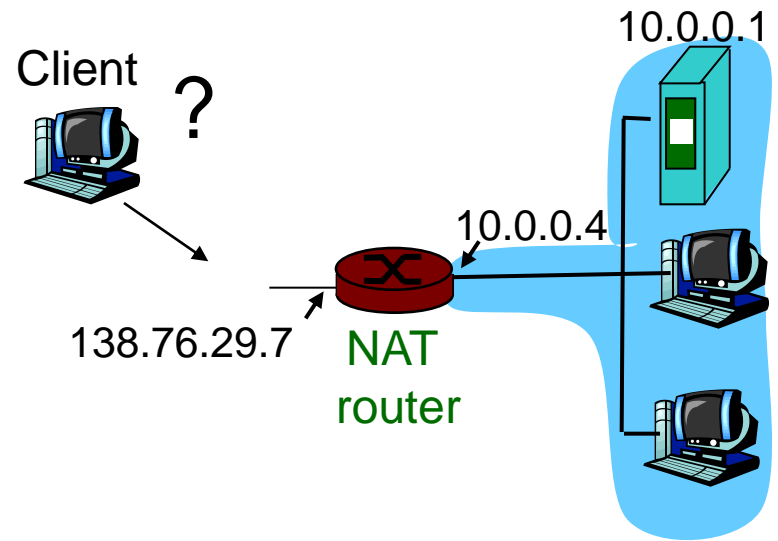
NAT traversal problem

o solution 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATted host to:

  ❖ learn public IP address (138.76.29.7)

  ❖ add/remove port mappings (with lease times)

  i.e., automate static NAT port map configuration

10.0.0.1

IGD

10.0.0.4

138.76.29.7   NAT router

NAT traversal problem

o solution 3: relaying (used in Skype)

   o NATed client establishes connection to relay

   o External client connects to relay

   o relay bridges packets between to connections

2. connection to relay initiated by client

1. connection to relay initiated by NATted host

3. relaying established

Client

138.76.29.7

NAT router

10.0.0.1

Network Layer

o 4.1 Introduction

o 4.2 What's inside a router

o 4.3 IP: Internet Protocol

- o Datagram format
- o IPv4 addressing
- o ICMP
- o IPv6

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 msgs carried in IP datagrams

○ 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

Traceroute and ICMP

o Source sends series of UDP segments to dest
  o First has TTL =1
  o Second has TTL=2, etc.
  o Unlikely port number
o When nth datagram arrives to nth router:
  o Router discards datagram
  o And sends to source an ICMP message (type 11, code 0)
  o Message includes name of router& IP address

o When ICMP message arrives, source calculates RTT
o Traceroute does this 3 times

Stopping criterion

o UDP segment eventually arrives at destination host
o Destination returns ICMP "host unreachable" packet (type 3, code 3)
o When source gets this ICMP, stops.

