Network: Data Forwarding

Reading: KR 4.4, 4.6, 4.7

IP Addressing Scheme

- We need an address to uniquely identify each destination

- Routing scalability needs flexibility in aggregation of destination addresses
  - We should be able to aggregate a set of destinations as a single routing unit

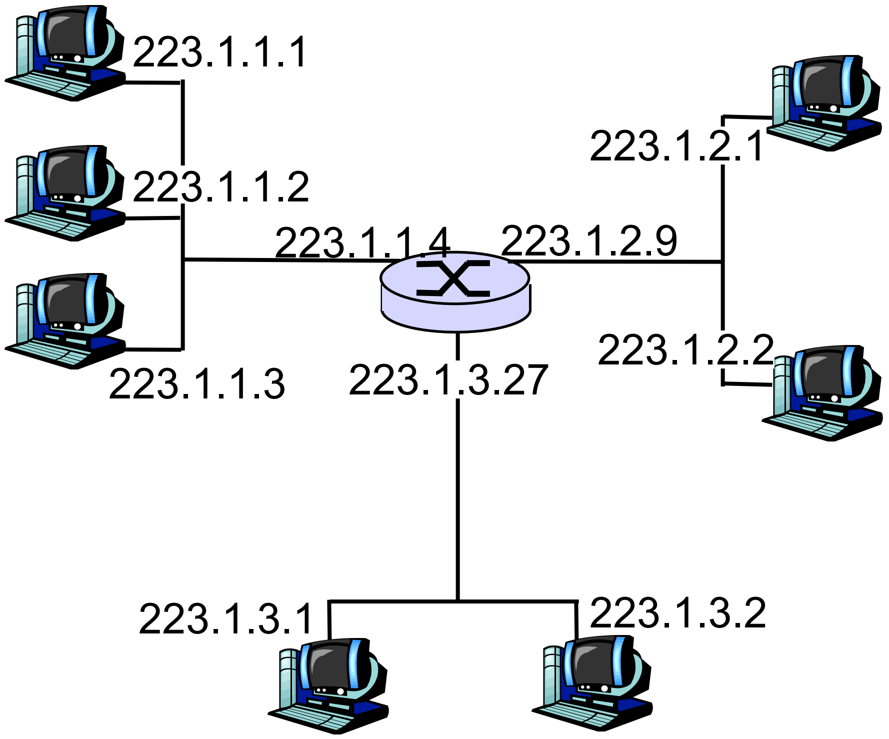
- Preview: the unit of routing in the Internet is a network---the destinations in the routing protocols are networks

IP Address: An IP Address Identifies an Interface

- IP address: 32-bit identifier for an *interface*
- *interface:*
  - Routers typically have multiple interfaces
  - Host may have multiple interfaces

```
%/sbin/ifconfig -a
```

```
C:\ipconfig
```

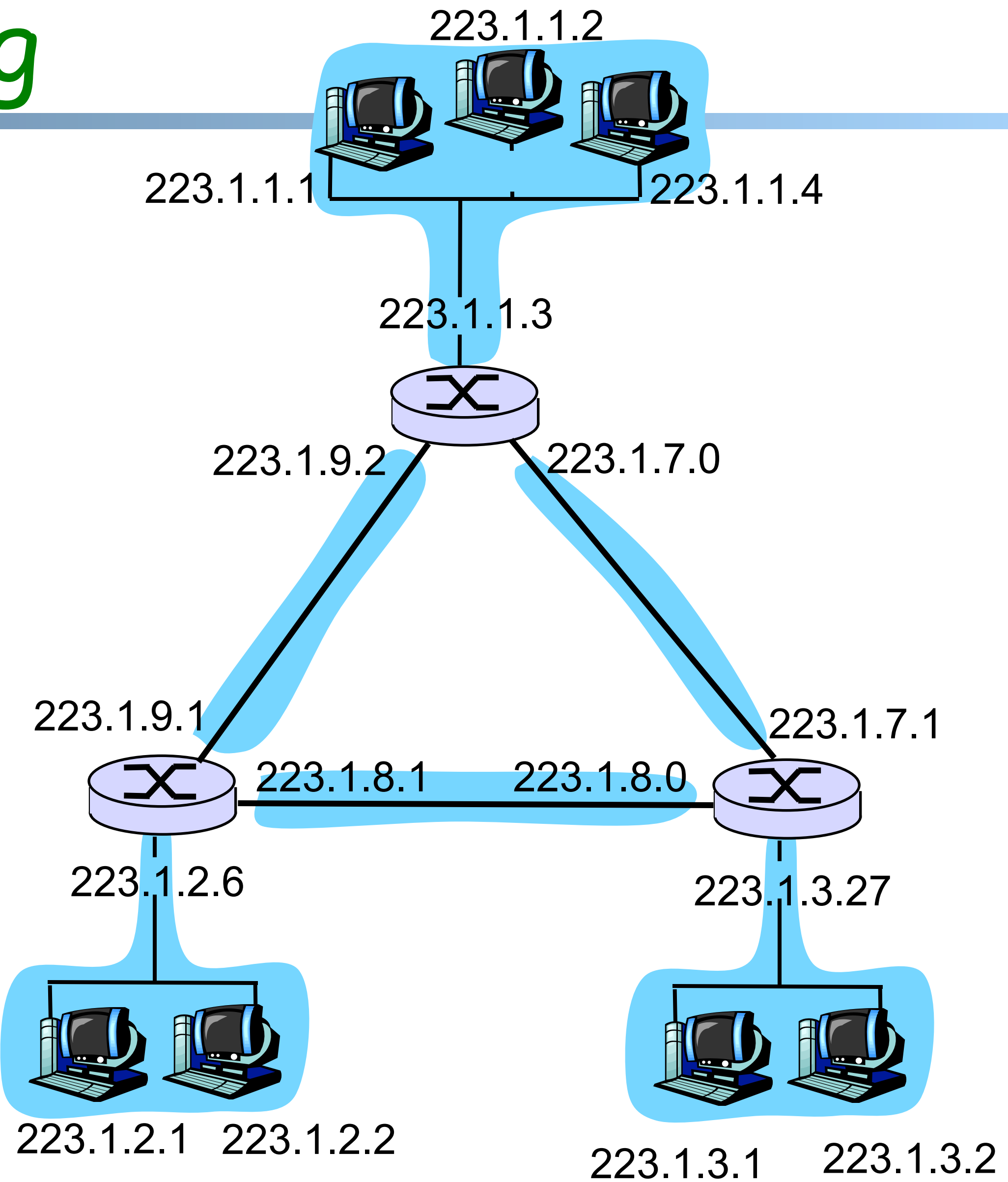


$223.1.3.2 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000011}_{3}\ \underbrace{00000010}_{2}$

IP Addressing

- IP address:
  - Network part
  - Host part

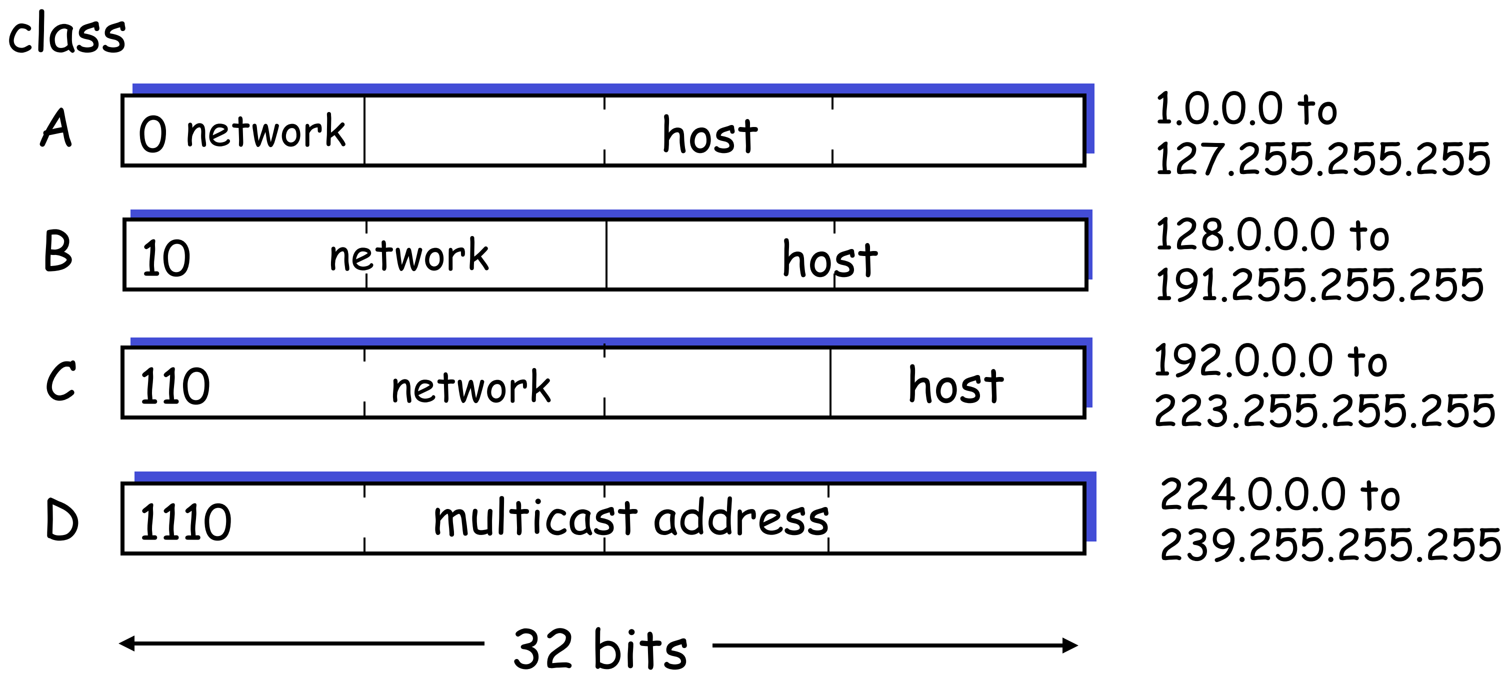
- *What's a network ?*
  (from IP address perspective)
  - is a unit of routing: can be routed together (depend on the routing protocol)



IP Addressing

given notion of “network”, let’s re-examine IP addresses:

“class-ful” addressing in the original IP design:



IP Addressing: CIDR

- (Static) classful addressing:
  - Inefficient use of address space, address space exhaustion
    - E.g., a class A net allocated enough addresses for 16 million hosts; a class B address may also be too big
  - Not flexible for aggregation

