COMPUTER NETWORKS

TEAM NETWORKS
Department of Computer Science and Engineering

Unit – 3 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What’s Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

Unit – 3 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

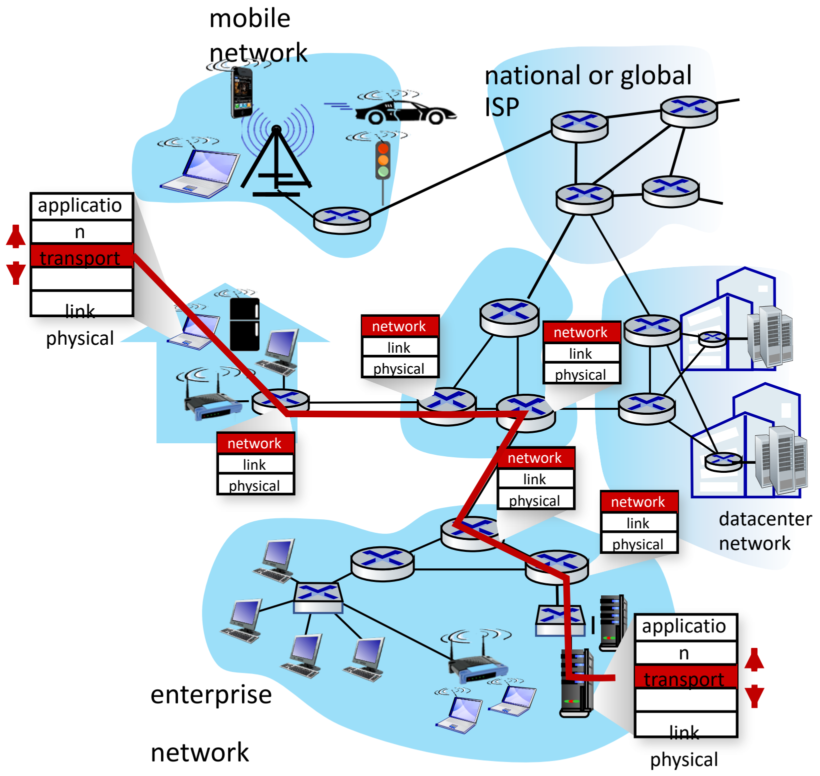
4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

COMPUTER NETWORKS

Network-layer Services and Protocol

- transport segment from sending to receiving host
  - sender: encapsulates segments into datagrams, passes to link layer
  - receiver: delivers segments to transport layer protocol
- network layer protocols in *every Internet device*: hosts, routers
- routers:
  - examines header fields in all IP datagrams passing through it
  - moves datagrams from input ports to output ports to transfer datagrams along end-end path



Two Key Network-layer Functions

network-layer functions:

- *forwarding:* move packets from a router's input link to appropriate router output link

- *routing:* determine route taken by packets from source to destination
  - *routing algorithms*

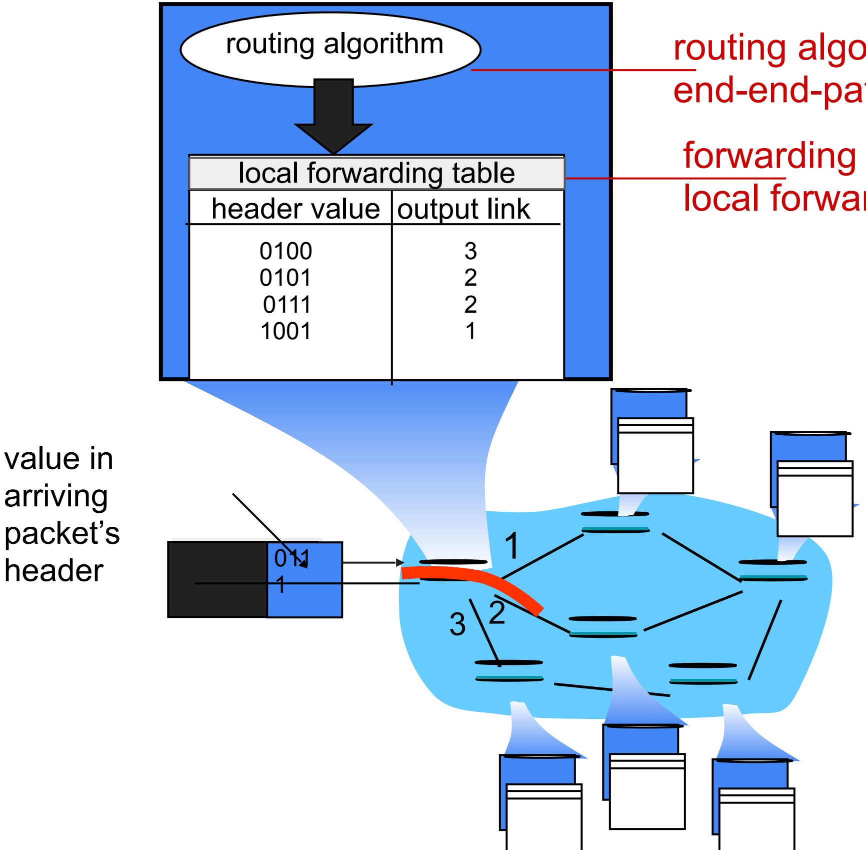
analogy: taking a trip

- *forwarding:* process of getting through single interchange
- *routing:* process of planning trip from source to destination

forwarding

routing

Interplay between Routing and Forwarding



routing algorithm determines end-end-path through network

forwarding table determines local forwarding at this router

Connection Setup

- 3rd important function in *some* network architectures:
  - 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

Q: What *service model* for "channel" transporting datagrams from sender to receiver?

example services for *individual* datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

example services for a *flow* of datagrams:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing

Network-layer Service Model

Network Architecture	Service Model	Quality of Service (QoS) Guarantees?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no

Internet "best effort" service model

No guarantees on*:*

i. successful datagram delivery to destination
ii. timing or order of delivery
iii. bandwidth available to end-end flow

Network-layer Service Model

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no
ATM	Constant Bit Rate	Constant rate	yes	yes	yes
ATM	Available Bit Rate	Guaranteed min	no	yes	no
Internet	Intserv Guaranteed (RFC 1633)	yes	yes	yes	yes
Internet	Diffserv (RFC 2475)	possible	possibly	possibly	no

Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

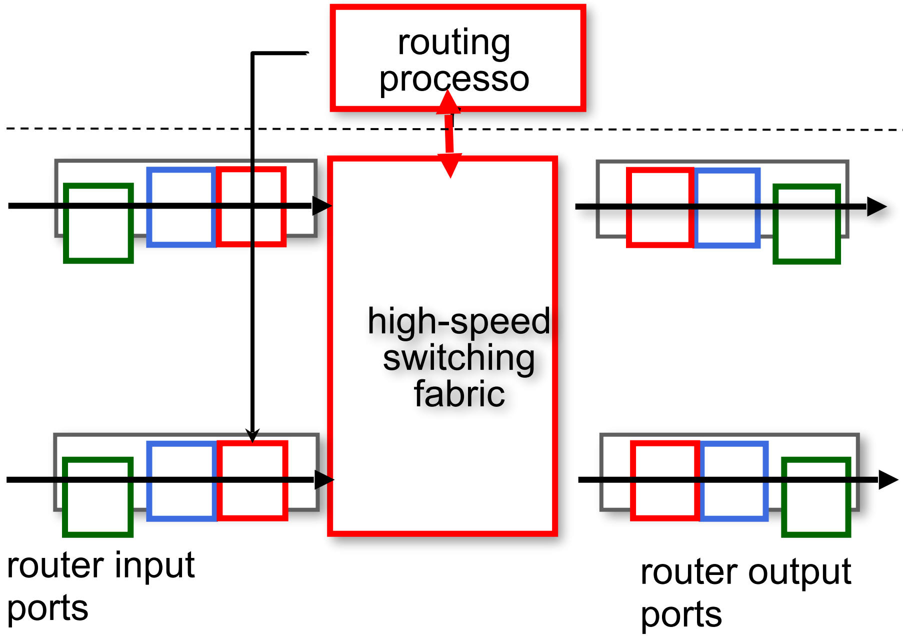
4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

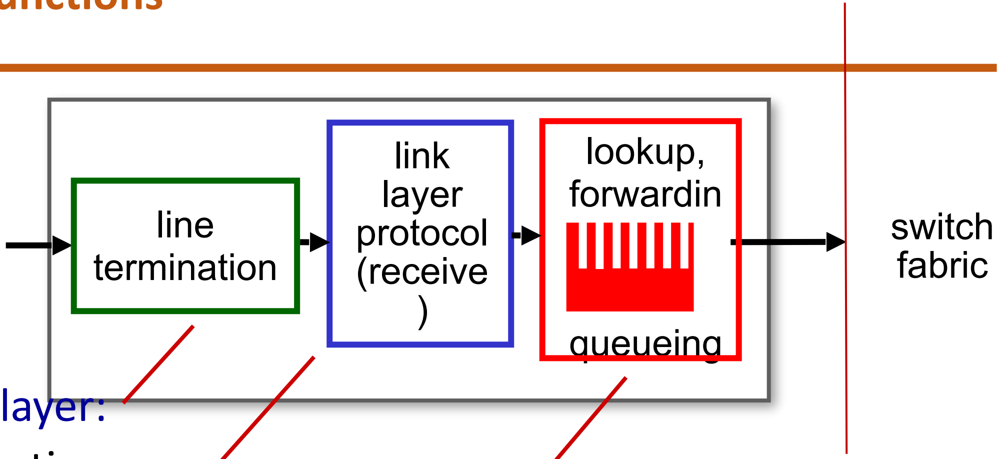
high-level view of generic router architecture:



