Chapter 4: Network Layer

- CSI 2470, Fall 2025
- Dr. Jie Hu

* Modified from the class notes by Kurose/Ross and by my Ph.D. advisor Dr. Do Young Eun

1

Chapter 4: Network Layer

Chapter goals:

- understand principles behind network layer services:
  - network layer service models
  - forwarding versus routing
  - how a router works
  - routing (path selection)
  - dealing with scale
  - advanced topics: IPv6, mobility
- instantiation, implementation in the Internet
  - IP protocol
  - NAT, middleboxes

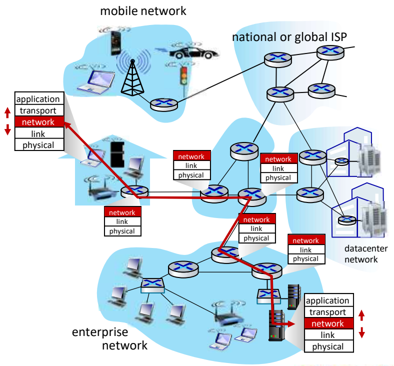
2

Contents

3

Network layer

- 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



4

Two Key Network-Layer Functions

- ***forwarding:*** move packets from router's input to appropriate router output
- ***routing:*** determine route taken by packets from source to dest.
  - *routing algorithms*

analogy:

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

forwarding

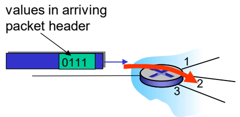
routing

5

Network layer: data plane, control plane

Data plane:

- *local*, per-router function
- determines how datagram arriving on router input port is forwarded to router output port



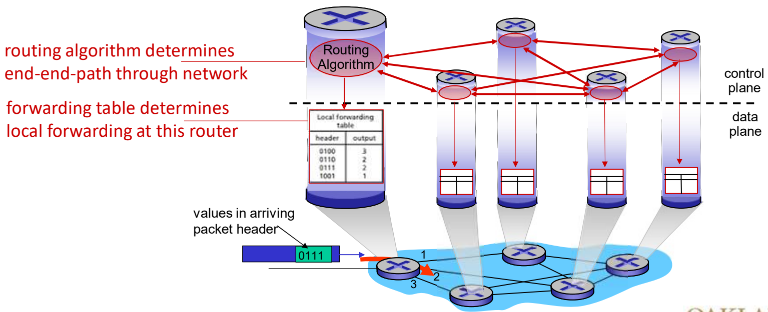
Control plane

- *network-wide* logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- two control-plane approaches:
  - *traditional routing algorithms:* implemented in routers
  - *software-defined networking (SDN)*: implemented in (remote) servers

6

Interplay between routing and forwarding

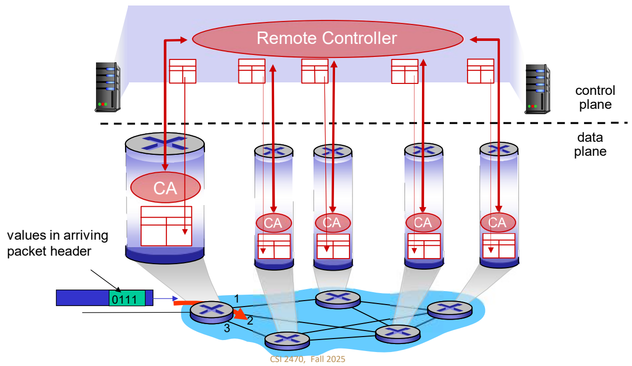
Individual routing algorithm components *in each and every router* interact in the control plane



7

Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



8

Network service model

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

Reflections on best-effort service:

- simplicity of mechanism has allowed Internet to be widely deployed adopted
- sufficient provisioning of bandwidth allows performance of real-time applications (e.g., interactive voice, video) to be "good enough" for "most of the time"
- replicated, application-layer distributed services (datacenters, content distribution networks) connecting close to clients' networks, allow services to be provided from multiple locations
- congestion control of "elastic" services helps