- CIDR: Classless InterDomain Routing
  - Network portion of address of arbitrary length
  - Address format: a.b.c.d/x, where x is # bits in network portion of address

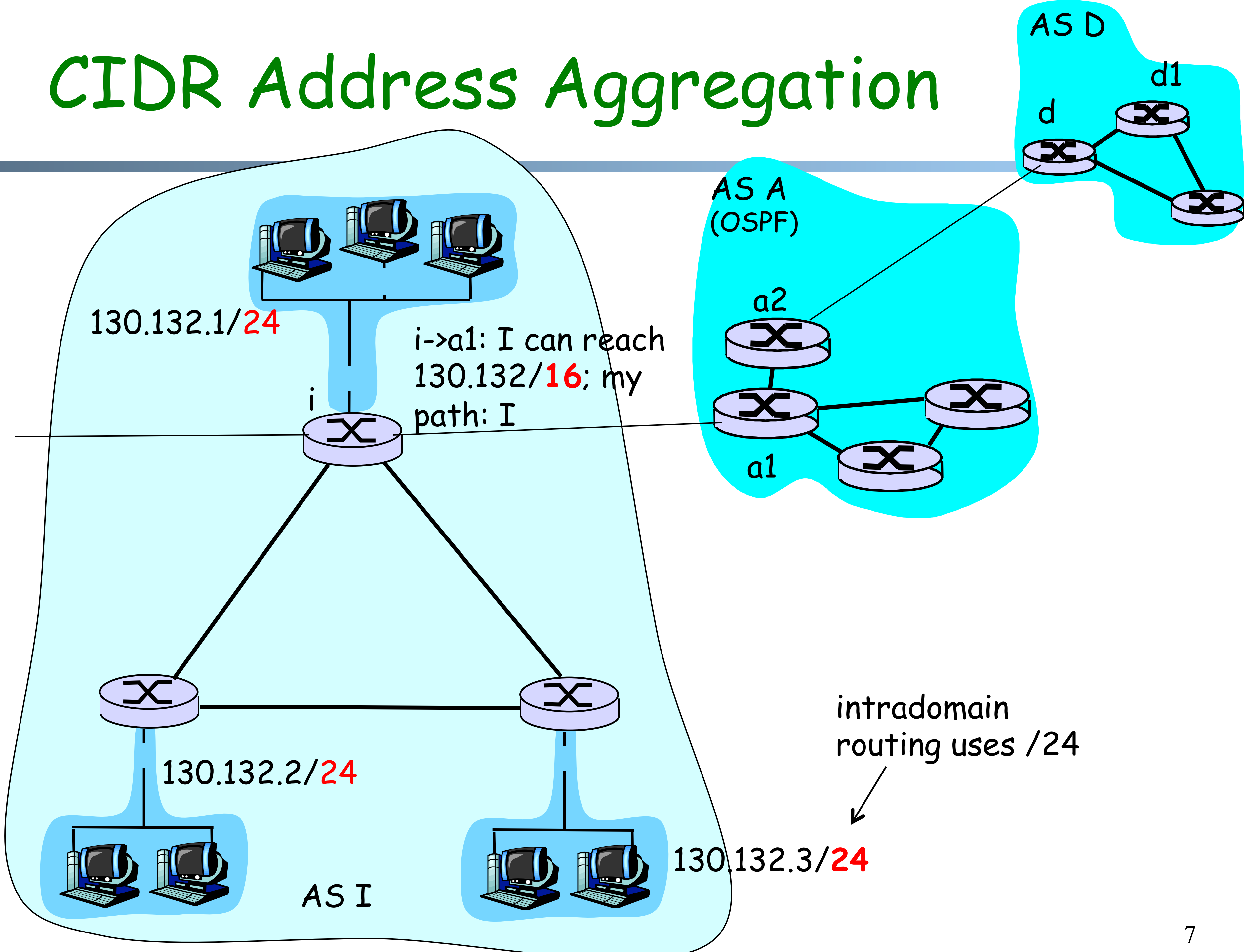
network part ←→ host part

11001000 00010111 00010000 00000000

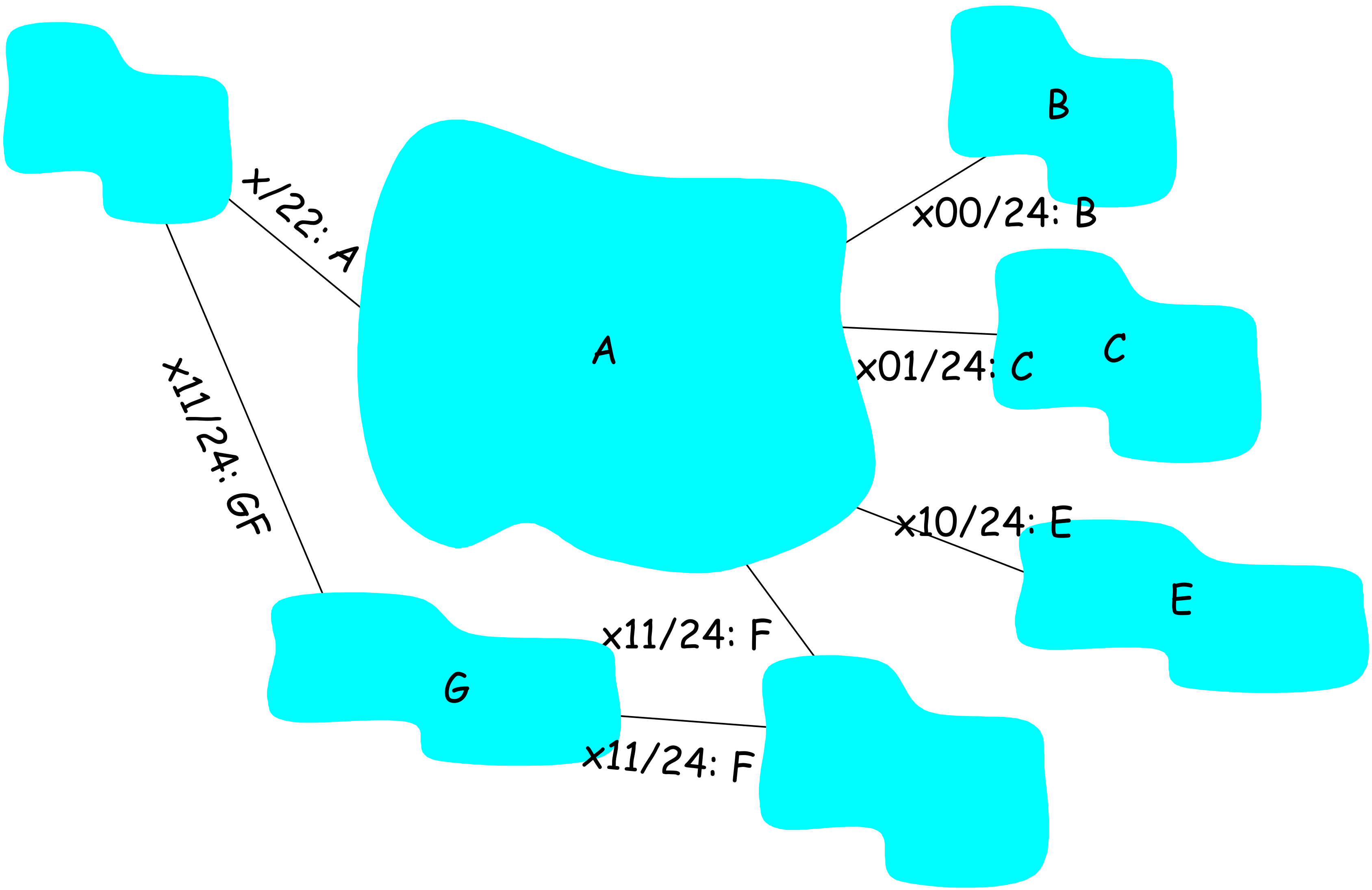
200.23.16.0/23

Some systems use mask (1's to indicate network bits), instead of the /x format

CIDR Address Aggregation



CIDR Address Aggregation



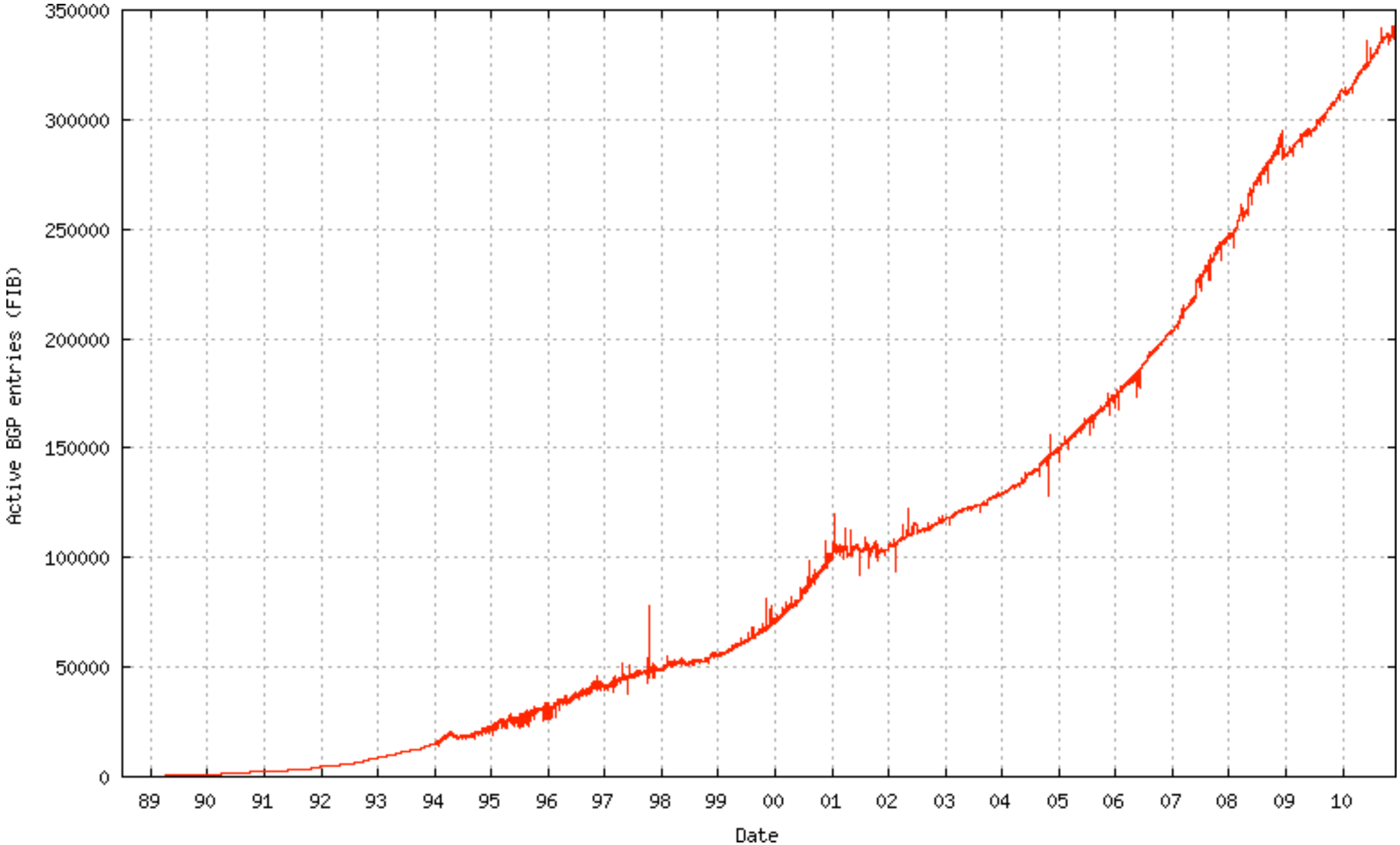
IP Addressing: How to Get One?