routing, management
control plane (software)
operates in millisecond
time frame

forwarding data plane
(hardware) operates
in nanosecond
timeframe

Input port functions



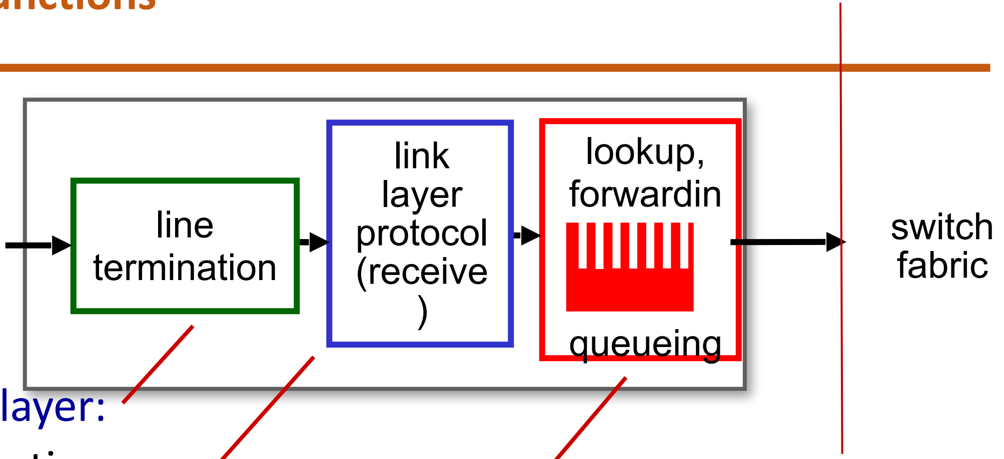
physical layer:
bit-level reception

link layer:
e.g., Ethernet
(chapter 6)

decentralized switching:

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

Input port functions



physical layer:
bit-level reception

link layer:
e.g., Ethernet
(chapter 6)

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- destination-based forwarding: forward based only on destination IP address (traditional)
- generalized forwarding: forward based on any set of header field values

Destination-based Forwarding

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through	0
11001000 00010111 00010000 00000100 through 11001000 00010111 00010000 00000111	3
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

longest prefix match

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:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

longest prefix match

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	[illegible]	00011***	********	2
otherwise				3

match!

examples:

11001000	00010111	00010110	10100001	which interface?
11001000	00010111	00011000	10101010	which interface?

Longest prefix matching

longest prefix match

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

match!

examples:

11001000 00010111 00010110 — which interface?

11001000 00010111 00011000 — which interface?

10100001

10101010

longest prefix match

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

match!

examples:

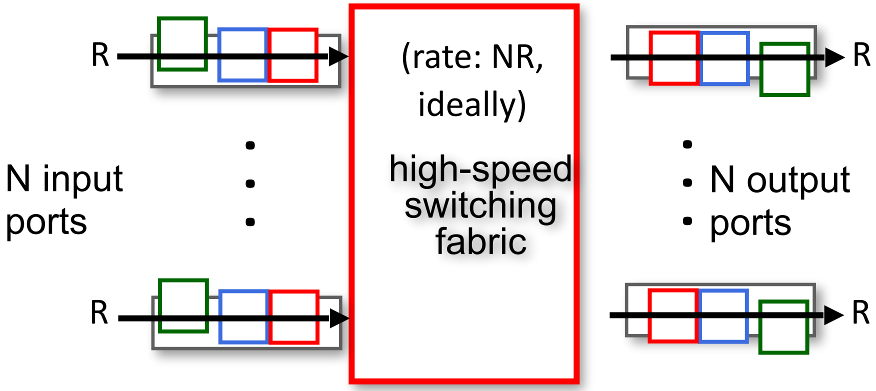
11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

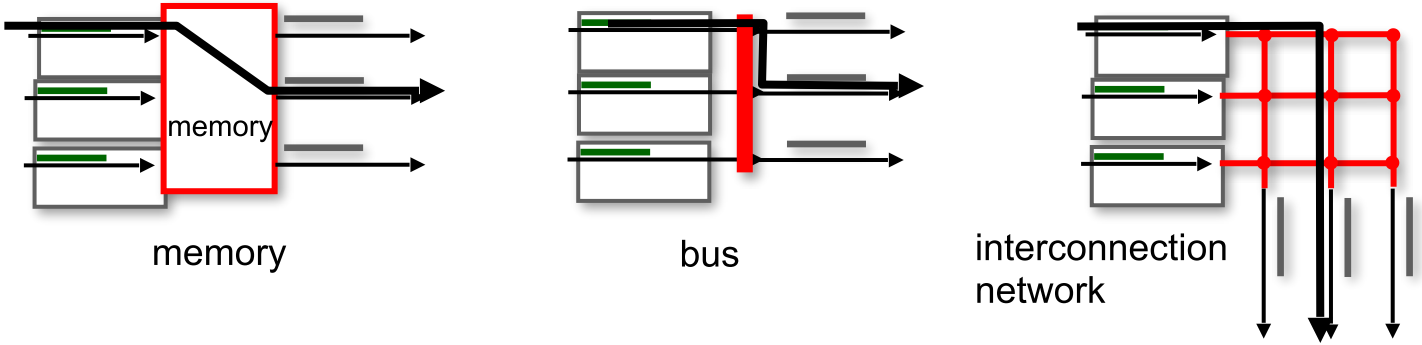
- we'll see *why* longest prefix matching is used shortly, when we study addressing
- longest prefix matching: often performed using ternary content addressable memories (TCAMs)
  - *content addressable:* present address to TCAM: retrieve address in one clock cycle, regardless of table size
  - Cisco Catalyst: ~1M routing table entries in TCAM

- switching rate: rate at which packets can be transferred from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable



Switching Fabrics

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



memory

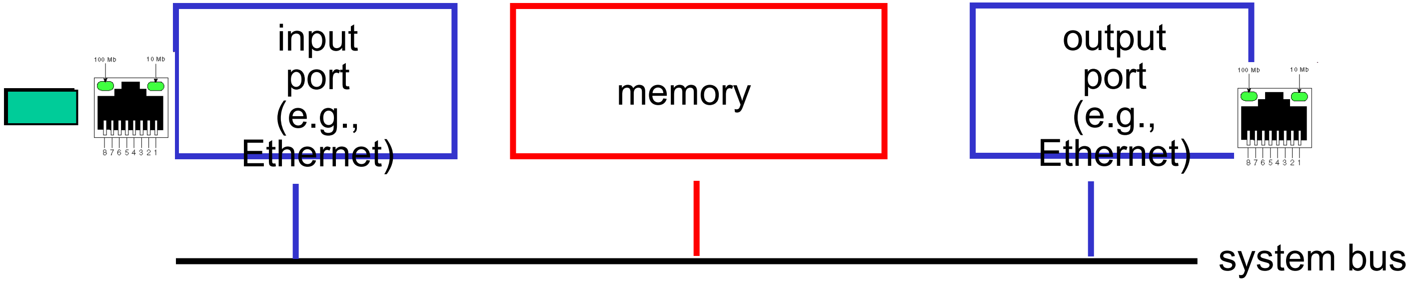
bus

interconnection network

Switching via Memory

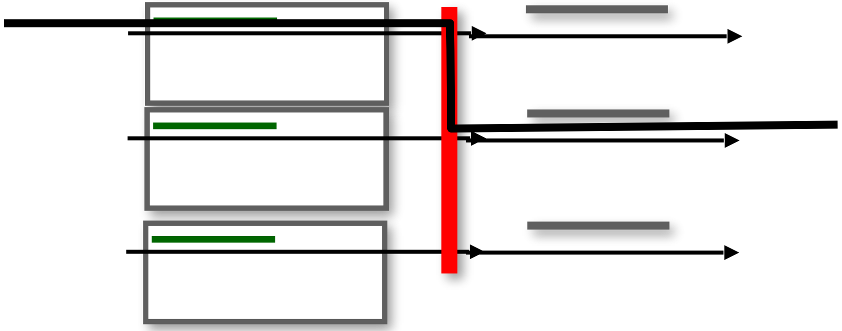
first generation routers:

- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth (2 bus crossings per datagram)



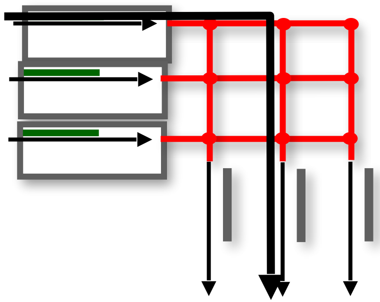
Switching via a Bus

- 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 routers



Switching via interconnection network

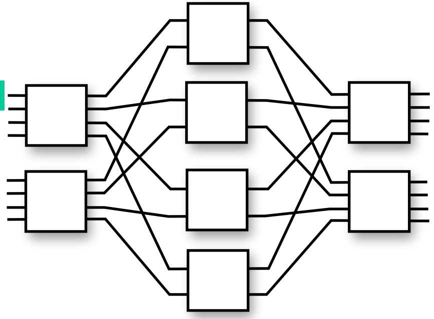
- Crossbar, Clos networks, other interconnection nets initially developed to connect processors in multiprocessor



3x3 crossbar

- multistage switch: *nxn* switch from multiple stages of smaller switches

- exploiting parallelism:
  - fragment datagram into fixed length cells on entry
  - switch cells through the fabric, reassemble datagram at exit