It's hard to argue with success of best-effort service model

Network layer: connection and connection-less service

- datagram network provides network-layer connectionless service
- VC network provides network-layer connection service
- analogous to the transport-layer services, but:
  - service: host-to-host
  - no choice: network provides one or the other
  - implementation: in the network core + end hosts

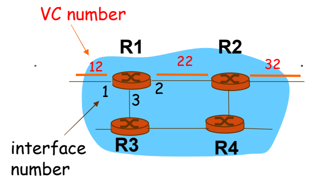
Virtual Circuits

"source-to-dest path behaves much like telephone circuit"
- performance-wise
- network actions along source-to-dest path

- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- *every* router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)

Virtual Circuits: Forwarding Table

Forwarding table in Northwest (R1) router:

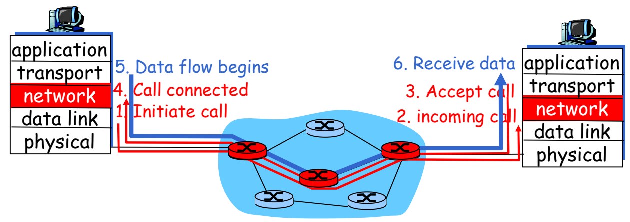


Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

Routers maintain connection state information!

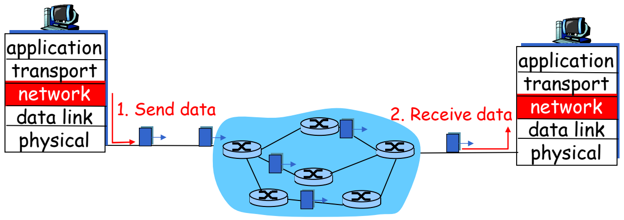
Virtual circuits: signaling protocols

- used to setup, maintain teardown VC
- used in ATM, frame-relay, X.25
- not used in today's Internet

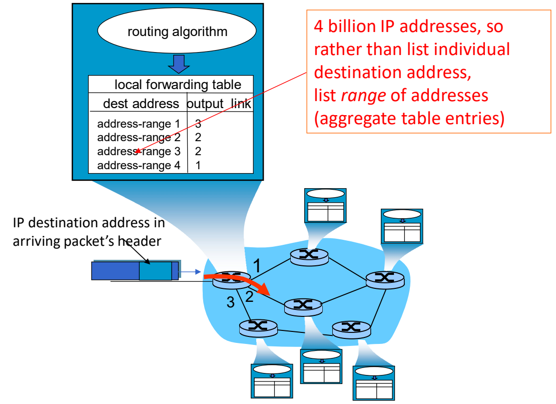


Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address
  - packets between same source-dest pair may take different paths



Datagram Forwarding Table



Datagram network: Forwarding 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

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

Datagram network: Longest prefix matching

Longest prefix matching

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

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

Examples

DA: 11001000 00010111 00010110 10100001 Which interface?

DA: 11001000 00010111 00011000 10101010 Which interface?

Datagram or VC network: why?

Internet (Datagram)

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

ATM (VC)

- **evolved from telephony**
- **human conversation:**
  - strict timing, reliability requirements
  - need for guaranteed service
- **"dumb" end systems**
  - telephones
  - *complexity inside network*

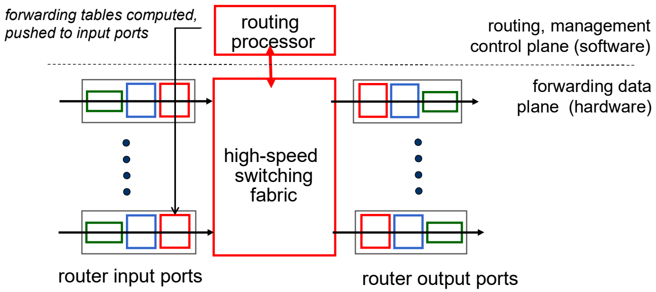
Contents

- Network layer -- Virtual Circuit and Datagram networks
- **Router architecture and functions: What's inside a router?**
- Internet Protocol (IP)
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- Routing in the Internet: RIP, OSPF, and BGP