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

A: ICANN: Internet Corporation for Assigned Names and Numbers

- Allocates addresses
- Manages DNS
- Assigns domain names, resolves disputes

Routing Table Size of BGP



Active BGP Entries (http://bgp.potaroo.net/as1221/bgp-active.html)

IP addresses: How to Get One?

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

- Static configured
  - wintel: control-panel->network->configuration->tcp/ip->properties
  - unix: %/sbin/ifconfig eth0 inet 192.168.0.10 netmask 255.255.255.0
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - "plug-and-play"

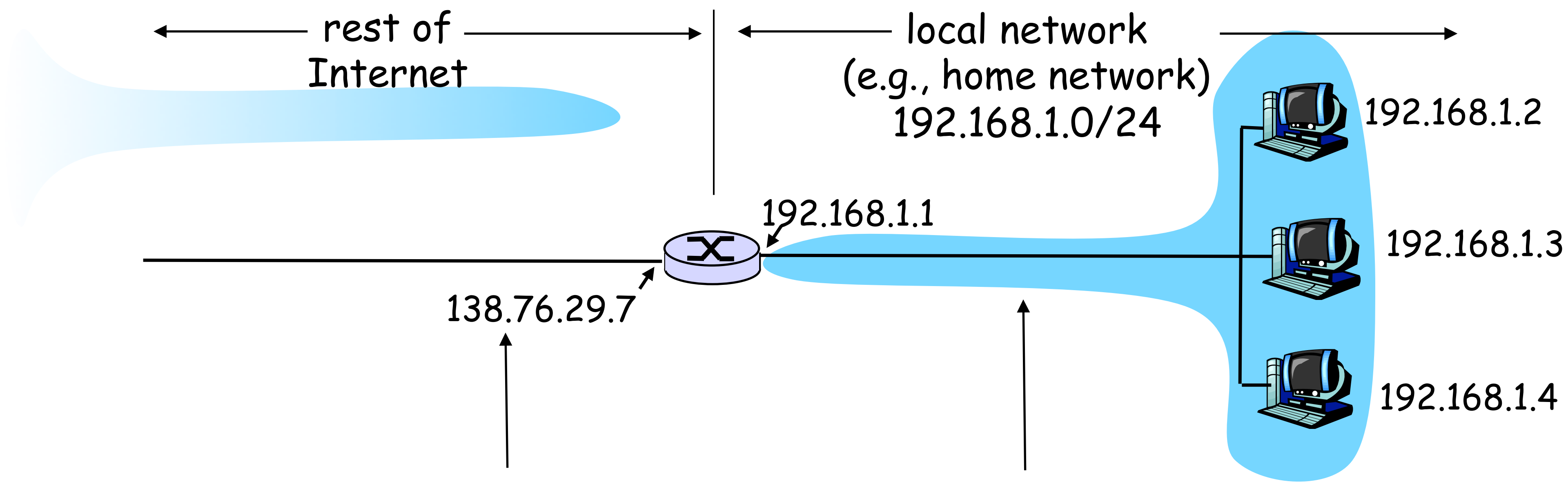
DHCP: Dynamic Host Configuration Protocol

- 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)
  - Support for mobile users who want to join network

- DHCP msgs:
  - Host broadcasts "DHCP discover" msg
  - DHCP server responds with "DHCP offer" msg
  - Host requests IP address: "DHCP request" msg
  - DHCP server sends address: "DHCP ack" msg

Network Address Translation: Motivation

- A local network uses just one public IP address as far as outside world is concerned
- Each device on the local network is assigned a private IP address



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 192.168.1/24 address for source, destination (as usual)

NAT: Network Address Translation

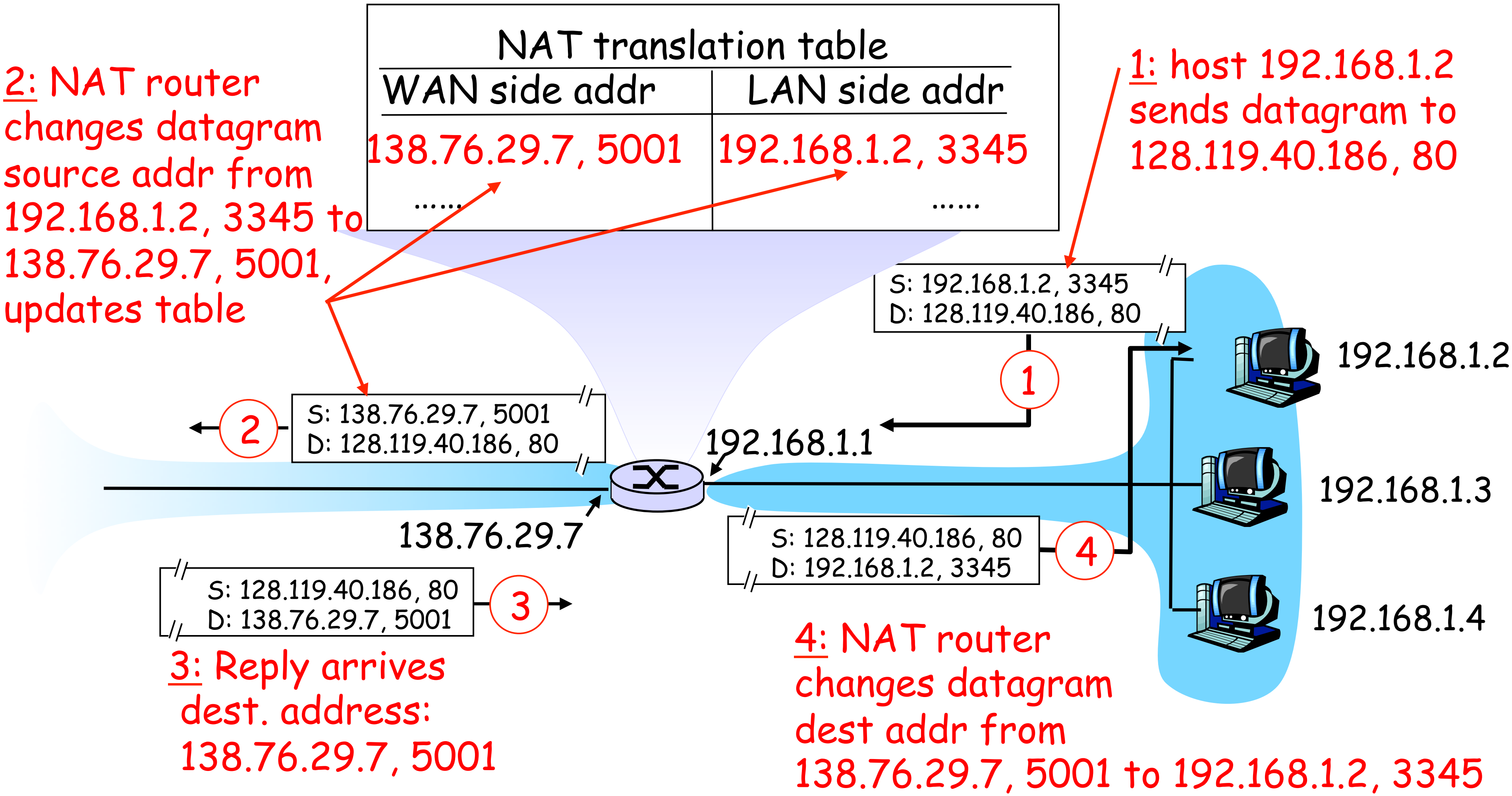
- Implementation: NAT router must:

  - *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



Network Address Translation: Advantages

- No need to be allocated range of addresses from ISP -- one public IP address is used for all devices
  - 16-bit port-number field allows 60,000 simultaneous connections with a single LAN-side address !
  - Can change ISP without changing addresses of devices in local network
  - Can change addresses of devices in local network without notifying outside world
- Devices inside local net not explicitly addressable, visible by outside world (a security plus)

NAT: Network Address Translation

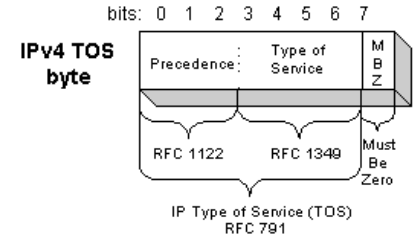
- If both hosts are behind NAT, they will have difficulty establishing connection

- NAT is controversial:
  - Routers should process up to only layer 3
  - Violates end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - Address shortage should instead be solved by having more addresses --- IPv6 !