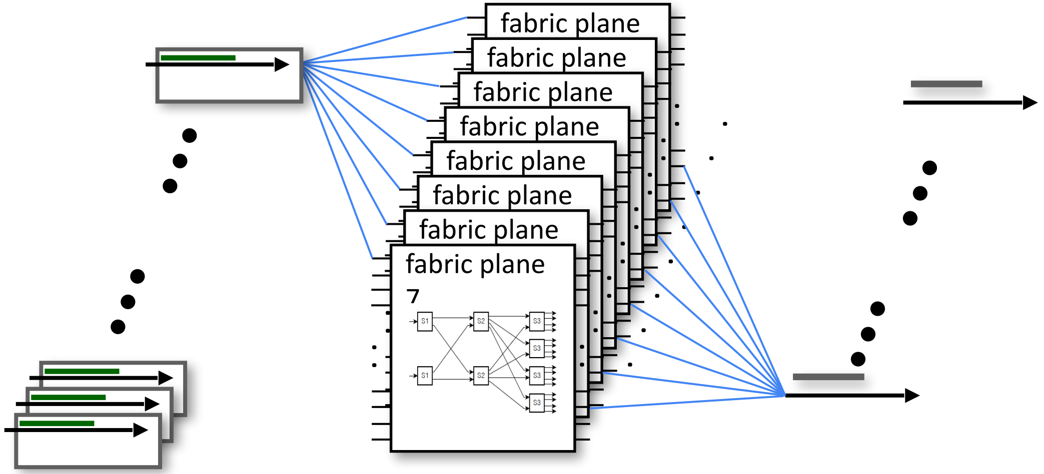


8x8 multistage switch built from smaller-sized switches

Switching via interconnection network

- scaling, using multiple switching "planes" in parallel:
  - speedup, scaleup via parallelism

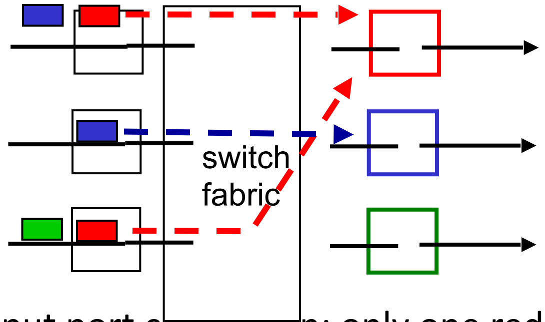
- Cisco CRS router:
  - basic unit: 8 switching planes
  - each plane: 3-stage interconnection network
  - up to 100's Tbps switching capacity



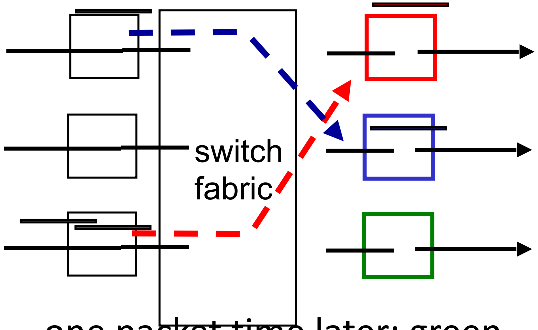
Input port queuing

- If switch fabric slower than input ports combined -> queueing may occur at input queues
  - queueing delay and loss due to input buffer overflow!

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



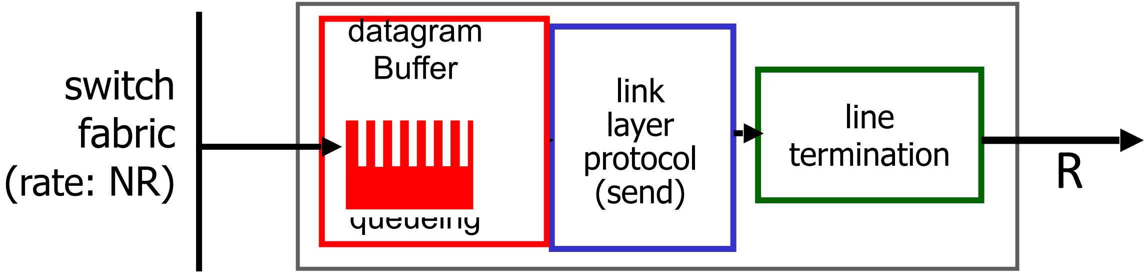
output port contention: only one red datagram can be transferred. lower red packet is *blocked*



one packet time later: green packet experiences HOL blocking

Output port queuing

This is a really important slide



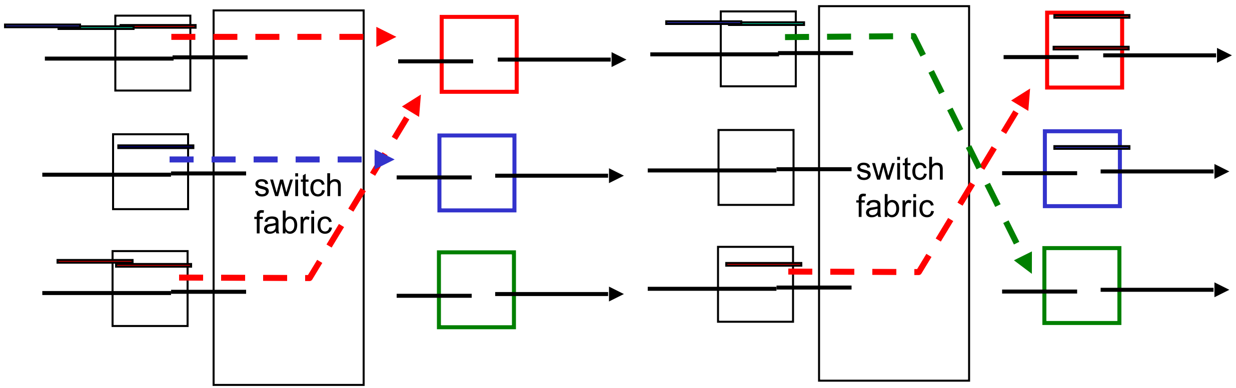
- *Buffering* required when datagrams arrive from fabric faster than link transmission rate. *Drop policy:* which datagrams to drop if no free buffers?

Datagrams can be lost due to congestion, lack of buffers

- *Scheduling discipline* chooses among queued datagrams for transmission

Priority scheduling – who gets best performance, network neutrality

Output port queuing



at *t*, packets more from input to output

one packet time later

- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

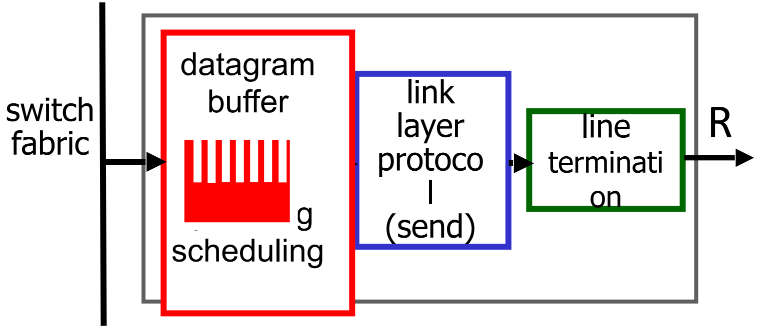
How much buffering?

- RFC 3439 rule of thumb: average buffering equal to “typical” RTT (say 250 msec) times link capacity C
  - e.g., C = 10 Gbps link: 2.5 Gbit buffer
- more recent recommendation: with *N* flows, buffering equal to

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

- but *too* much buffering can increase delays (particularly in home routers)
  - long RTTs: poor performance for realtime apps, sluggish TCP response
  - recall delay-based congestion control: “keep bottleneck link just full enough (busy) but no fuller”

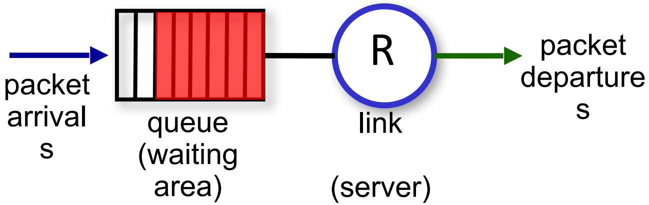
Buffer Management



buffer management:

- drop: which packet to add, drop when buffers are full
  - tail drop: drop arriving packet
  - priority: drop/remove on priority basis

Abstraction: queue



- marking: which packets to mark to signal congestion (ECN, RED)

Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

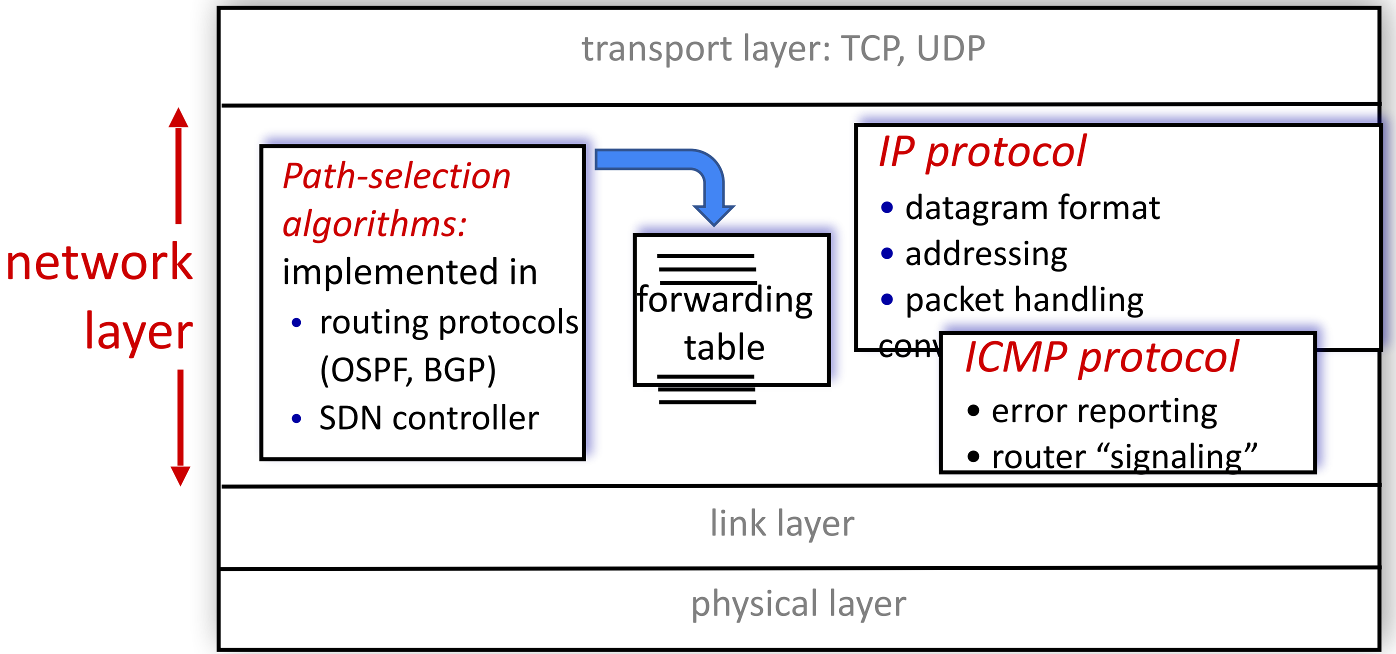
4.2 What's Inside a Router?