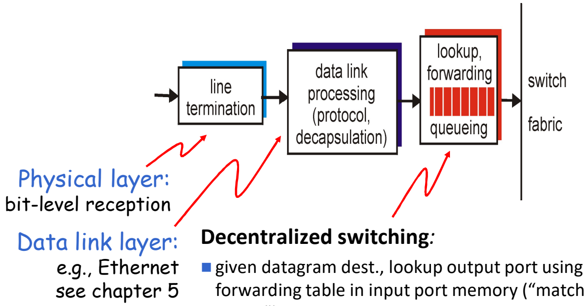
Router Architecture Overview

two key router functions:

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



Input Port Functions

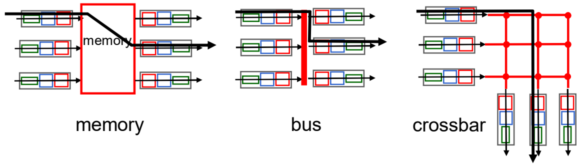


Decentralized switching:

- 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

Switching fabrics

- transfer packet from input buffer to appropriate output buffer
- 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 types of switching fabrics

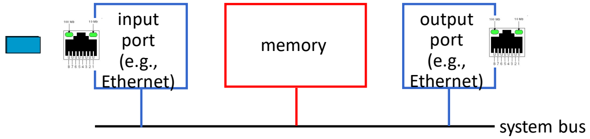


25

Switching via memory

first generation routers:

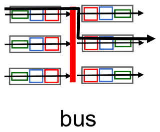
- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth (2 bus crossings per datagram)



26

Switching via a bus

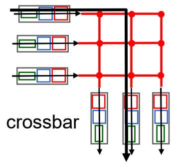
- datagram from input port memory
  to output port memory via a shared bus
- *bus contention:* switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers

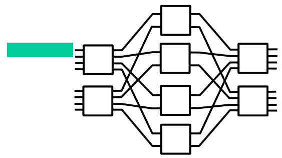


27

Switching via interconnection network

- overcome bus bandwidth limitations
- banyan networks, crossbar, other interconnection nets initially developed to connect processors in multiprocessor
- advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.



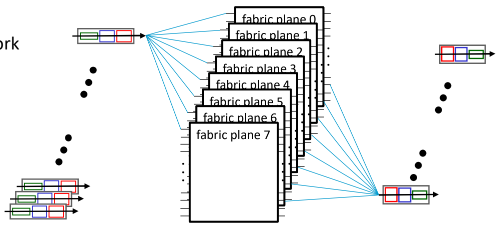


8x8 multistage switch
built from smaller-sized switches

28

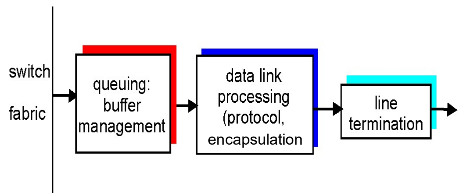
Switching via interconnection network

- 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



29

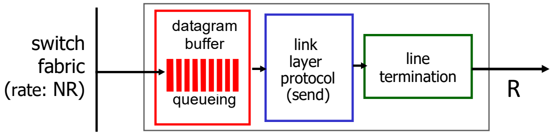
Output Ports



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

30

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

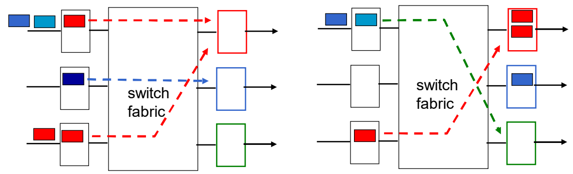
- *Scheduling discipline* chooses among queued datagrams for transmission