Outline

- Admin. and recap
- IP addressing
- *IP forwarding*

IP Datagram Format



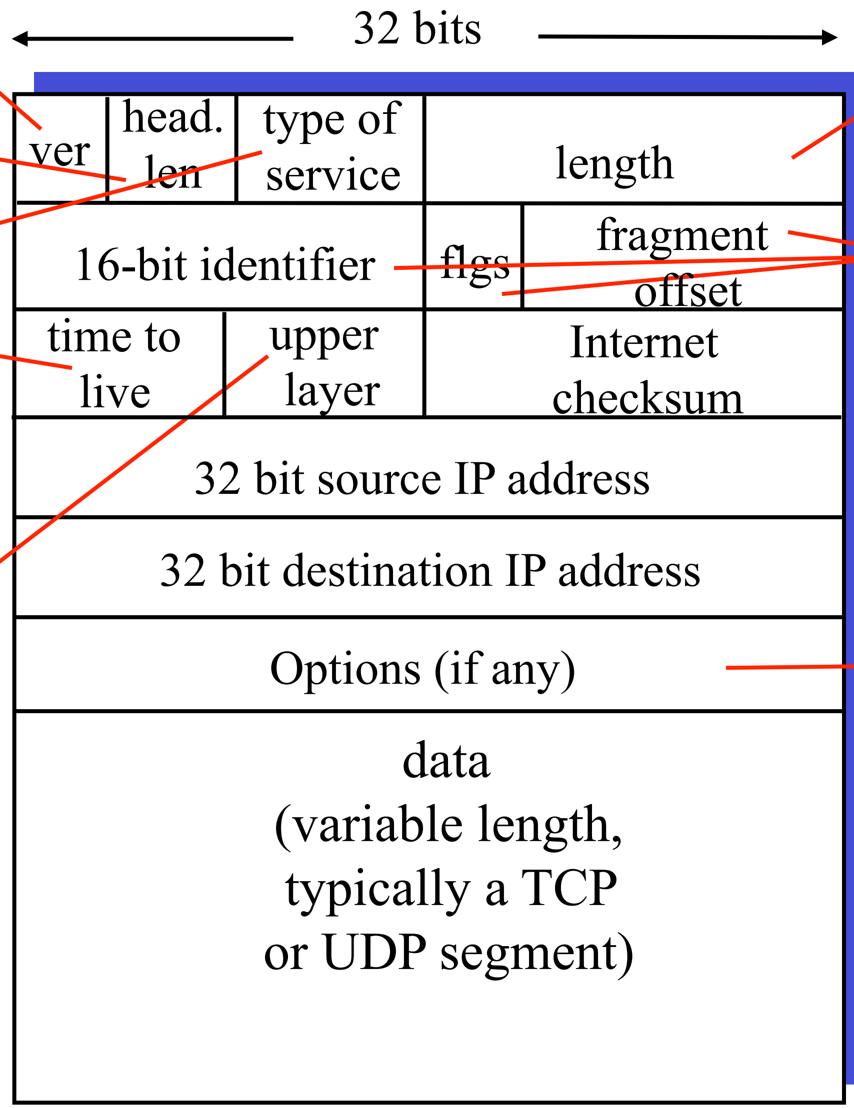
IP protocol version number

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)

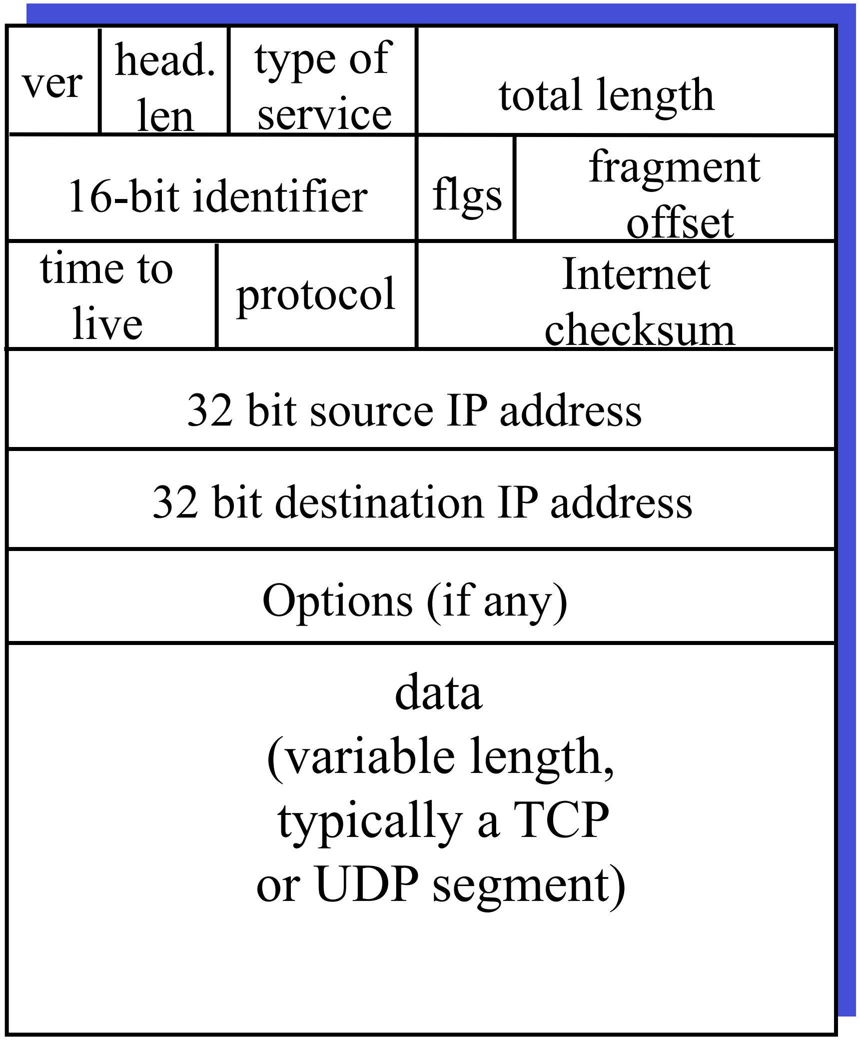
for fragmentation/ reassembly

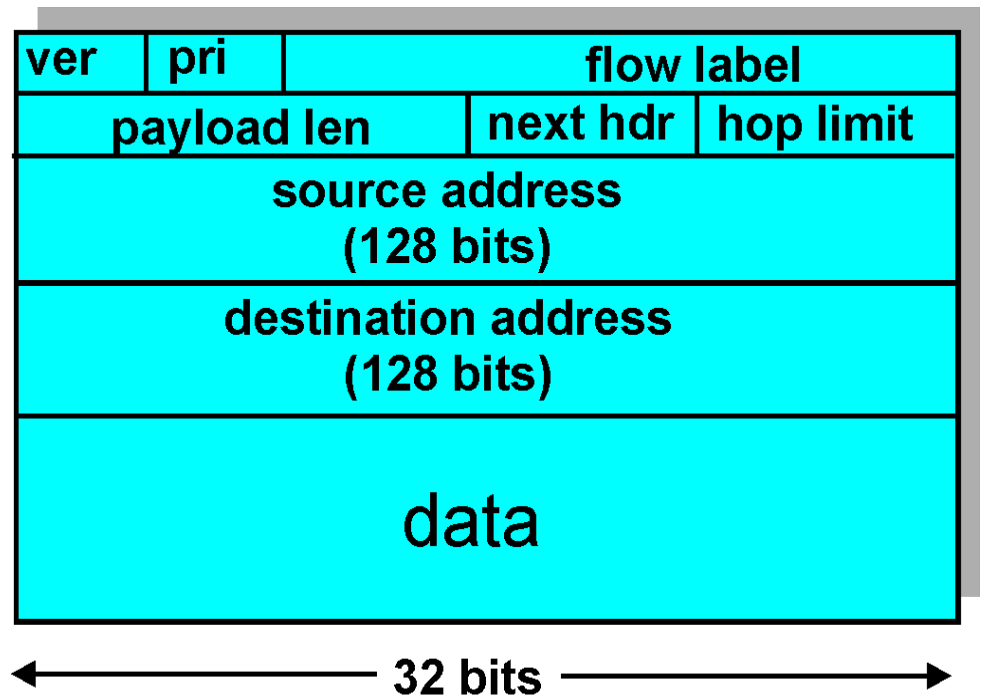
E.g. timestamp, record route taken, specify list of routers to visit.

<u>how much overhead with TCP?</u>

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

IPv4 vs. IPv6


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


ver
pri
flow label
payload len
next hdr
hop limit
source address (128 bits)
destination address (128 bits)
data
32 bits

Data Forwarding: Steps

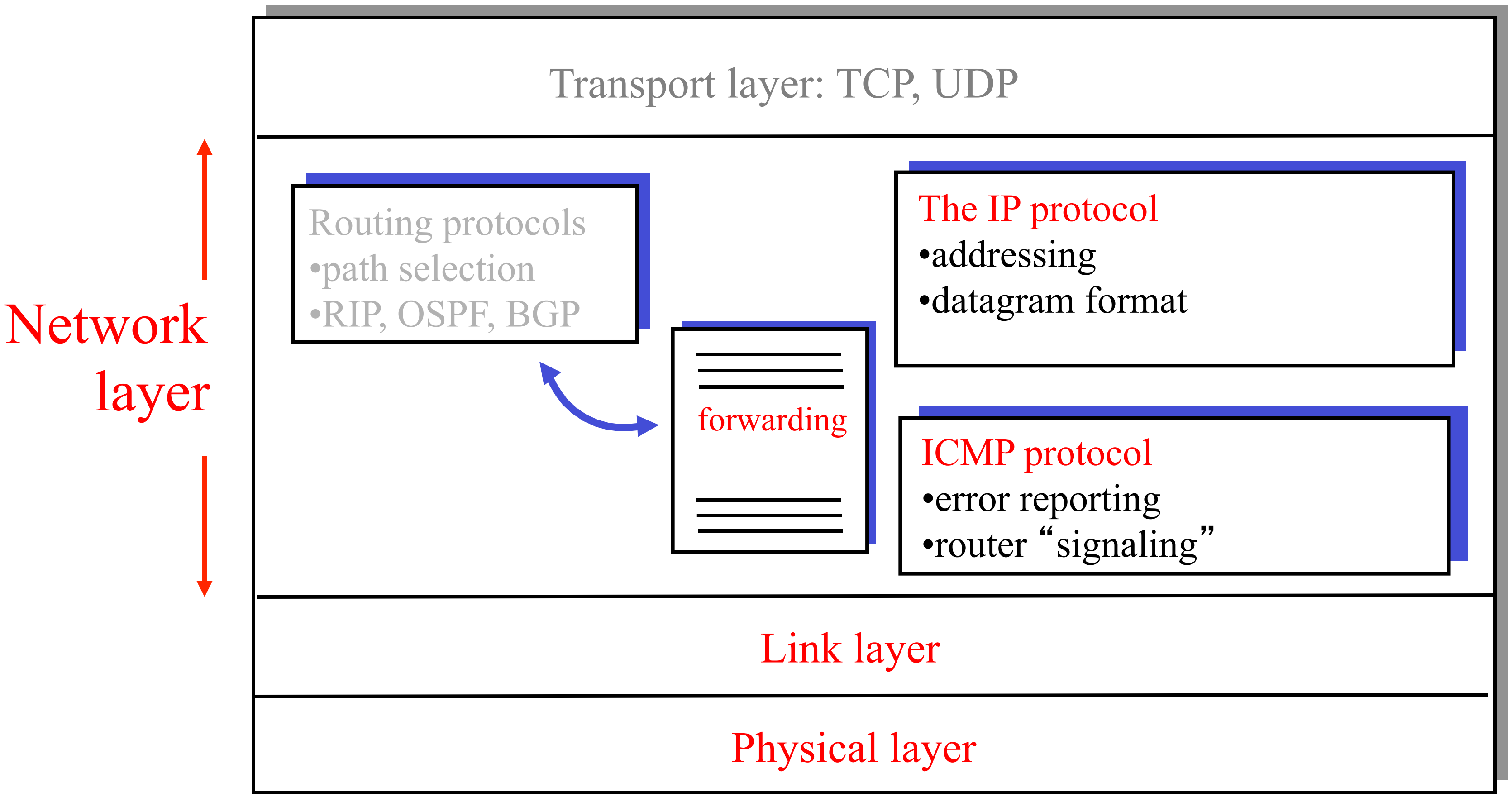
- Error checking, e.g., check header checksum; if error, set up error flag