4.3 Switching

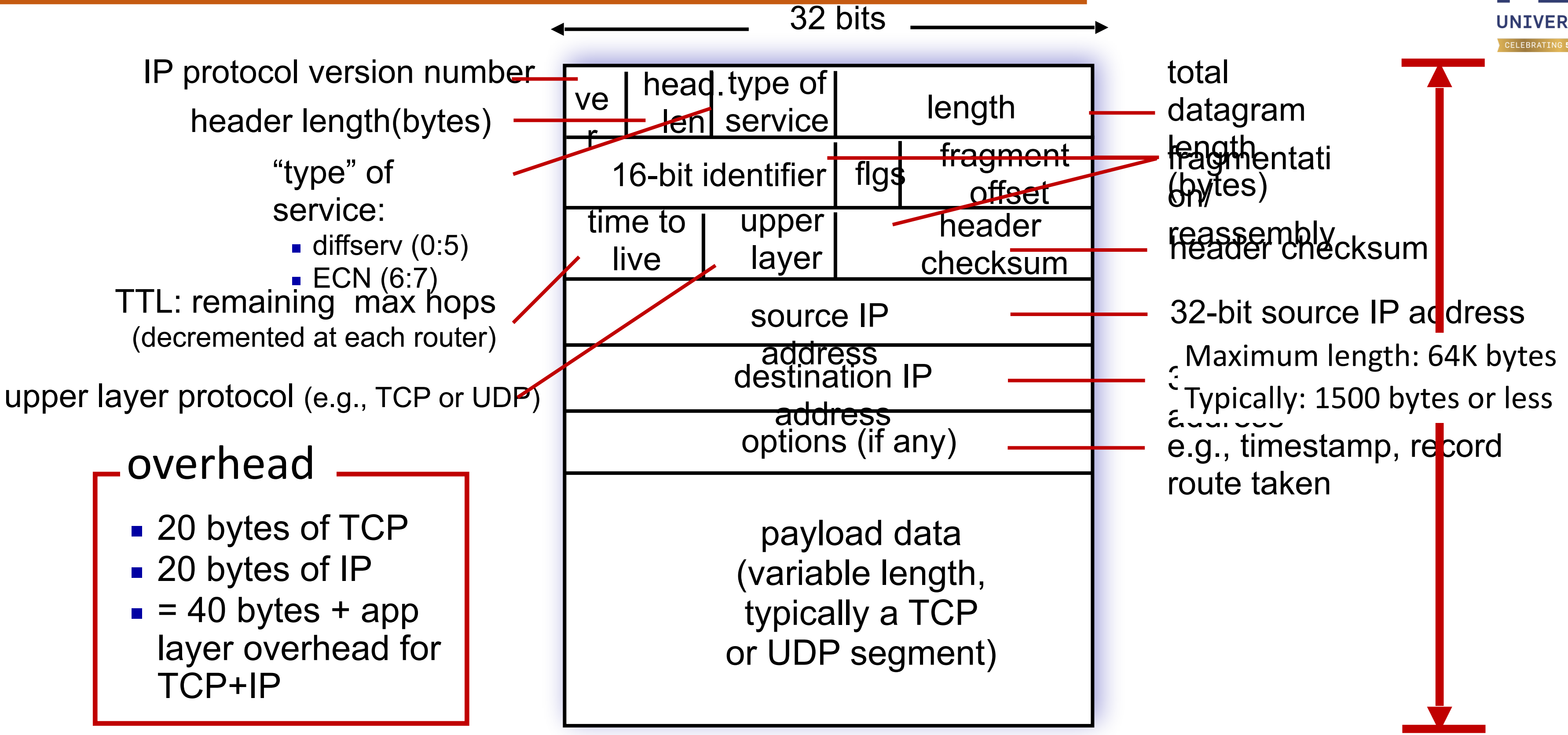
4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

host, router network layer functions:



IP Datagram Format



overhead

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

Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

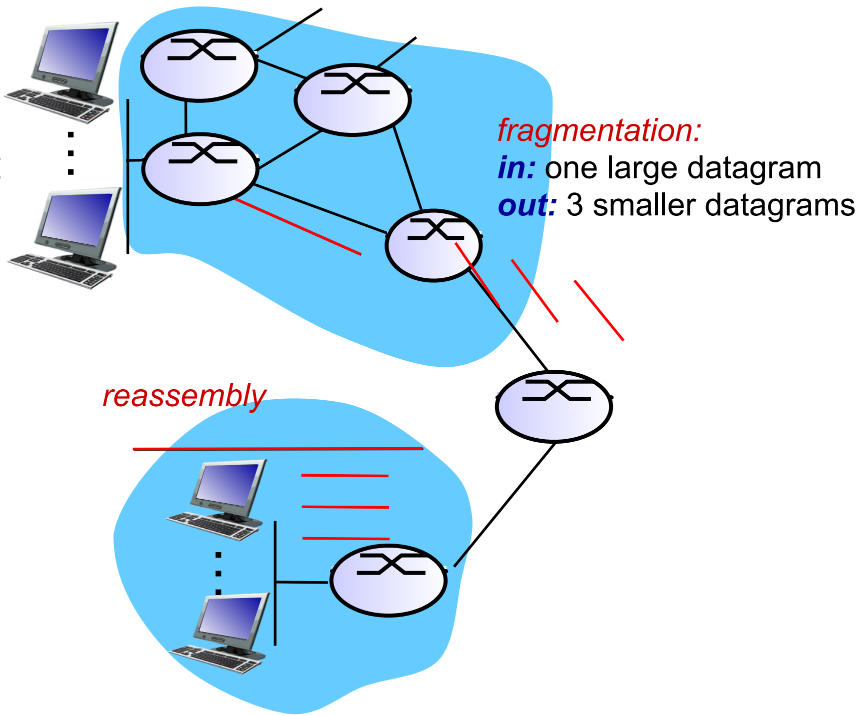
4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

IP fragmentation, reassembly

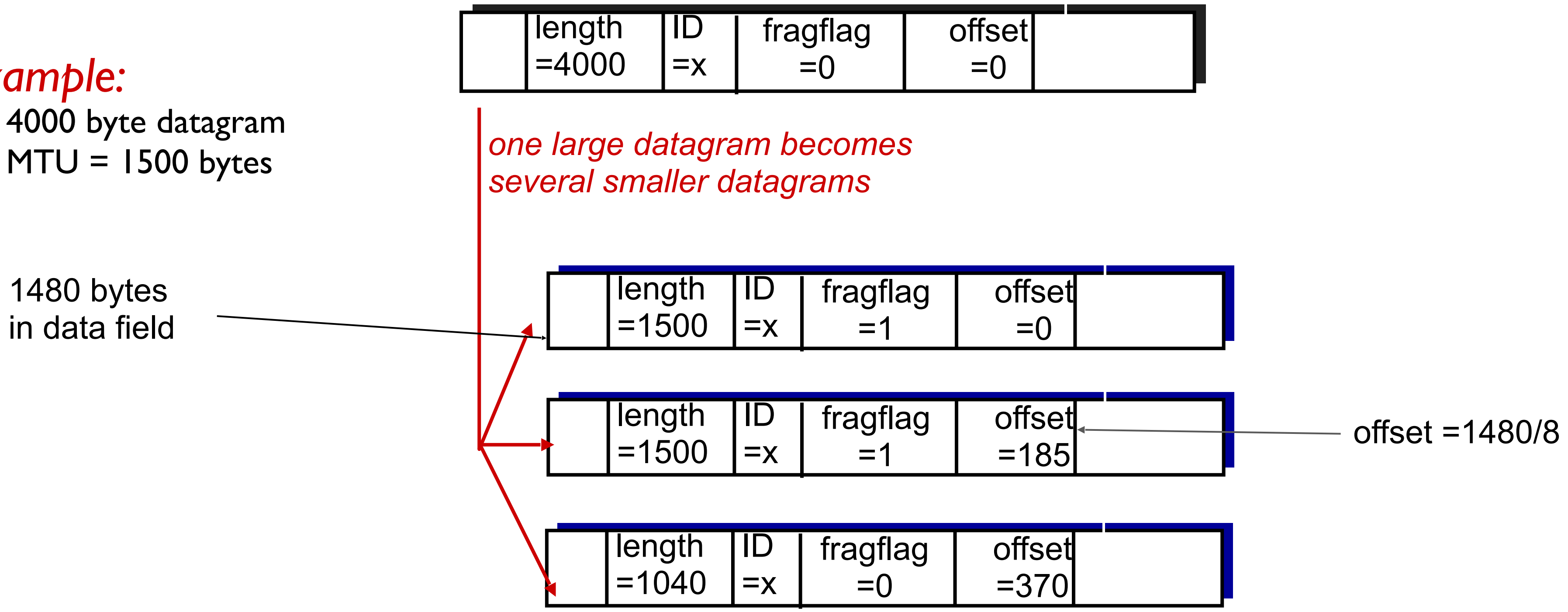
- network links have MTU (max.transfer size) - largest possible link-level frame
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments



IP fragmentation, reassembly

example:

- 4000 byte datagram
- MTU = 1500 bytes



Fragmentation – Example

Original IP Datagram

Sequence	Identifier	Total Length	DF May / Don't	MF Last / More	Fragment Offset
0	345	5140	0	0	0

IP Fragments (Ethernet)

Sequence	Identifier	Total Length	DF May / Don't	MF Last / More	Fragment Offset
0-0	345	1500	0	1	0
0-1	345	1500	0	1	185
0-2	345	1500	0	1	370
0-3	345	700	0	0	555

Fragmentation – Numerical Example

- An IP router with a Maximum Transmission Unit (MTU) of 200 bytes has received an IP packet of size 520 bytes with an IP header of length 20 bytes. The values of the relevant fields in the IP header.

	20 | 176	20 | 176	20 | 148
Fragment Offset	0	22	44
MF	1	1	0
Header length	5	5	5
Total length	196	196	168

Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

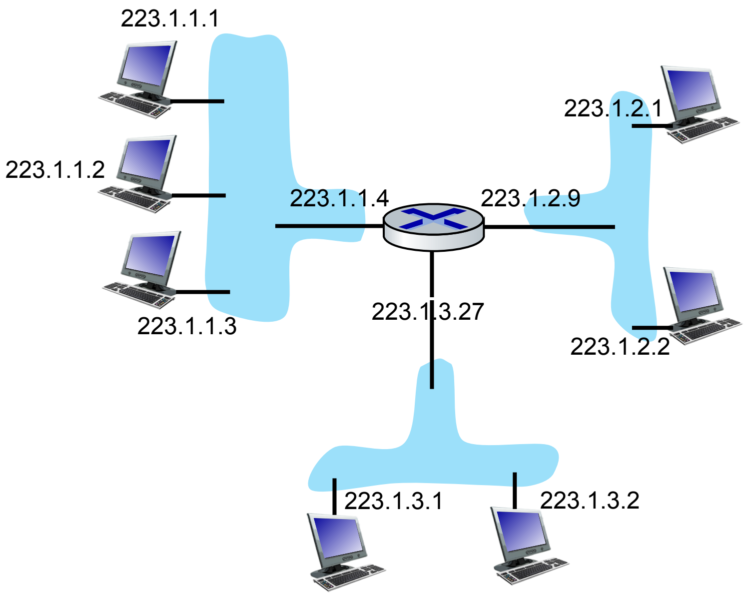
4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

IP addressing: Introduction

- IP address: 32-bit unique ID associated with each host or router *interface*
- interface: connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)



dotted-decimal IP address notation:

$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111

Solution:

We replace each group of 8 bits with its equivalent decimal number and add dots for separation.

a. 129.11.11.239
b. 193.131.27.255

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78

b. 221.34.7.82

Solution:

We replace each decimal number with its binary equivalent.

a. 01101111 00111000 00101101 01001110

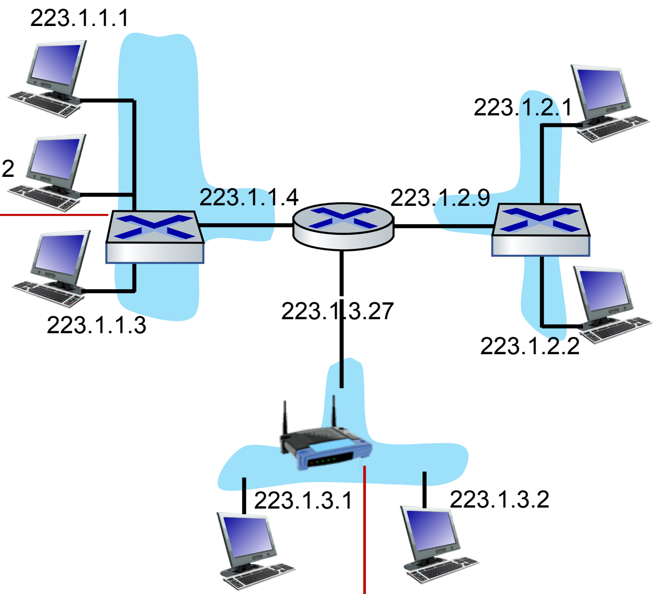
b. 11011101 00100010 00000111 01010010

IP addressing: Introduction

Q: how are interfaces actually connected?

A: we'll learn about that in chapters 6, 7

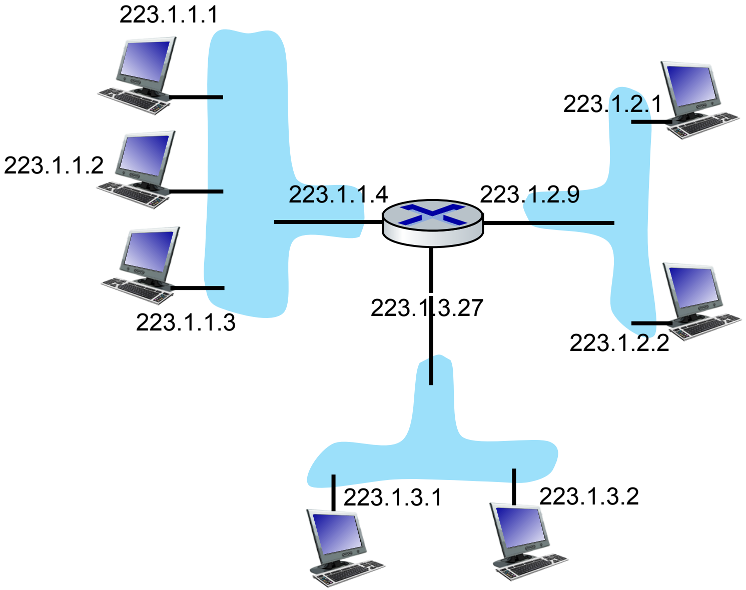
A: wired Ethernet interfaces connected by Ethernet switches



A: wireless WiFi interfaces connected by WiFi base station

For now: don't need to worry about how one interface is connected to another (with no intervening router)

- *What's a subnet ?*
  - device interfaces that can physically reach each other without passing through an intervening router



network consisting of 3 subnets

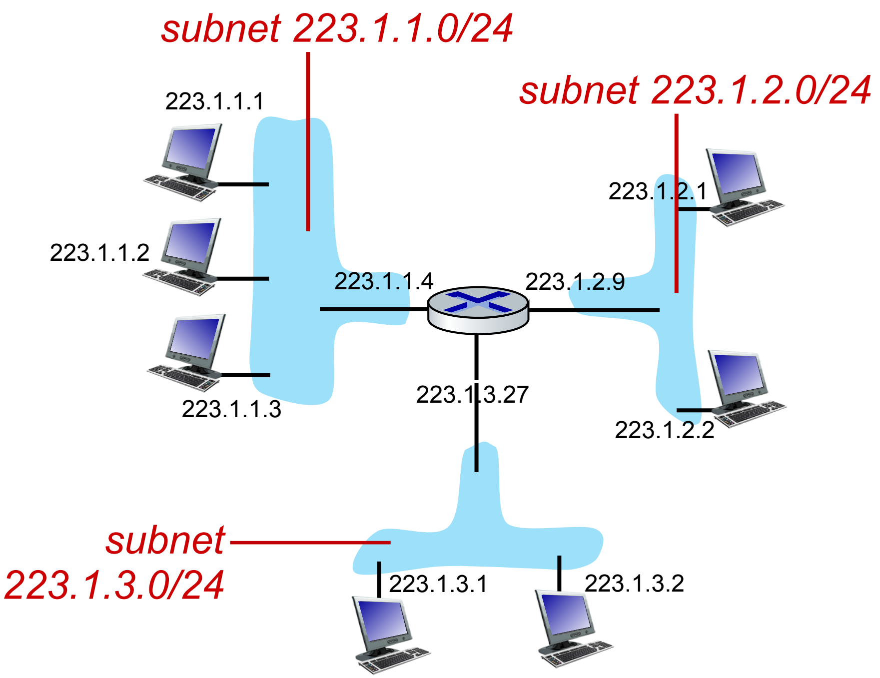
- IP addresses have structure:
  - subnet part: devices in same subnet have common high order bits
  - host part: remaining low order bits

A portion of a network that shares a particular subnet address.