Network Layer

o 4.1 Introduction

o 4.2 What's inside a router

o 4.3 IP: Internet Protocol

    o Datagram format
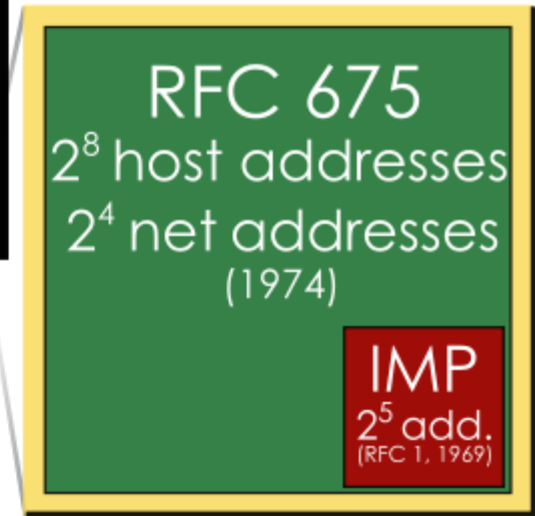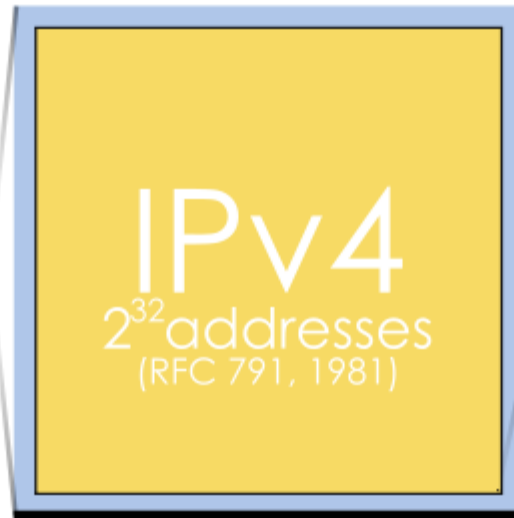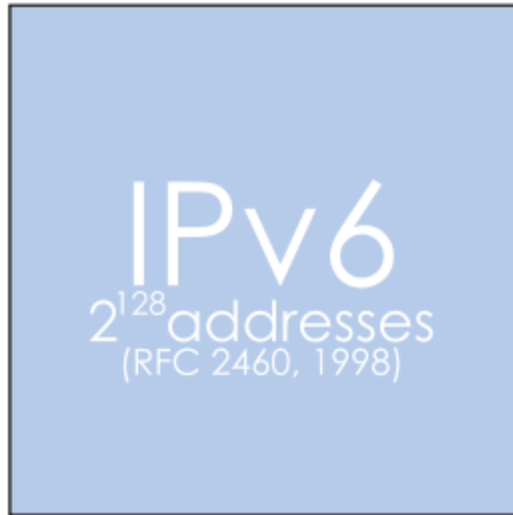
    o IPv4 addressing

    o ICMP

    o IPv6

IPv6

o Initial motivation: 32-bit address space soon to be completely allocated.

o Additional motivation:

  o header format helps speed processing/forwarding

  o header changes to facilitate QoS

  IPv6 datagram format:

  o fixed-length 40 byte header

  o no fragmentation allowed

IPv4 vs IPv6

IPv6
2^{128} addresses
(RFC 2460, 1998)

IPv4
2^{32} addresses
(RFC 791, 1981)

RFC 675
2^8 host addresses
2^4 net addresses
(1974)

IMP
2^5 add.
(RFC 1, 1969)

A diagram demonstrating the massive growth in address space under each protocol.

Each cascading block is a magnification of a tiny area in the preceding block, represented by a black square.

Image is to scale, except the black area is enlarged for ease of viewing

computer
N E T
W O R K S

IPv6 Header (Cont)