- Decrement TTL; if TTL == 0, set error flag

- If error, drop the packet, and generate ICMP report

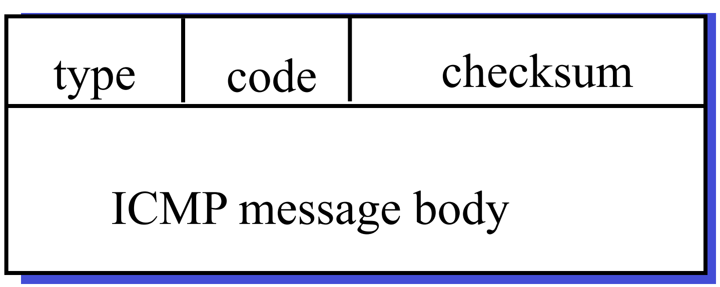
The Network Layer

Host, router network layer functions:



ICMP: Internet Control Message Protocol

- 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 is developed by a clever use of ICMP

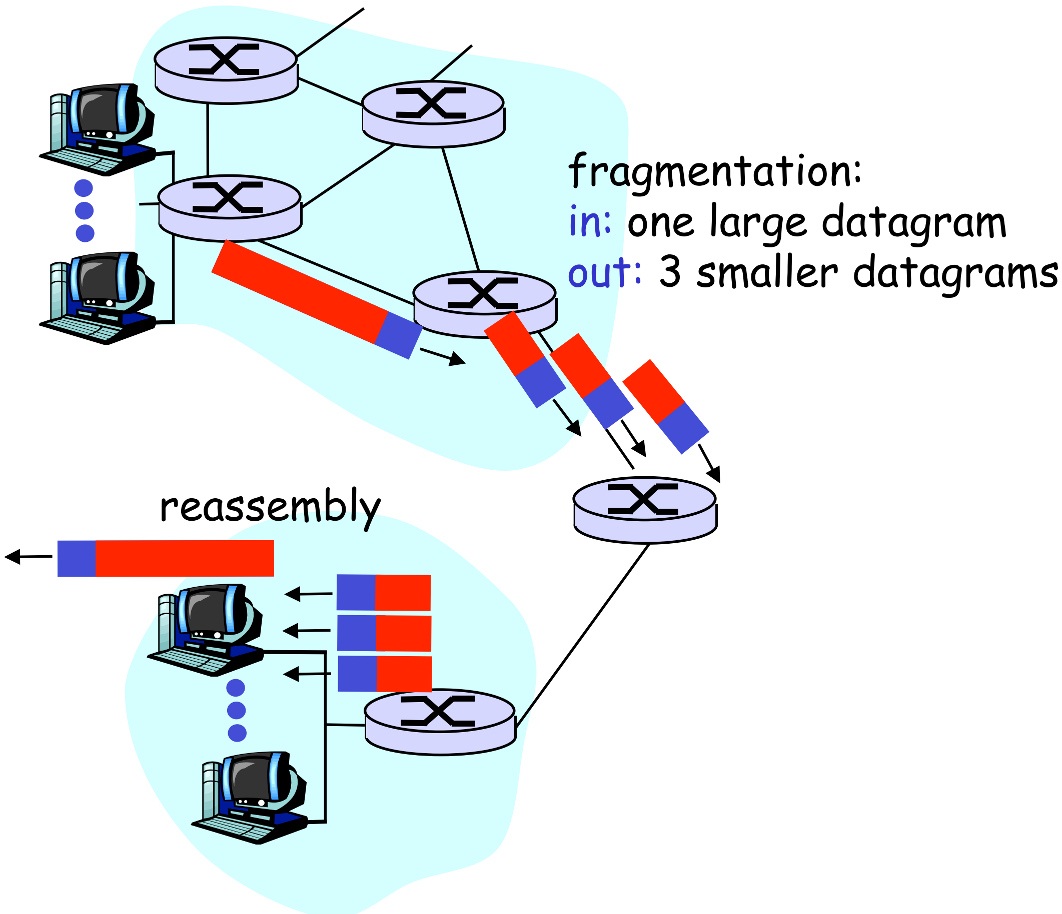
Data Forwarding: Steps

- If no error, look up packet destination address in forwarding table:
  - If datagram for a host on directly attached network, it is the job of the link layer now
  - Otherwise,
    - lookup: find *next-hop router,* and its outgoing interface
    - if needed, do fragmentation
    - forward packet to outgoing interface (to the next hop neighbor)

try %netstat -rn to see the forwarding table

IP Fragmentation & Reassembly

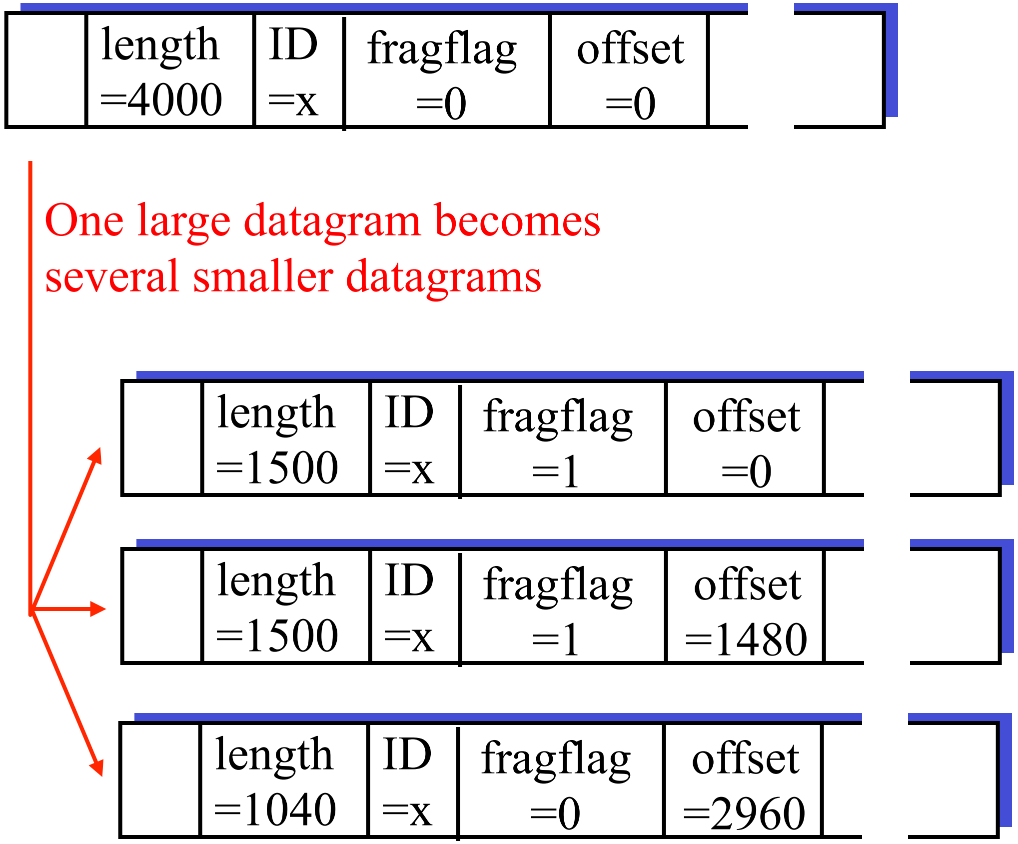
- Network links have MTU (max.transfer size) - largest possible link-level frame.
  - Different link types, different MTUs, e.g. Ethernet MTU is 1500 bytes
- Large IP datagram divided ("fragmented")
  - One datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments



IP Fragmentation and Reassembly

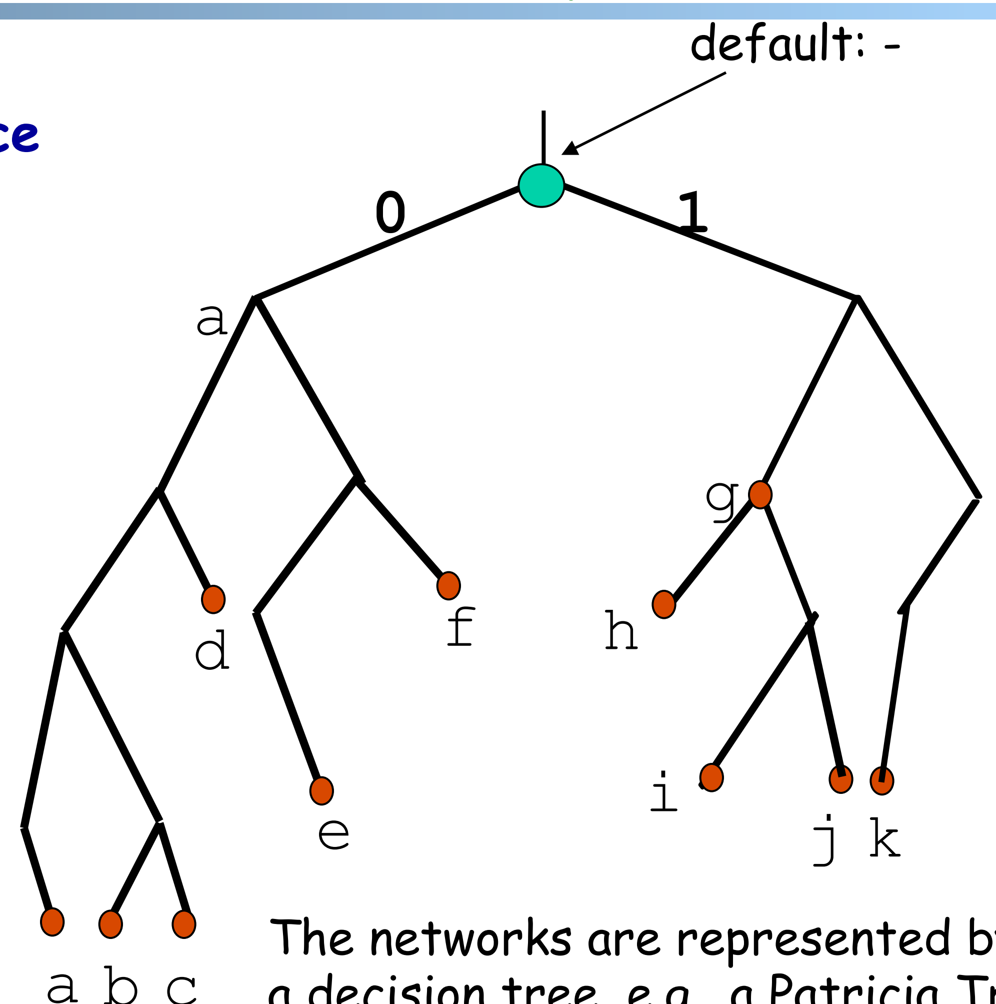
Example

- 4000 byte datagram
- MTU = 1500 bytes



Forwarding Look up

#	prefix	interface
a)	00001	
b)	00010	
c)	00011	
d)	001	
e)	0101	
f)	011	
g)	10	
h)	100	
i)	1010	
j)	1011	
k)	1100	



The networks are represented by a decision tree, e.g., a Patricia Trie to look for the longest match of the destination address

What A Router Looks Like: Outside

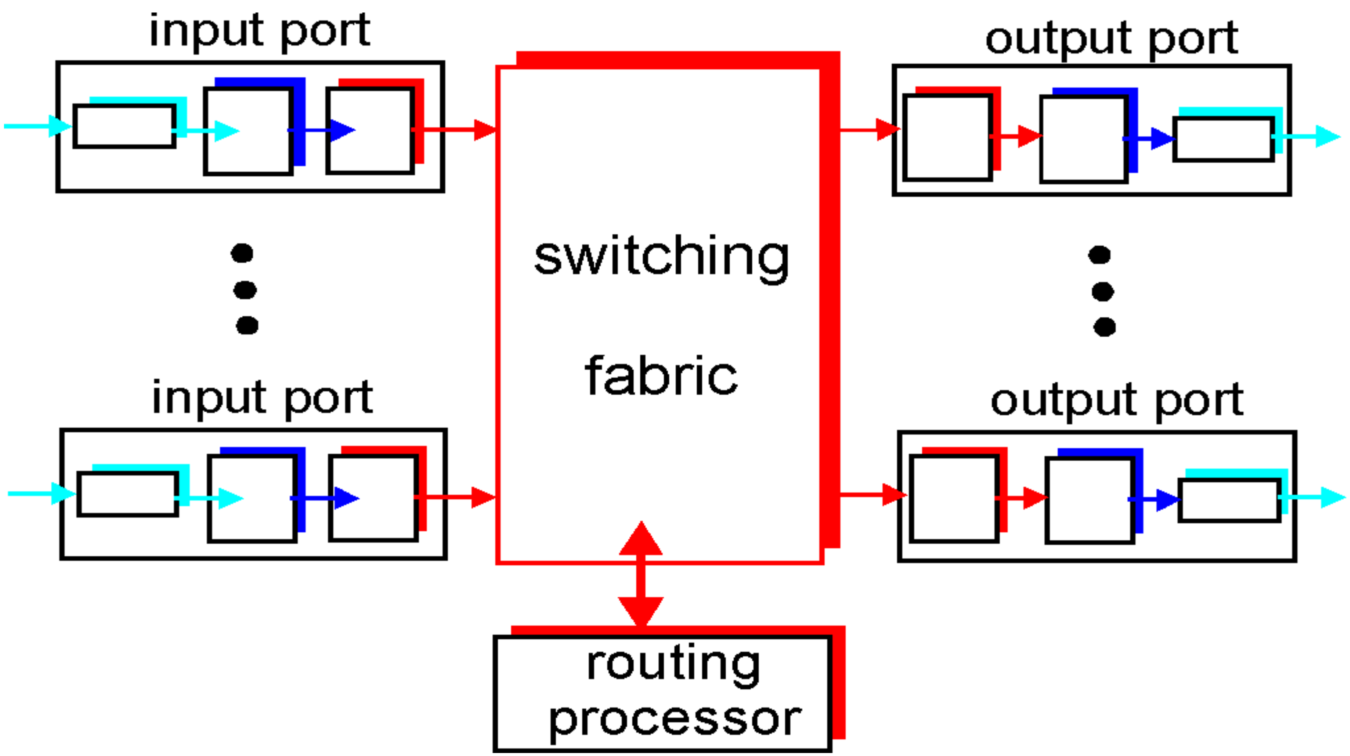
Cisco CRS-1:
each one: 16 cards at 40 Gbps each
upto 1152 cards at a total of 92 Tbps

Juniper T1600

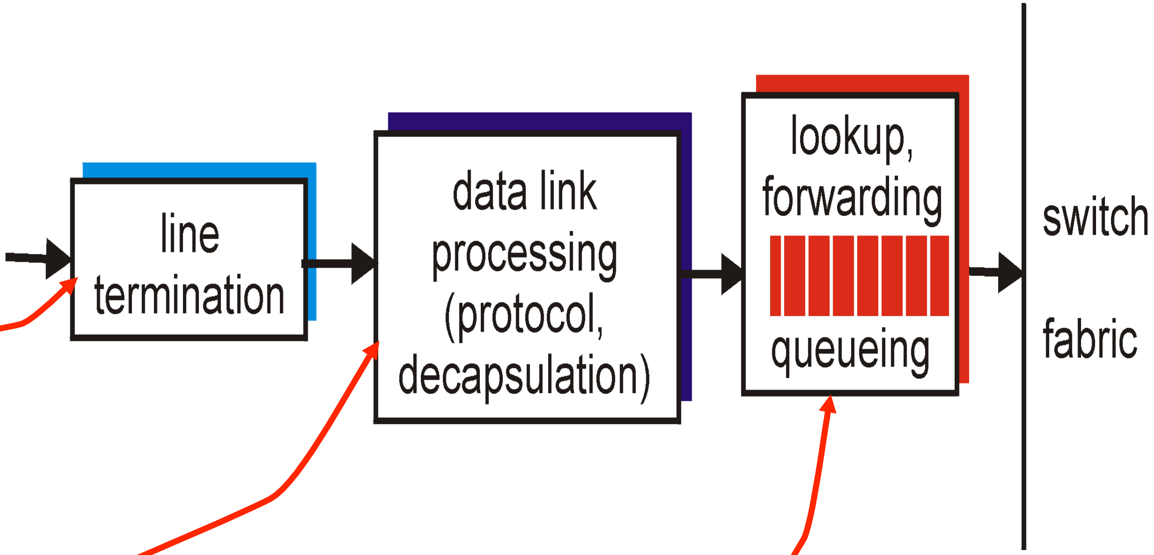
Look Inside a Router

Two key router functions:

- Run routing algorithms/protocol (RIP, OSPF, BGP)
- *Switching* datagrams from incoming to outgoing ports



Input Port Functions



physical layer:
bit-level reception

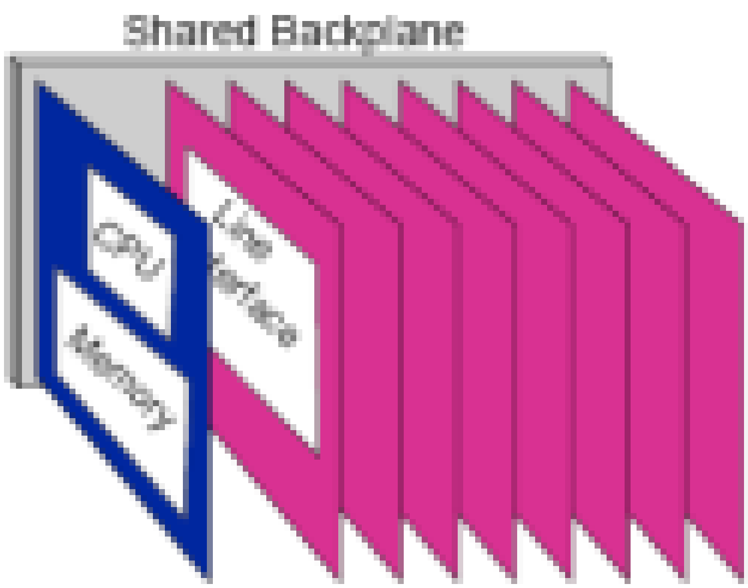
data link layer:
e.g., Ethernet

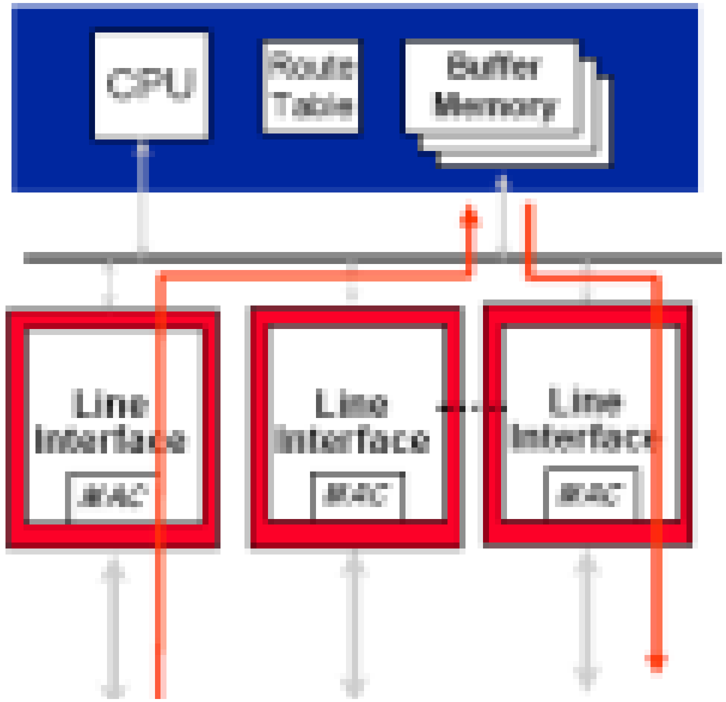
network layer:
lookup output port
using forwarding table