Priority scheduling – who gets best performance, network neutrality

31

Output port queueing

Switching fabric speed: the rate at which the switching fabric can move packets from input ports to output ports



at *t*, packets move 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!*

32

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 Gps link: 2.5 Gbits buffer
- Recent recommendation: with *N* flows, buffering equal to

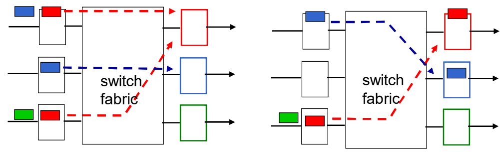
$$\frac{RTT \times C}{\sqrt{N}}$$

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

33

Input port queuing

- **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: <u>only one red datagram can be transferred.</u> *lower red packet is blocked*

one packet time later: green packet experiences HOL blocking

34

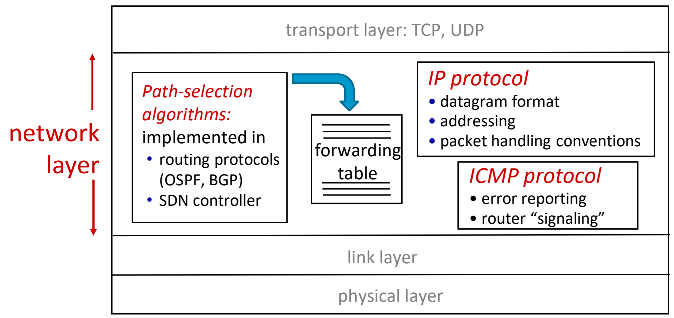
Contents

- Network layer -- Virtual Circuit and Datagram networks
- Router architecture and functions: What's inside a router?
- **Internet Protocol (IP)**
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- Routing in the Internet: RIP, OSPF, and BGP

35

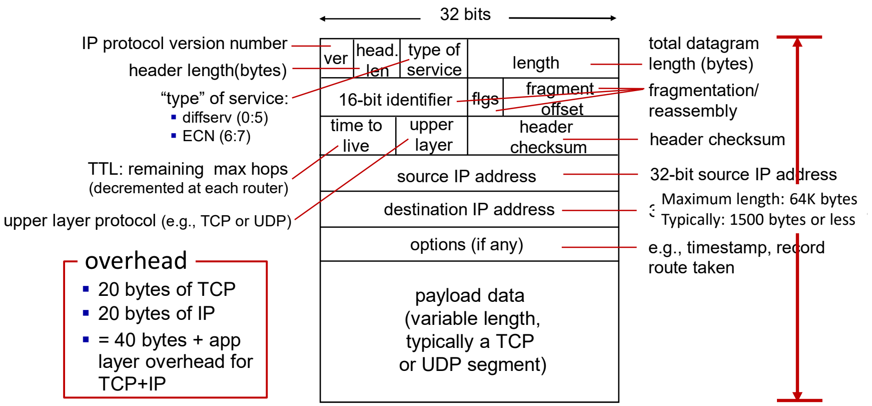
The Internet Network layer

Host, router network layer functions:

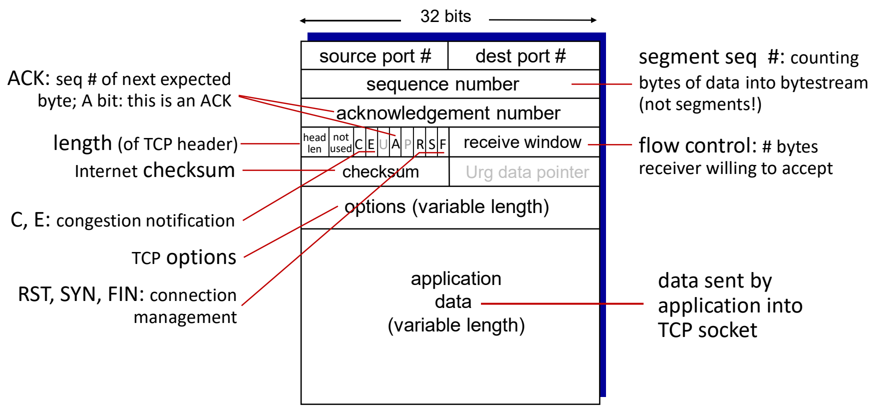


36

IP datagram format



TCP segment structure: Recall



IP Packet Header

0 Version	4 IHL	8 Type of Service	16 Total Length	19	24	31
Identification			Flags	Fragment Offset		

Version: current IP version is 4.