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
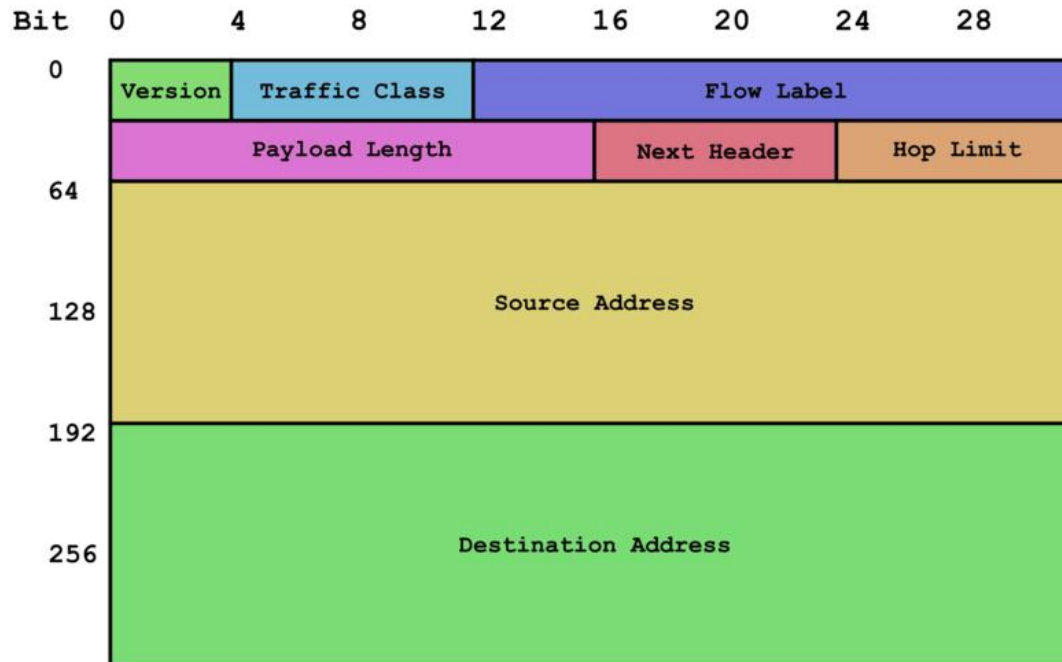
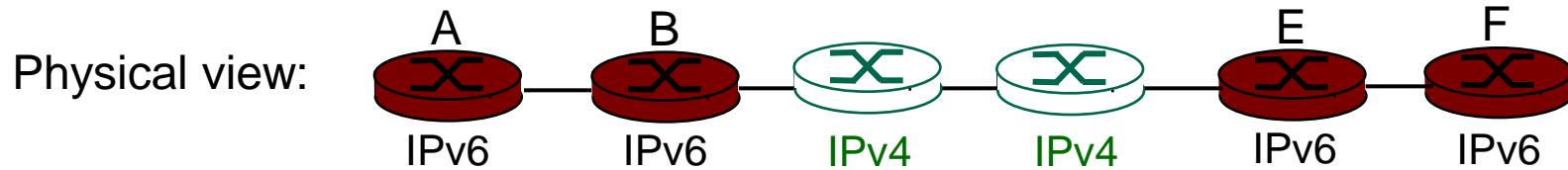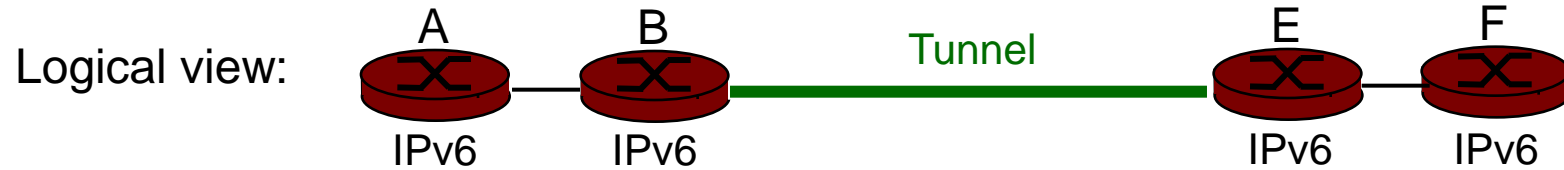Source: http://commons.wikimedia.org/wiki/File:IPv6_header_rv1.png