Subnets

Recipe for defining subnets:

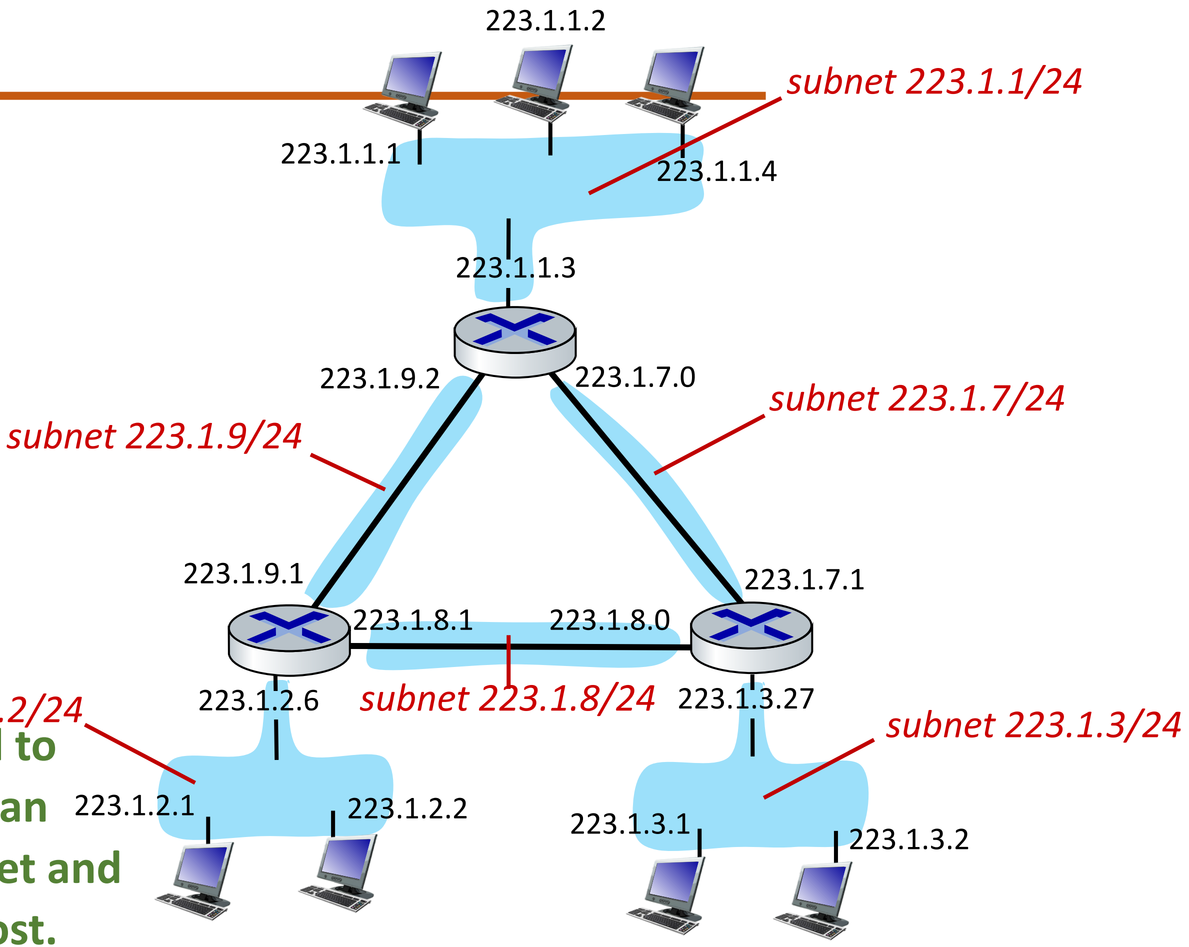
- detach each interface from its host or router, creating "islands" of isolated networks
- each isolated network is called a *subnet*



subnet mask: /24

(high-order 24 bits: subnet part of IP address)

- where are the subnets?
- what are the /24 subnet addresses?



A 32-bit combination used to describe which portion of an address refers to the subnet and which part refers to the host.

CIDR: Classless InterDomain Routing (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

subnet part | host part

11001000 00010111 00010000 00000000

200.23.16.0/23

IP addressing: CIDR

CIDR Block Prefix	# of Host Addresses (inclusive of network and broadcast-id)
/30	4 hosts (valid host would be 2)
/29	8 hosts (valid host would be 6)
/28	16 hosts (valid host would be 14)
/27	32 hosts
/26	64 hosts
/25	128 hosts
/24	256 hosts
/23	512 hosts
/22	1,024 hosts
/21	2,048 hosts
/20	4,096 hosts
/19	8,192 hosts
/18	16,384 hosts
/17	32,768 hosts
/16	65,536 hosts
/15	131,072 hosts
/14	262,144 hosts
/13	524,288 hosts
/12	1,048,576 hosts and so on…….

IP addressing: Why not Classful Addressing?

- Network portion - 8, 16, or 24 bits in length – known as **Class A, B** and **C** networks respectively
- A class **C (/24)** - accommodate only up to $2^8 - 2 =$ **254 hosts** (two of the $2^8 = 256$ addresses are reserved for special use)
- A class **B (/16)** subnet, supports up to **65,634 hosts**
- A class **A (/8)** subnet, up to **16,777,214 hosts**

too small for many organizations

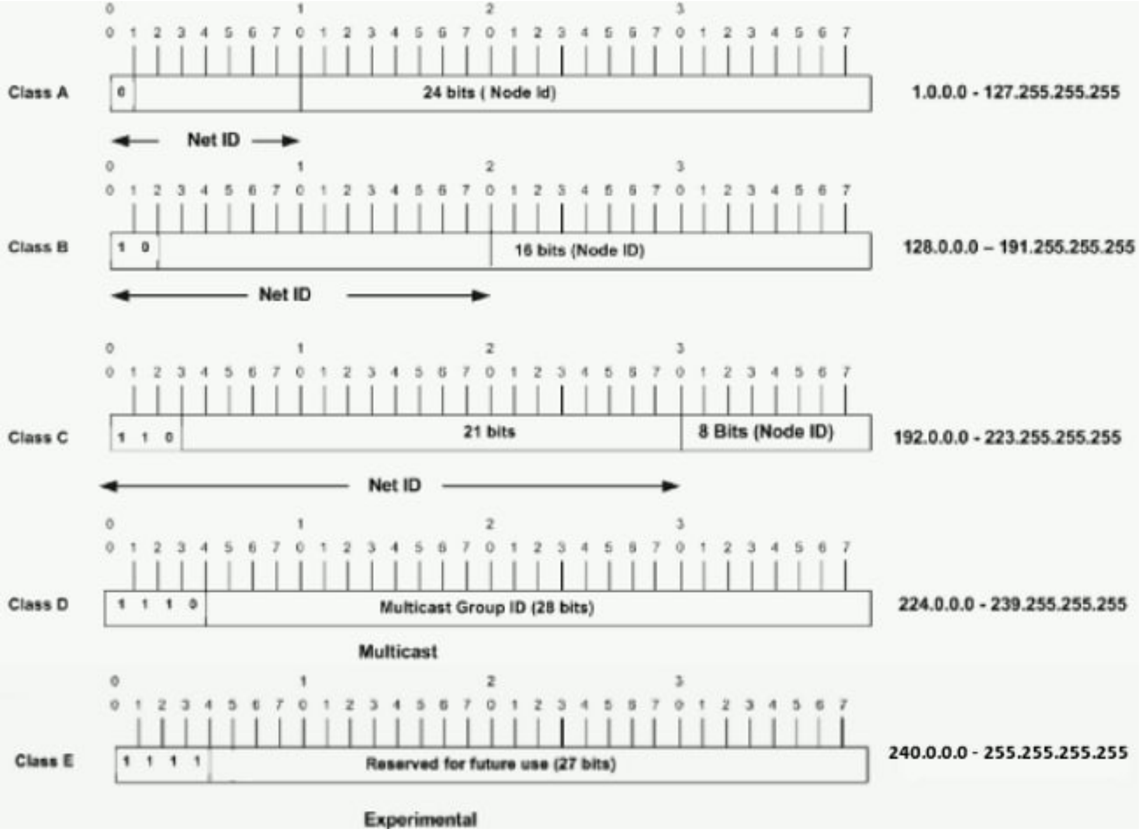
too large, poor utilization

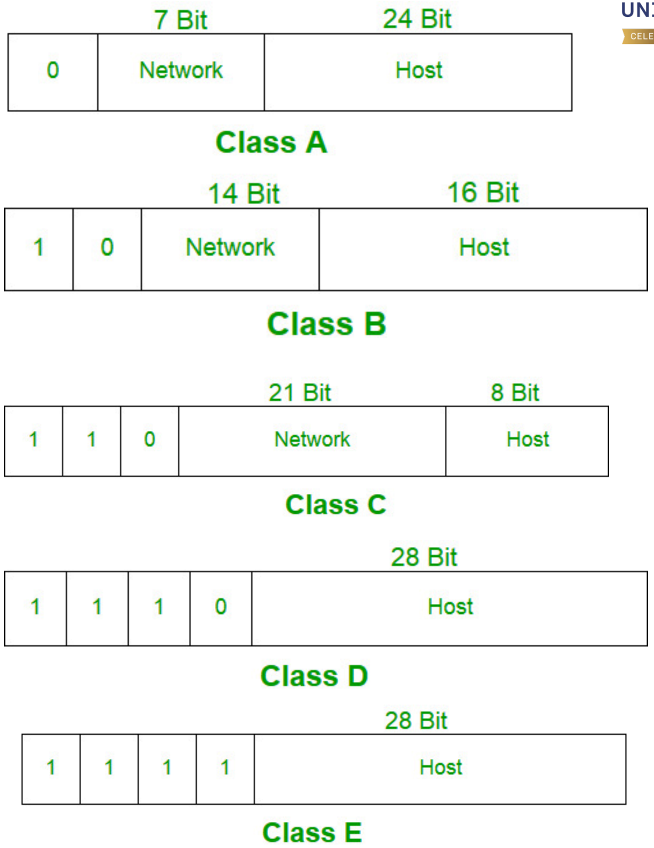
1. 2^n=Number of Network
2. 2^h-2 = Number of Host

Classful addressing, which is almost obsolete, is replaced with classless addressing.