Internet header length (IHL): length of the header in 32-bit words.

Type of service (TOS): traditionally priority of packet at each router. Recent Differentiated Services redefines TOS field to include other services besides best effort.

Total length: number of bytes of the IP packet including header and data, maximum length is 65535 bytes.

Identification, Flags, and Fragment Offset: used for fragmentation and reassembly (More on this shortly).

IP Packet Header (Cont'd)

0 Version	4 IHL	8 Type of Service	16 Total Length	19	24	31
Identification			Flags	Fragment Offset		
Time to Live		Protocol	Header Checksum			

Time to live (TTL): number of hops packet is allowed to traverse in the network. Each router along the path to the destination decrements this value by one.

If the value reaches zero before the packet reaches the destination, the router discards the packet and sends an error message back to the source.

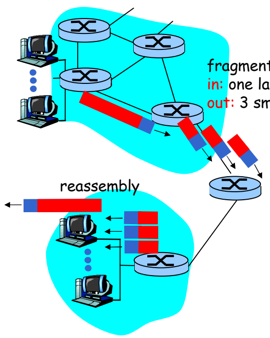
Protocol: specifies upper-layer protocol that is to receive IP data at the destination. Examples include TCP (protocol = 6), UDP (protocol = 17), and ICMP (protocol = 1).

Header checksum: verifies the integrity of the IP header.

IP Fragmentation and Reassembly

- Link layers have MTU (max. transmission Unit) - 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

Ethernet frame can carry up to 1500 bytes of data → hard limit!



IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes
- 20 bytes IP header

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

One large datagram becomes several smaller datagrams

offset value are specified in units of 8-bytes chunks

length	ID	fragflag	offset
=	=x	=	=

length	ID	fragflag	offset
=	=x	=	=

length	ID	fragflag	offset
=	=x	=	=

Flag = 1: there's more, Flag = 0 : this is the end

IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes
- 20 bytes IP header

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

One large datagram becomes several smaller datagrams

1480 bytes in data field

offset = 1480/8

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

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

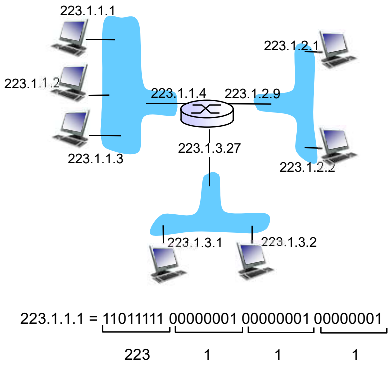
length	ID	fragflag	offset
=1040	=x	=0	=370

Flag = 1: there's more, Flag = 0 : this is the end

Contents

IP addressing: introduction

- *IP address:* 32-bit identifier for host, 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)
- *IP addresses associated with each interface*

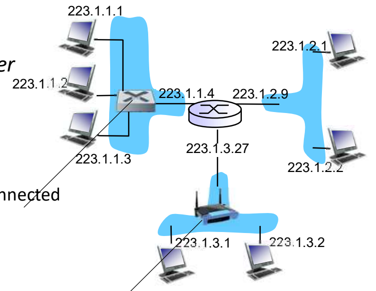


45

IP Addressing: introduction

Q: how are interfaces actually connected?