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"
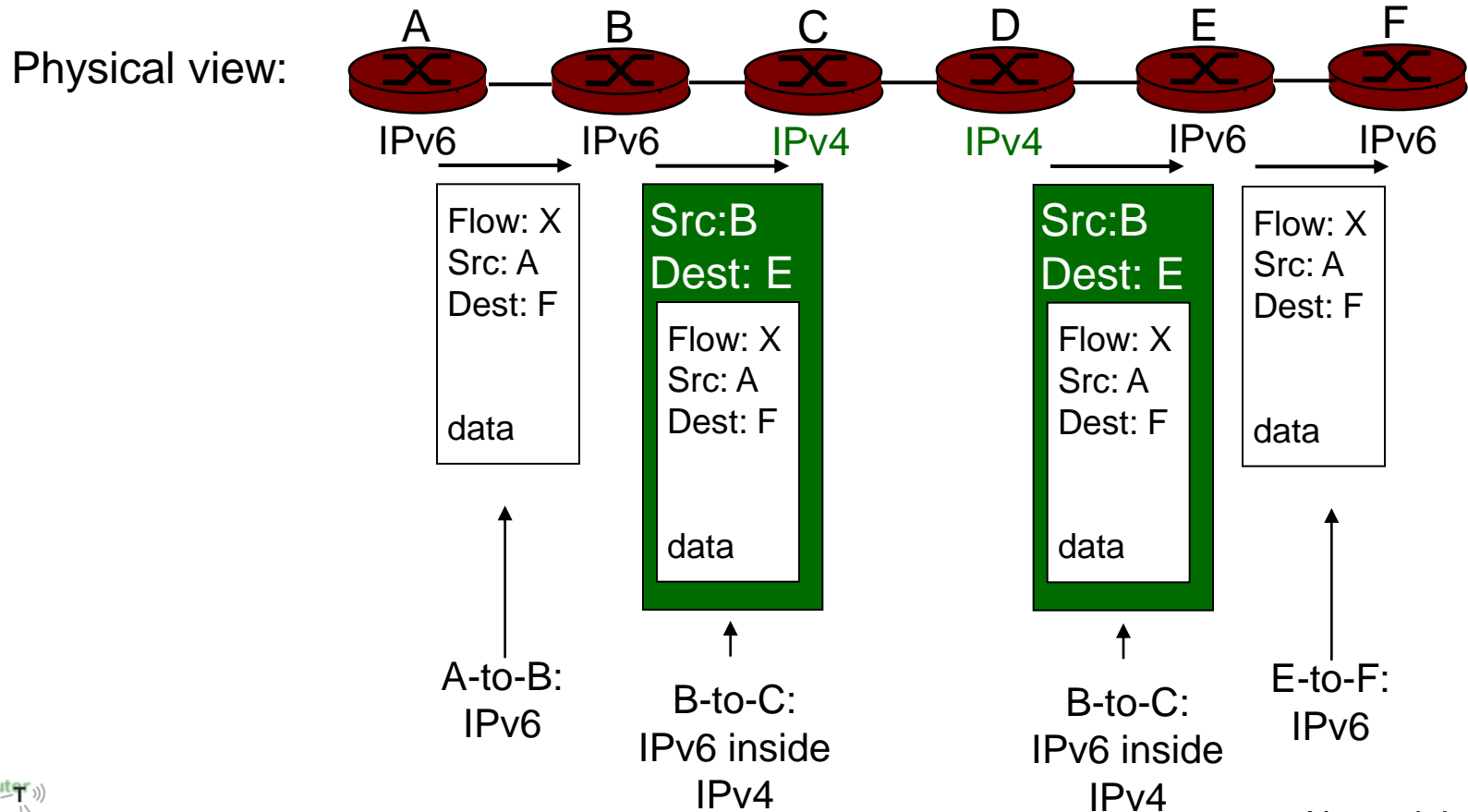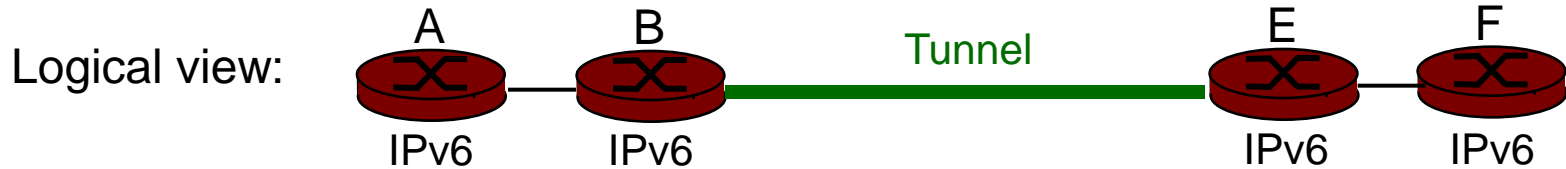
　　○ multicast group management functions

Transition From IPv4 To IPv6

o Not all routers can be upgraded simultaneously
  o no "flag days"
  o How will the network operate with mixed IPv4 and IPv6 routers?

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

o Dual stack: Network stack supports IPv4 and IPv6

Tunneling

Logical view:

A — B ——— Tunnel ——— E — F

IPv6   IPv6   IPv6   IPv6

Physical view:

A — B — IPv4 — IPv4 — E — F

IPv6   IPv6   IPv4   IPv4   IPv6   IPv6

Tunneling

Logical view:

A   B   Tunnel   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

Dual stack

o Nodes have the ability to send and receive both IPv4 and IPv6 packets

o Direct connection to IPv4 nodes using IPv4 packets

o Direct connection to IPv6 nodes using IPv6 packets

o Can be used together with tunneling

Done for Today

○ Thank you for your patience

○ Any Questions?