Switching Via Memory

First generation or home routers: packet handled by system's (single) CPU

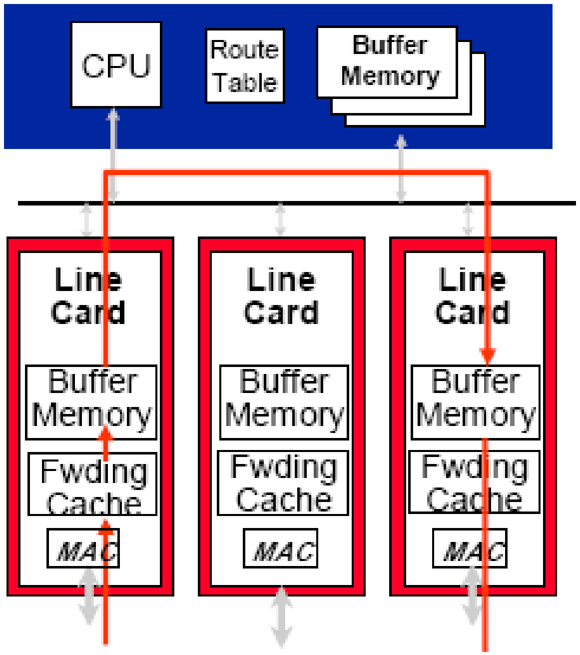
- Bottleneck: shared memory access



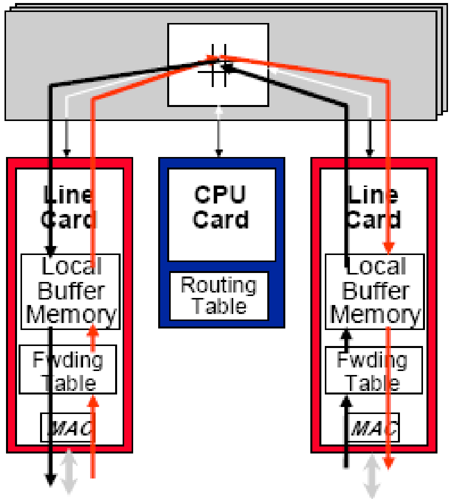


Switching Via a Bus

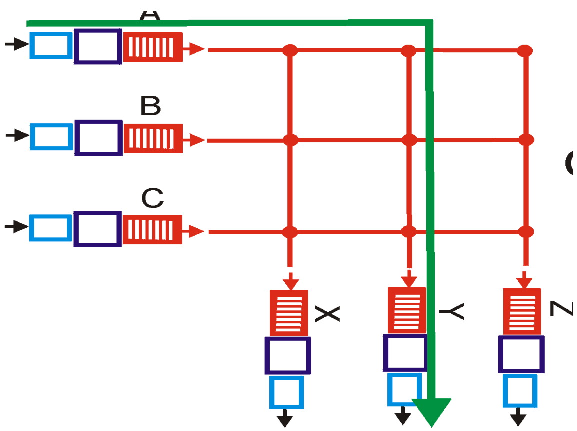
- Datagram from input port memory to output port memory via a shared bus
- Bottleneck: bus contention
  - < 5Gbps, e.g., 1 Gbps bus, Cisco 1900: sufficient speed for access and enterprise routers (not regional or backbone)

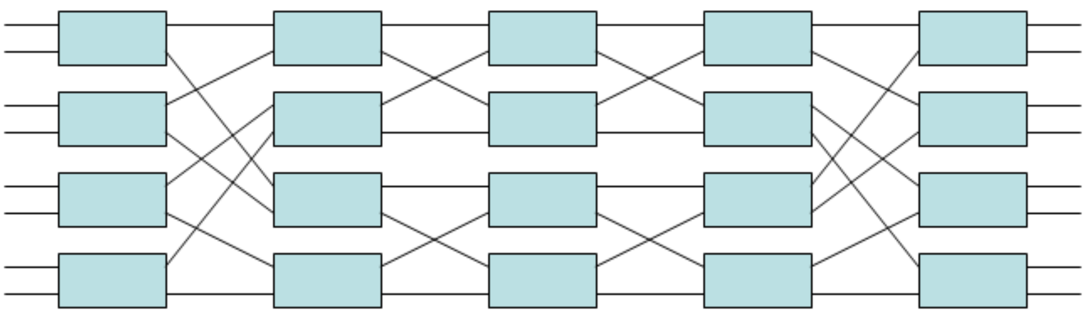


Switching Via An Interconnection Network



- Overcome bus bandwidth limitations
- fragmenting datagram into fixed length cells, switch cells through the fabric.
- Crossbar, Banyan networks, and others



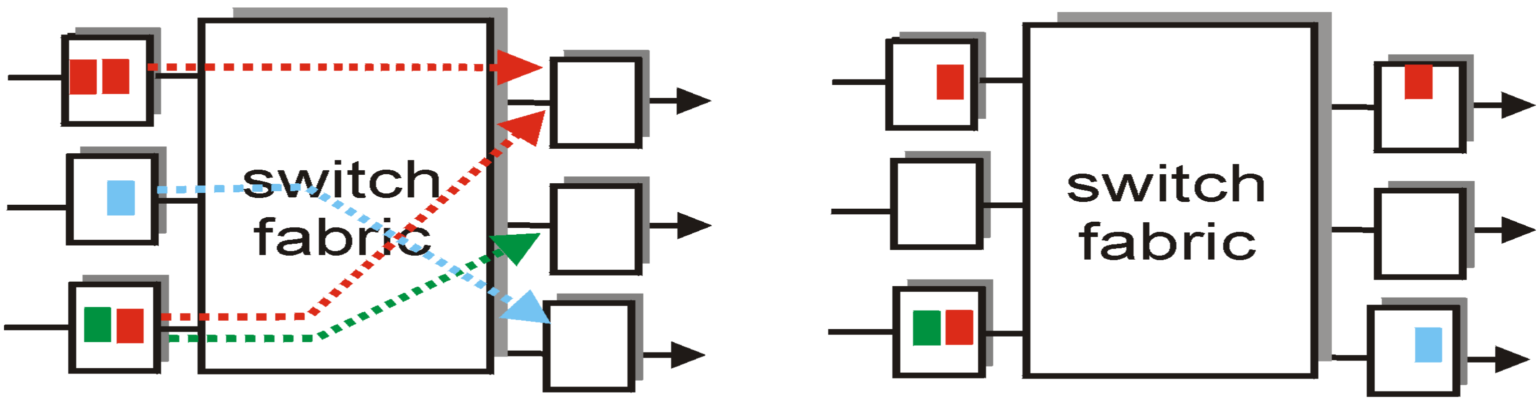


- Cisco CRS-1: using Benes connect

http://en.wikipedia.org/wiki/Clos_network#Clos_networks_with_more_than_three_stages

New Potential Bottleneck: Output Ports

- Due to output port contention and head-of-the-Line (HOL) blocking (i.e., queued datagram at front of queue prevents others in queue from moving forward)