A: we'll learn about that later on layer 2 in chapters 6, 7



A: wired Ethernet interfaces connected by Ethernet switches

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

A: wireless WiFi interfaces connected by WiFi base station

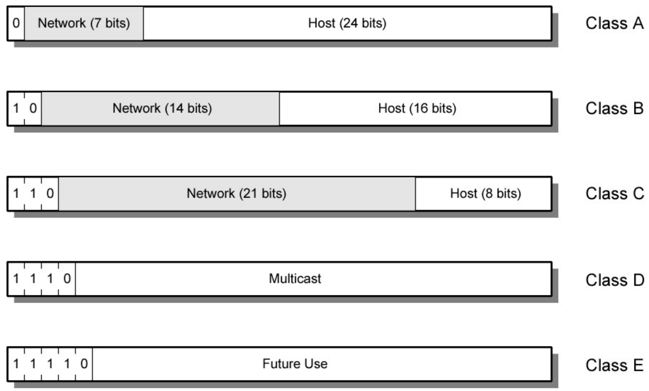
46

IPv4 Addressing

- IP requires each host and router interface to have its own IP address
  - Each host and router is capable of sending and receiving IP datagrams.
- Each IP address is 32 bits long (4 bytes), and there are a total of 2^{32} possible IP addresses.
- These addresses are written in dotted-decimal notation.
  - E.g., 193.32.216.9, each section is the decimal equivalent of eight bits.
  - → 11000001 00100000 11011000 00001001
- Each interface on every host and router in the global internet must have an IP address that is <u>globally unique</u>.

47

IPv4 Address Formats (Classful): Old History

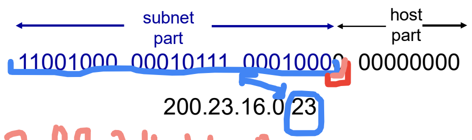


48

IP addressing: CIDR

CIDR: Classless InterDomain Routing (pronounced "cider")

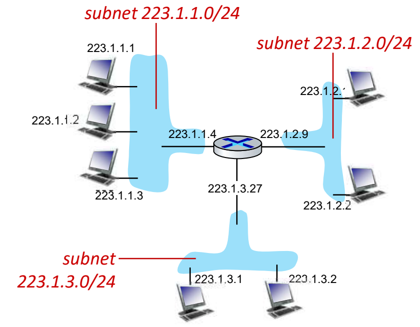
- subnet portion of address of **arbitrary length**
- address format: a.b.c.d/x, where x is # bits in subnet portion of address



200.24.16.0/24
200.23.170/24

What is a Subnet?

- 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

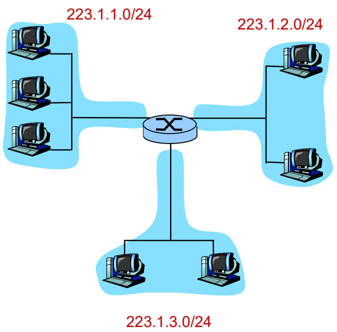


network consisting of 3 subnets

Subnets

Recipe

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



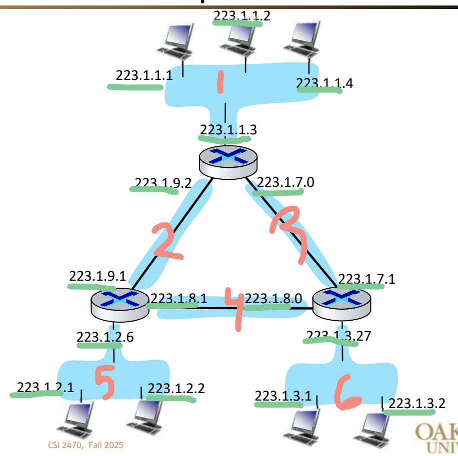
subnet mask: /24
(high-order 24 bits: subnet part of IP address)

Subnets: Example

How many?