COMPUTER NETWORKS

IP addressing: Classful Addressing





IP addressing: Classful Addressing

Class	HOB	NET ID Bits	Host ID Bits	No of Networks	Host Per Network	Start Address	End Address
Class A	0	8	24	2^7=128	2^{24}=16,777,216	0.0.0.0	127.255.255.255
Class B	10	16	16	2^{14}=16,384	2^{16}=65,536	128.0.0.0	191.255.255.255
Class C	110	24	8	2^{21}=2,097,152	2^8=256	192.0.0.0	223.255.255.255
Class D	1110	-	-	-	-	224.0.0.0	239.255.255.255
Class E	1111	-	-	-	-	240.0.0.0	255.255.255.255

Class D: multicast, Class E: reserved (experimental)

Broadcast Address: 255.255.255.255

Network Masks

Class A: 255.0.0.0
Class B: 255.255.0.0
Class C: 255.255.255.0

IP addressing: Classful Subnet Masks

Class	Binary	Dotted-Decimal	CIDR Notation
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

IP addresses: how to get one?

Q: how does *network* get subnet part of IP address?

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

ISP's block	11001000 00010111 00010000 00000000	200.23.16.0/20

ISP can then allocate out its address space in 8 blocks:

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

Find the class of each address.

a. <u>0</u>0000001 00001011 00001011 11101111
b. <u>110</u>00001 10000011 00011011 11111111
c. <u>14</u>.23.120.8
d. <u>252</u>.5.15.111

Solution

a. The first bit is 0. This is a class A address.
b. The first 2 bits are 1; the third bit is 0. This is a class C address.
c. The first byte is 14; the class is A.
d. The first byte is 252; the class is E.

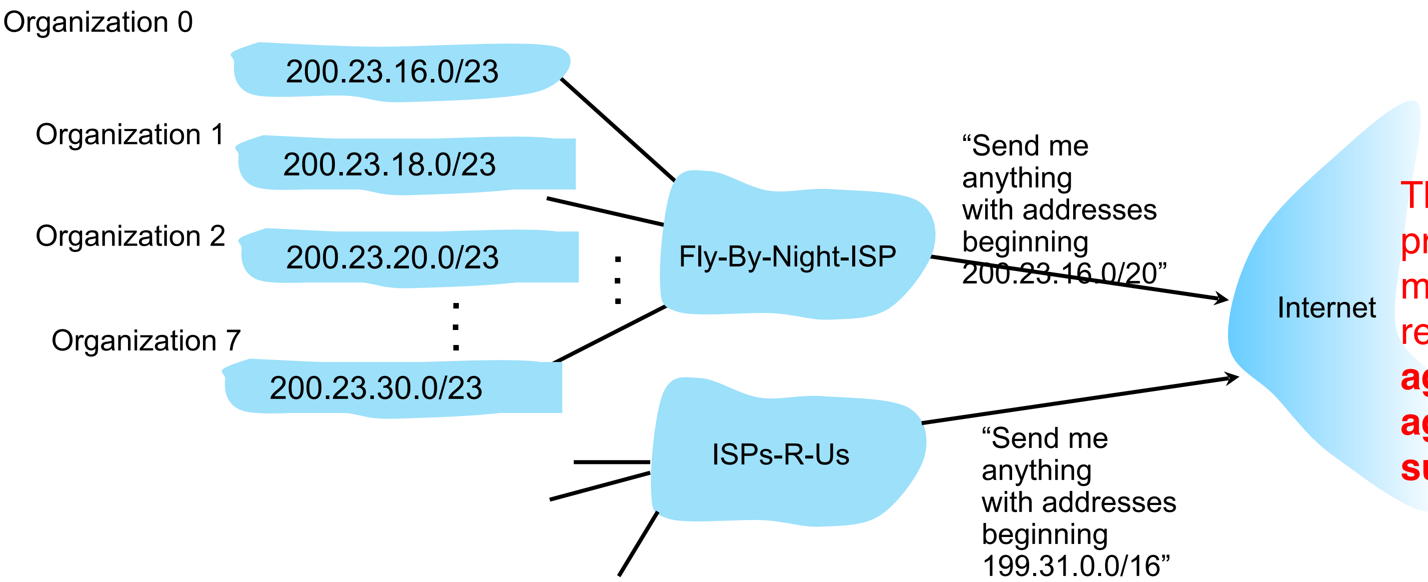
IP addressing: CIDR – Prefix Lengths

/n	Mask	/n	Mask	/n	Mask	/n	Mask
/1	128.0.0.0	/9	255.128.0.0	/17	255.255.128.0	/25	255.255.255.128
/2	192.0.0.0	/10	255.192.0.0	/18	255.255.192.0	/26	255.255.255.192
/3	224.0.0.0	/11	255.224.0.0	/19	255.255.224.0	/27	255.255.255.224
/4	240.0.0.0	/12	255.240.0.0	/20	255.255.240.0	/28	255.255.255.240
/5	248.0.0.0	/13	255.248.0.0	/21	255.255.248.0	/29	255.255.255.248
/6	252.0.0.0	/14	255.252.0.0	/22	255.255.252.0	/30	255.255.255.252
/7	254.0.0.0	/15	255.254.0.0	/23	255.255.254.0	/31	255.255.255.254
/8	255.0.0.0	/16	255.255.0.0	/24	255.255.255.0	/32	255.255.255.255

The addresses in color are the default masks for classes A, B, and C.
Thus, classful addressing is a special case of classless addressing.

Hierarchical addressing: route aggregation

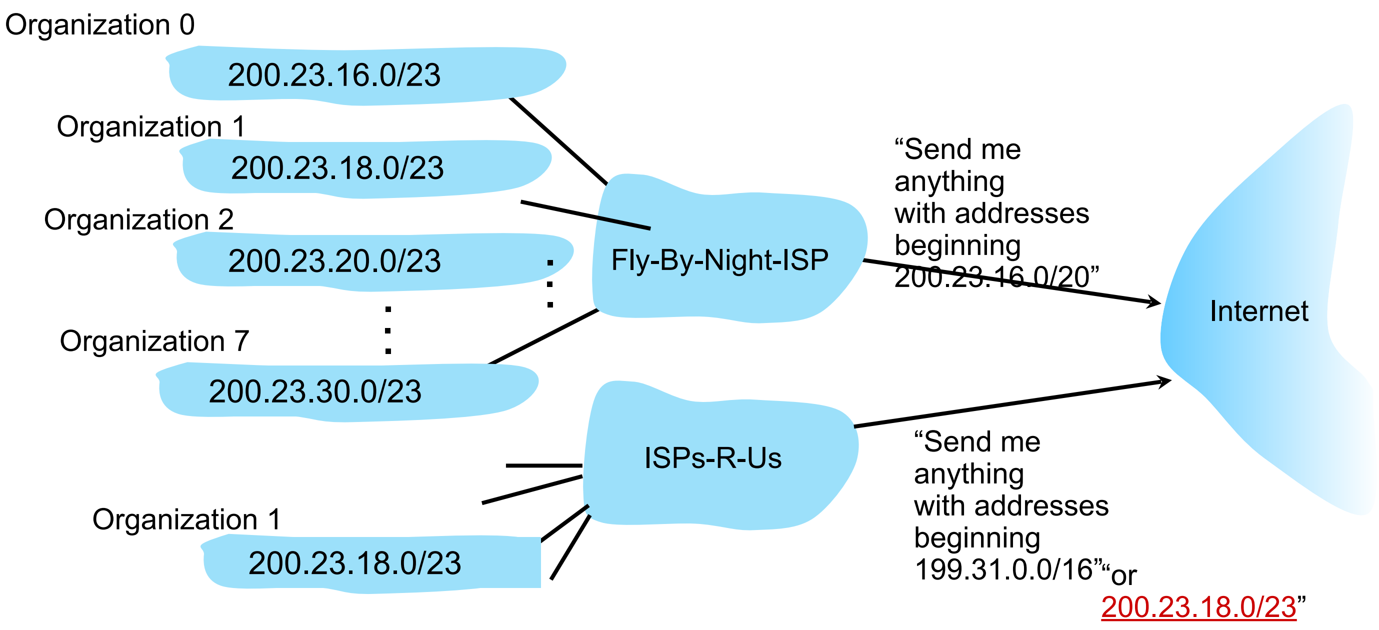
hierarchical addressing allows efficient advertisement of routing information:



The ability to use a single prefix to advertise multiple networks is often referred to as **address aggregation** (also **route aggregation** or **route summarization**).