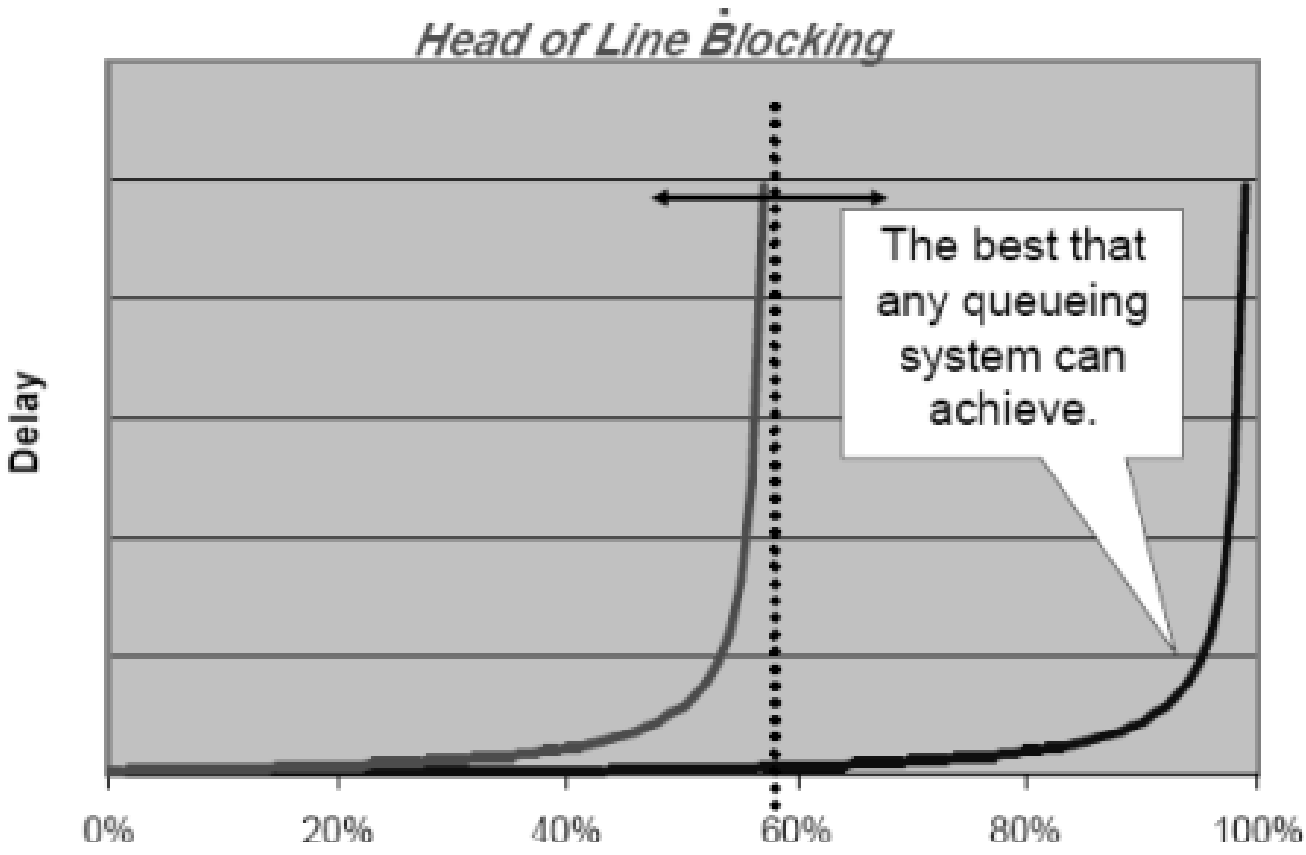


output port contention at time t - only one red packet can be transferred

green packet experiences HOL blocking

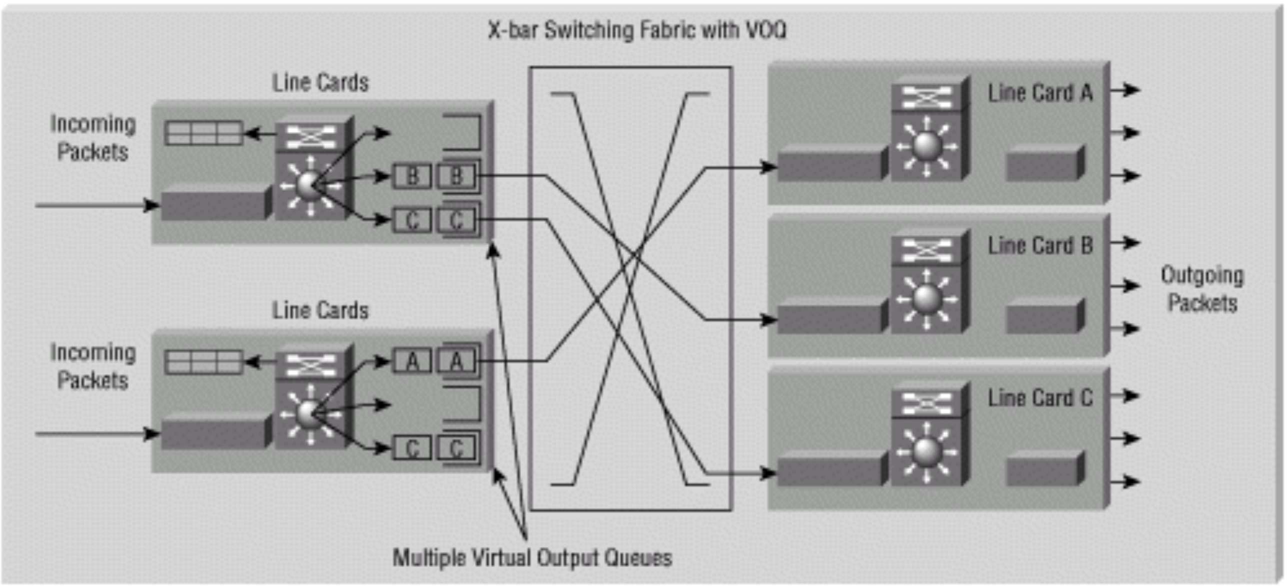
Head-of-Line Blocking Limits Thrput

- Due to output-port contention and HOL blocking, the stable throughput is only around 2 - sqrt(2) = 0.586 of line speed !



Avoiding Port Contention and HOB

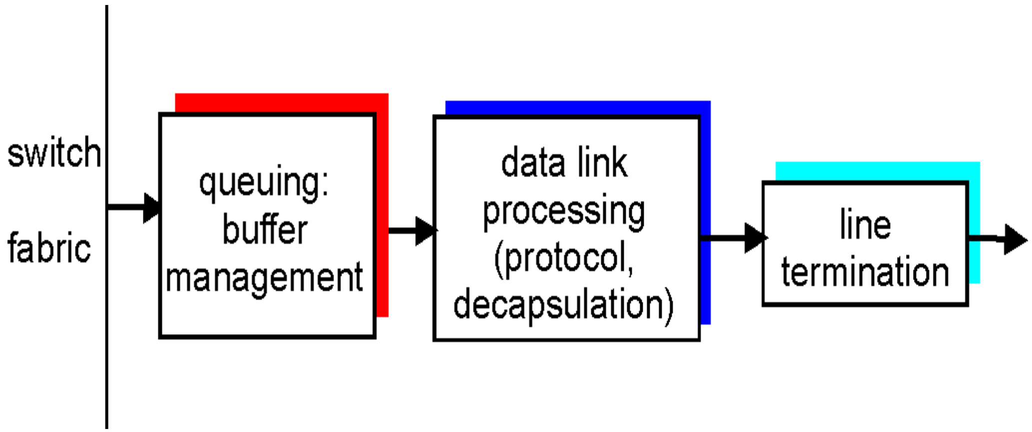
- Virtual output queueing



- Input/output ports matching algorithm
- Switch fabric speedup, e.g., two cells to one output port

For more details: http://www.cisco.com/warp/public/63/arch12000-swfabric.html

Output Ports



- *Buffering* required when datagrams arrive from fabric faster than the transmission rate
- *Queueing (delay) and loss due to output port buffer overflow !*
- *Scheduling and queue/buffer management* choose among queued datagrams for transmission

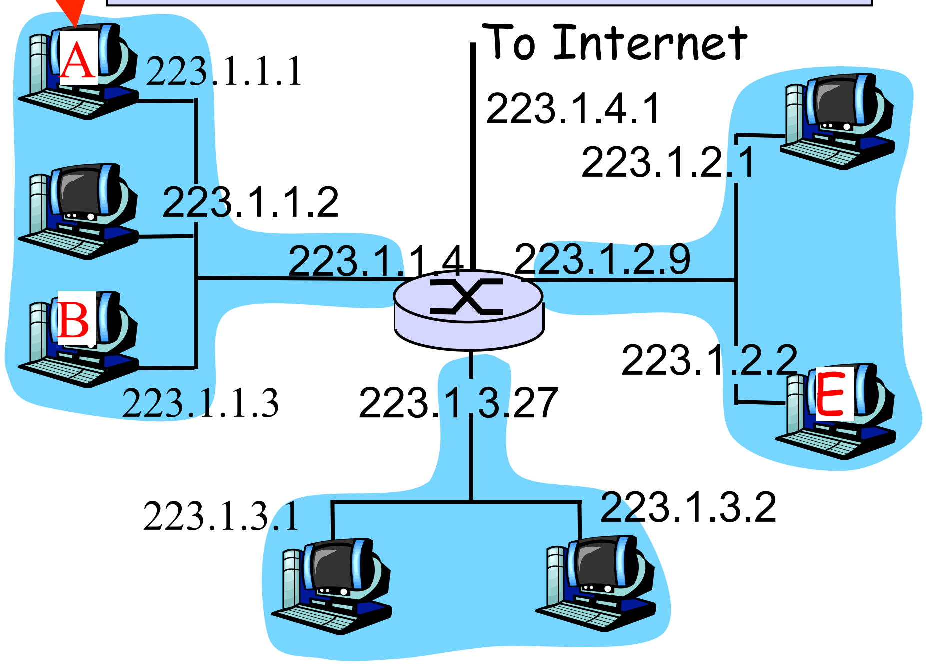
Example 1 (same network): A->B

	src	dst	
misc fields	223.1.1.1	223.1.1.3	data

- Look up dest address
- Find dest is on same net
- Link layer will send the datagram directly inside a link-layer frame

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1/24		1
223.1.2/24	223.1.1.4	2
223.1.3/24	223.1.1.4	2
0.0.0.0/0	223.1.1.4	-



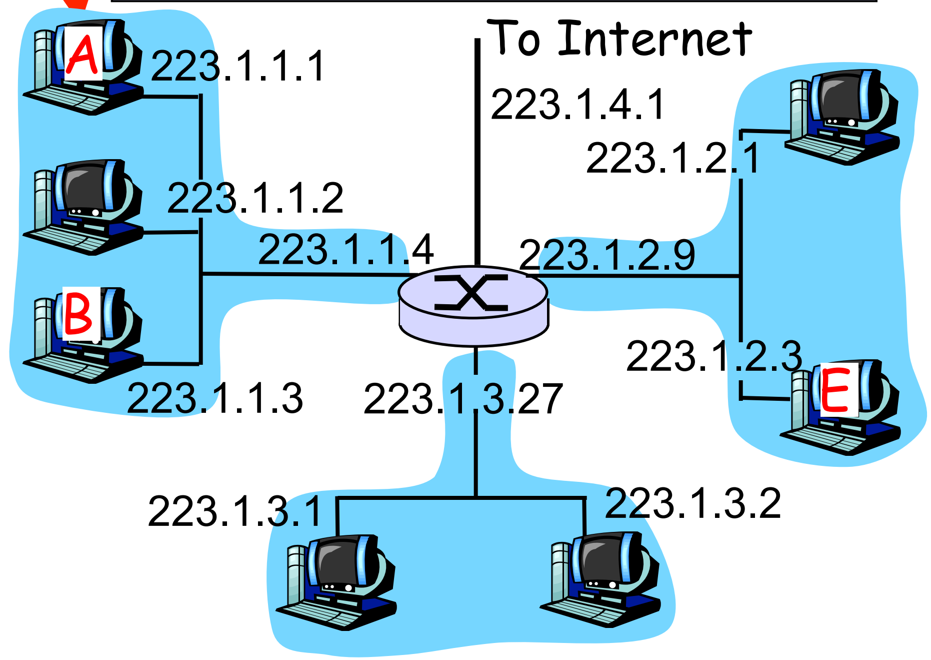
Example 2 (Different Networks): A-> E

	src	dst	
misc fields	223.1.1.1	223.1.2.3	data

- Look up dest address in forwarding table
- Routing table: next hop router to dest is 223.1.1.4
- Link layer sends datagram to router 223.1.1.4 inside a link-layer frame
  - The dest. of the link layer frame is 223.1.1.4

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1/24		1
223.1.2/24	223.1.1.4	2
223.1.3/24	223.1.1.4	2
0.0.0.0/0	223.1.1.4	-



Example 2 (Different Networks): A-> E

misc fields	223.1.1.1	223.1.2.3	data

forwarding table in router

Dest. Net	router	Nhops	interface
223.1.1/24	-	1	223.1.1.4
223.1.2/24	-	1	223.1.2.9
223.1.3/24	-	1	223.1.3.27
0.0.0.0/0	-	-	223.1.4.1

Arriving at 223.1.1.4, destined for 223.1.2.3

- Look up dest address in router's forwarding table
- E on *same* network as router's interface 223.1.2.9
  - Router & E directly attached
- Link layer sends datagram to 223.1.2.3 inside link-layer frame via interface 223.1.2.9
- Datagram arrives at 223.1.2.3!