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

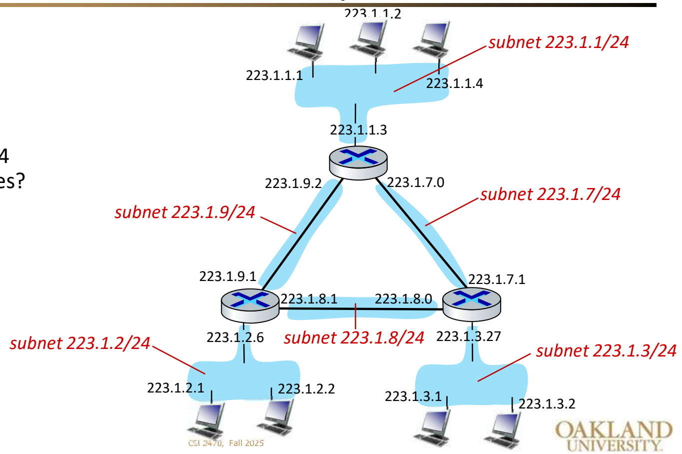
all



Subnets: Example

How many?

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



IP addresses: how to get one?

Q: How does *host* get IP address?

- hard-coded by system admin. in a file
  - Wintel: 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

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")

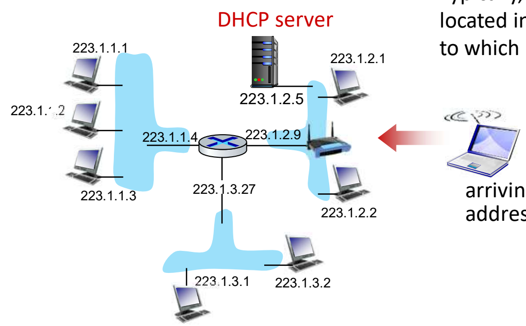
Support for mobile users who want to join network (more later on)

DHCP overview:

- 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

or NAK

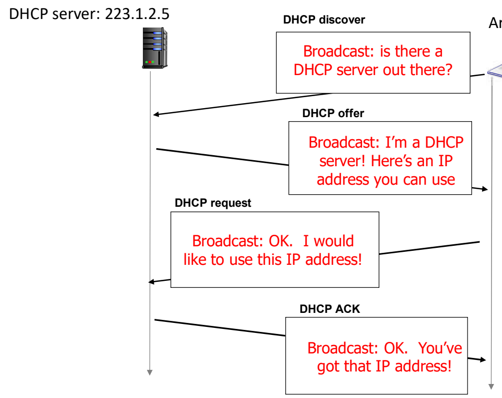
DHCP client-server scenario



Typically, DHCP server will be co-located in router, serving all subnets to which router is attached

arriving DHCP client needs address in this network

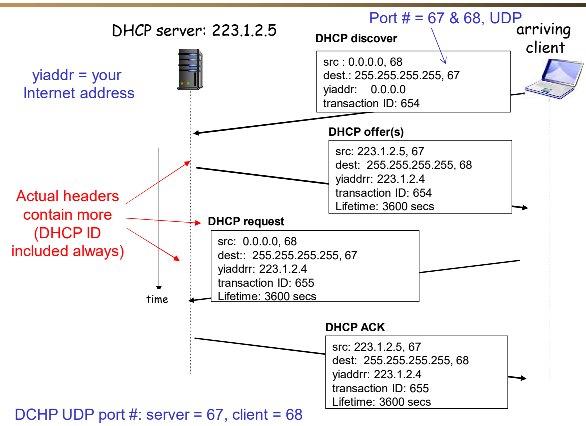
DHCP client-server scenario



The two steps above can be skipped "if a client remembers and wishes to reuse a previously allocated network address" [RFC 2131]

57

DHCP client-server scenario



DCHP UDP port #: server = 67, client = 68

58

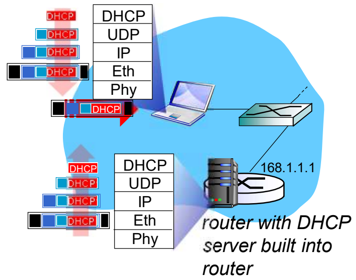
DHCP: more than IP addresses

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)