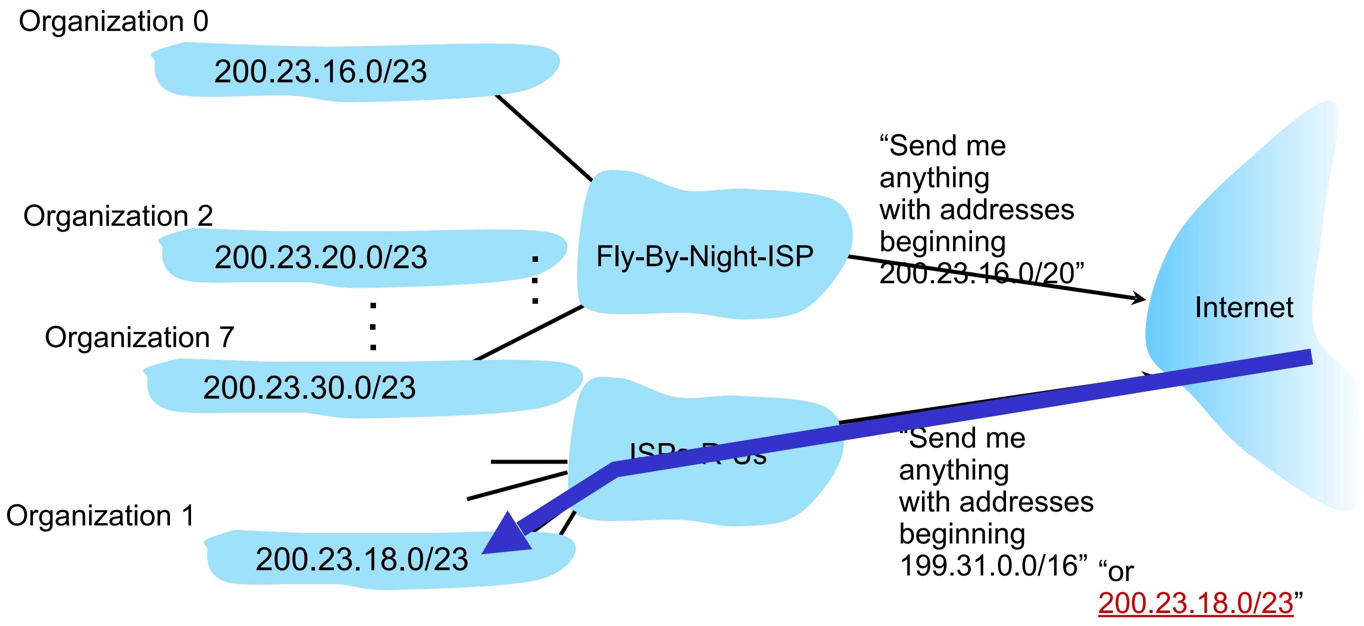
Hierarchical addressing: more specific routes

- Organization 1 moves from Fly-By-Night-ISP to ISPs-R-Us
- ISPs-R-Us now advertises a more specific route to Organization 1



Hierarchical addressing: more specific routes

- Organization 1 moves from Fly-By-Night-ISP to ISPs-R-Us
- ISPs-R-Us now advertises a more specific route to Organization 1



IP addressing: last words...

Q: how does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers http://www.icann.org/

- allocates IP addresses, through 5 regional registries (RRs) (who may then allocate to local registries)
- manages DNS root zone, including delegation of individual TLD (.com, .edu, …) management

Q: are there enough 32-bit IP addresses?

- ICANN allocated last chunk of IPv4 addresses to RRs in 2011
- NAT (next) helps IPv4 address space exhaustion
- IPv6 has 128-bit address space

"Who the hell knew how much address space we needed?" Vint Cerf (reflecting on decision to make IPv4 address 32 bits long)

Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

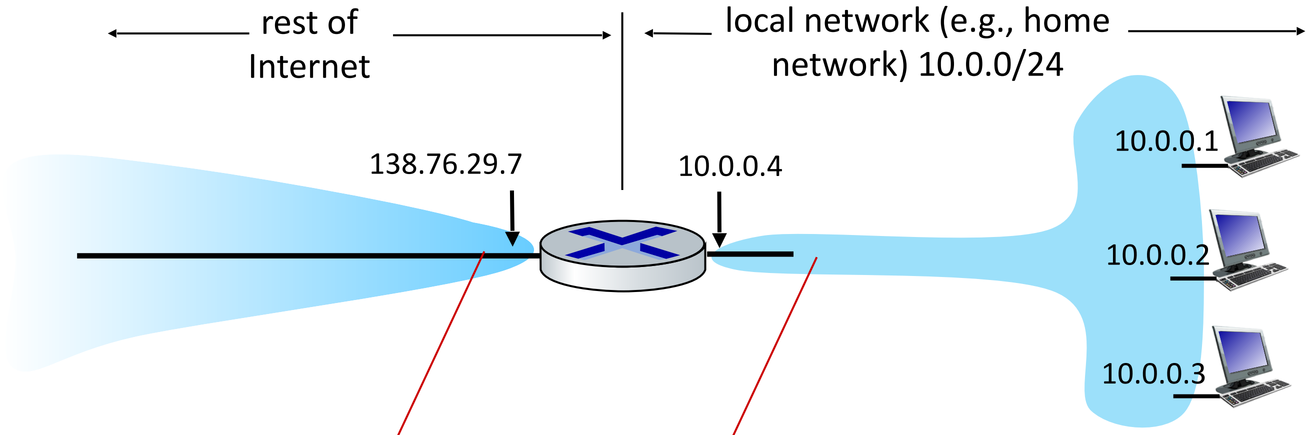
- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

NAT: Network Address Translation

- all devices in local network have 32-bit addresses in a "private" IP address space (10/8, 172.16/12, 192.168/16 prefixes) that can only be used in local network
- advantages:
  - just one IP address needed from provider ISP for *all* devices
  - can change addresses of host in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - security: devices inside local net not directly addressable, visible by outside world

NAT: Network Address Translation

NAT: all devices in local network share just one IPv4 address as far as outside world is concerned



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

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual) implementation: NAT router must (transparently):

- 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 address
- 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 destination fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

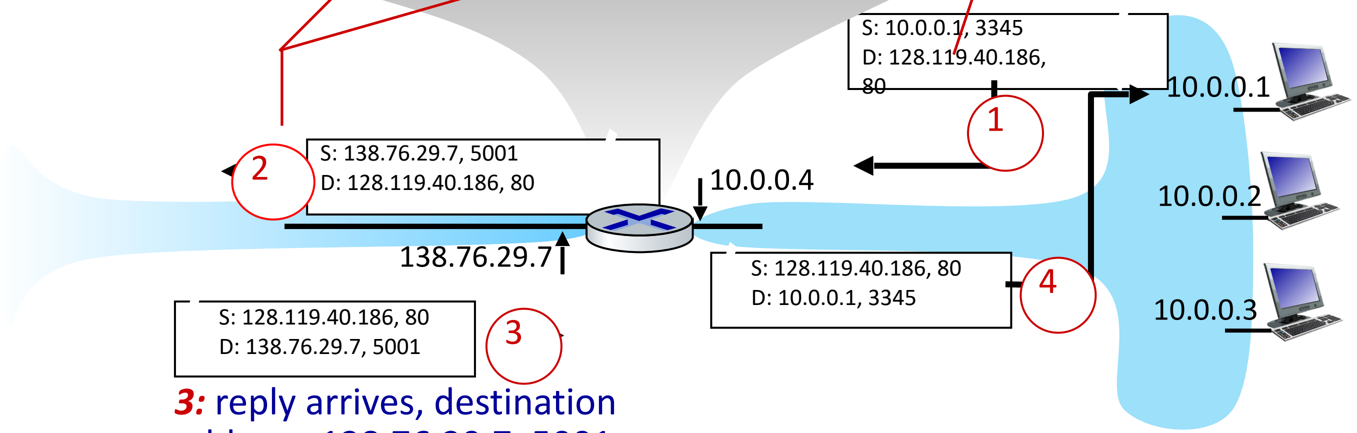
NAT: Network Address Translation

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

NAT translation table

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

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

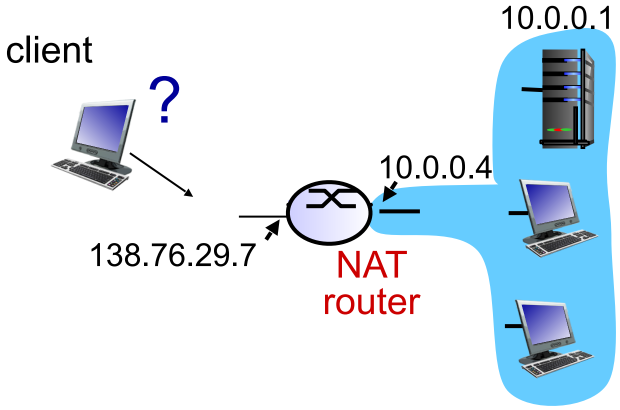


3: reply arrives, destination address: 138.76.29.7, 5001

NAT: Network Address Translation

- NAT has been controversial:
  - routers "should" only process up to layer 3
  - address "shortage" should be solved by IPv6
  - violates end-to-end argument (port # manipulation by network-layer device)
  - NAT traversal: what if client wants to connect to server behind NAT?
- but NAT is here to stay:
  - extensively used in home and institutional nets, 4G/5G cellular nets

NAT: Network Address Translation



Solution: statically configure NAT to forward incoming connection requests at given port to server

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

IPv4 Addressing & Subnetting – Numerical Example

Find the range of addresses in the following blocks.

- 123.56.77.32/29
- 180.34.64.64/30
- 200.17.21.128/27

IPv4 Addressing & Subnetting – Numerical Example

You are given the network address **175.200.0.0**; you are required to have 4 subnets.

What is the minimum number of Host Bits can you take in to the Network Bits for this purpose?

Write down the addresses of 4 subnets.

Write the subnet mask for the network.

IPv4 Addressing & Subnetting – Numerical Example

In a block of addresses, we know the IP address of one host is **25.34.12.56/16**.

What are the first address (network address) and the last address (limited broadcast address) in this block?

Please explain in full with detailed steps showing the calculation.

IPv4 Addressing & Subnetting – Numerical Example

An organization is granted the block **16.0.0.0/8**. The administrator wants to create 500 fixed- length subnets.

i) find the subnet mask

ii) find the number of addresses in each subnet.

iii) find the first and last address in first subnet and in the last subnet (500th subnet).

IPv4 Addressing & Subnetting – Numerical Example

An organization is granted a block of addresses with the beginning address **14.24.74.0/24**. The organization needs to have 3 subblocks of addresses to use in its three subnets: one subblock of **10 addresses**, one subblock of **60 addresses**, and one subblock of **120 addresses**. Design the subblocks.

IPv4 Addressing & Subnetting – Numerical Example

Find the network address and the directed broadcast address of subnetted Class B IPv4 address **172.25.171.182** with a subnet mask of **255.255.224.0**.

THANK YOU

TEAM NETWORKS
Department of Computer Science and Engineering