59

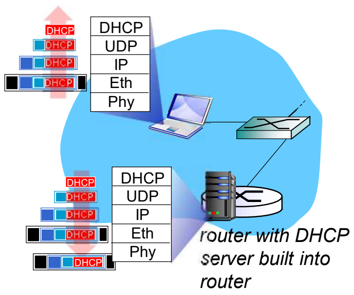
DHCP: example



- connecting laptop needs its IP address, addr of first-hop router, addr of DNS server: use DHCP
- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

60

DHCP: example



- DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation of DHCP server, frame forwarded to client, demuxing up to DHCP at client
- client now knows its IP address, name and IP address of DSN server, IP address of its first-hop router

Jie Hu CSI 2470, Fall 2025 OAKLAND UNIVERSITY

61

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

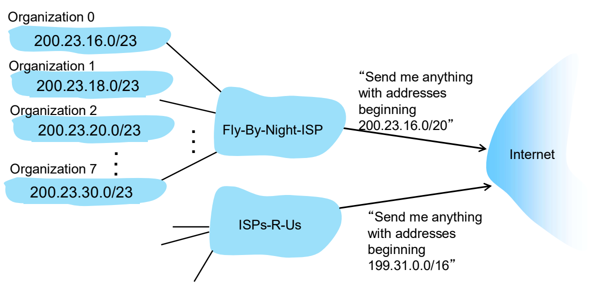
ISP's block	**11001000 00010111 00010000 00000000**	**200.23.16.0/20**
Organization 0	11001000 00010111 00010000 00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010 00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100 00000000	200.23.20.0/23
...	....	....
Organization 7	11001000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

Jie Hu CSI 2470, Fall 2025 OAKLAND UNIVERSITY

62

Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:

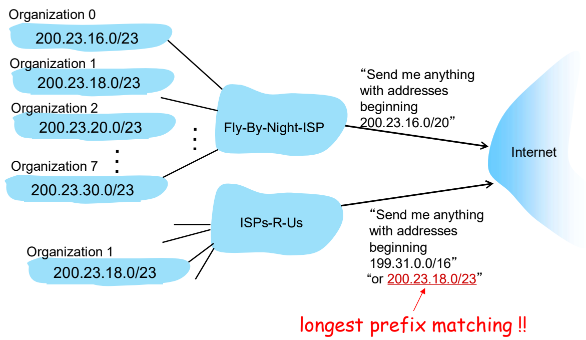


Jie Hu CSI 2470, Fall 2025 OAKLAND UNIVERSITY

63

Hierarchical addressing: more specific routes

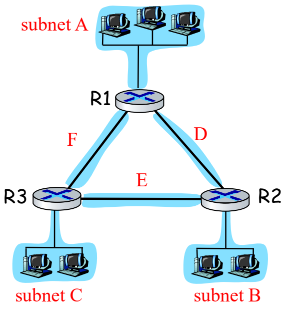
- Organization 1 moves from Fly-By-Night-ISP to ISPs-R-Us
- ISPs-R-Us now advertises a more specific route to Organization 1



Jie Hu CSI 2470, Fall 2025 OAKLAND UNIVERSITY

64

Class Exercise: IP addressing and LPM



- All addresses from 214.97.254/23
- Subnet A: >= 250 addresses
- B and C: at least 120 each
- D, E, F: at least two addresses

Question 1: Assign network addresses to each of six subnets, in the form of a.b.c.d/x or a.b.c.d/x – e.f.g.h/y (all ranges of address in a.b.c.d/x “**except**” those in e.f.g.h/y)

Question 2: Provide forwarding table (using Longest Prefix Matching) for router R1, R2, R3

Class Exercise: IP addressing and LPM

- Address assignments
  - Subnet A:
  - Subnet B:
  - Subnet C:
  - Subnet D:
  - Subnet E:
  - Subnet F:
- Forwarding Table in R1

Longest Prefix Match	Outgoing Interface
	A
	D
	F

Class Exercise: IP addressing and LPM

- Forwarding Table in R2

Longest Prefix Match	Outgoing Interface
	D
	B
	E

- Forwarding Table in R3

Longest Prefix Match	Outgoing Interface
	F
	E
	C

IP addressing: the last word...

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)