# Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6
- NAT

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

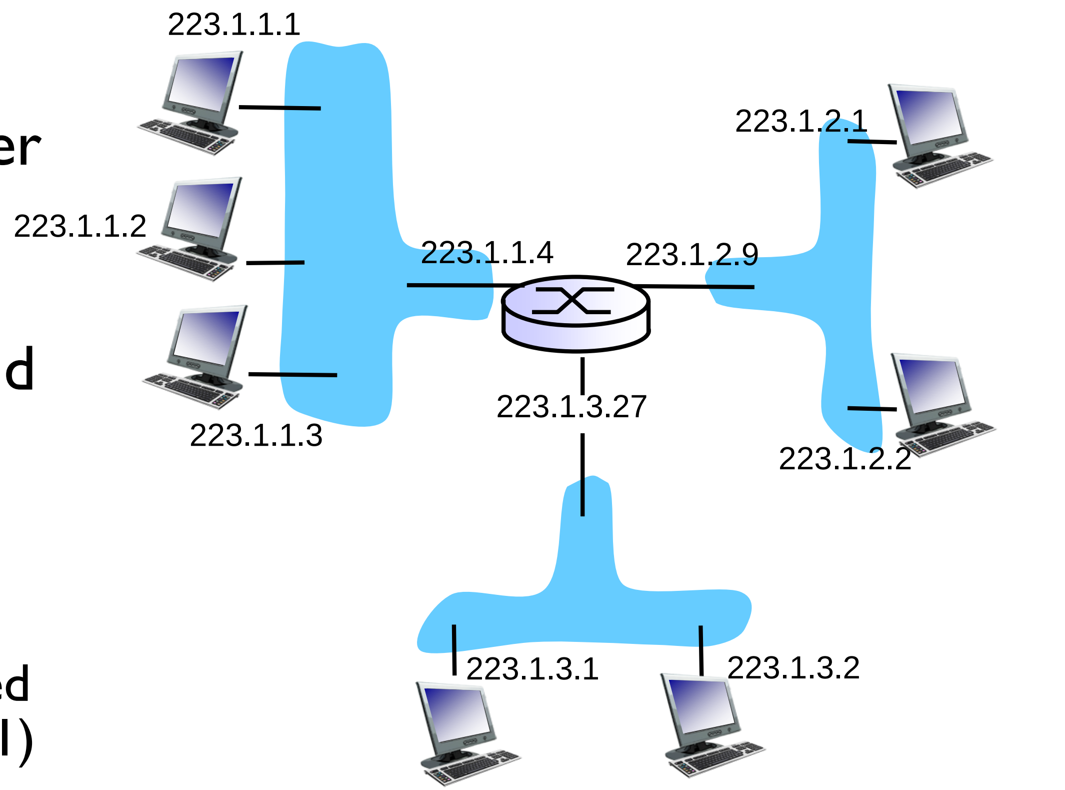
4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

# 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*

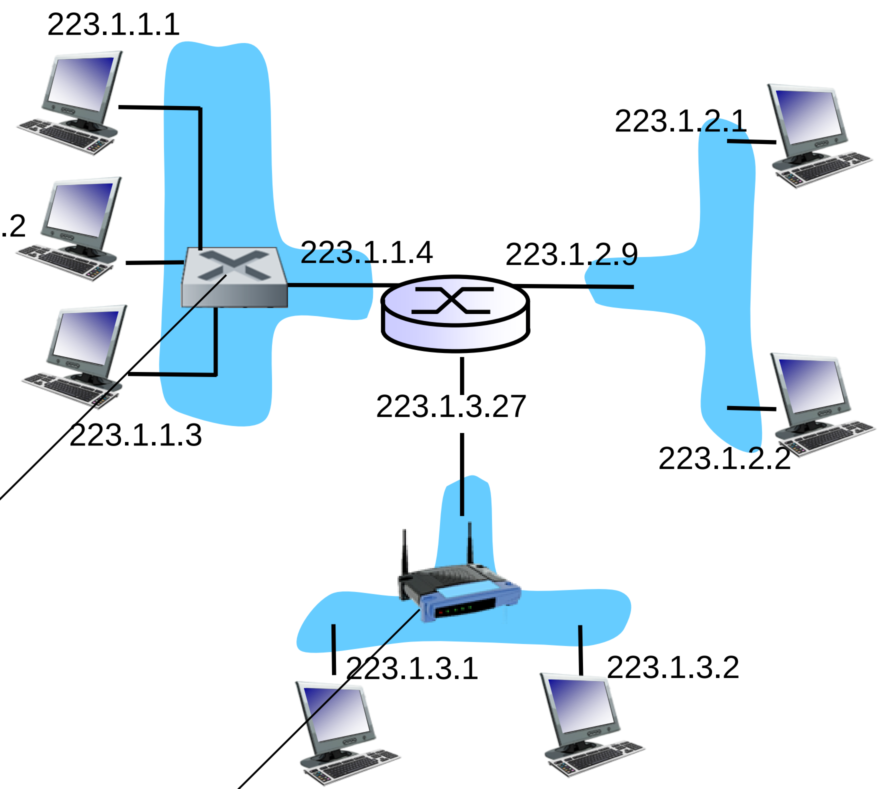



$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

# IP addressing: introduction

*Q: how are interfaces actually connected?*

*A: we'll learn about that in chapter 5, 6.*



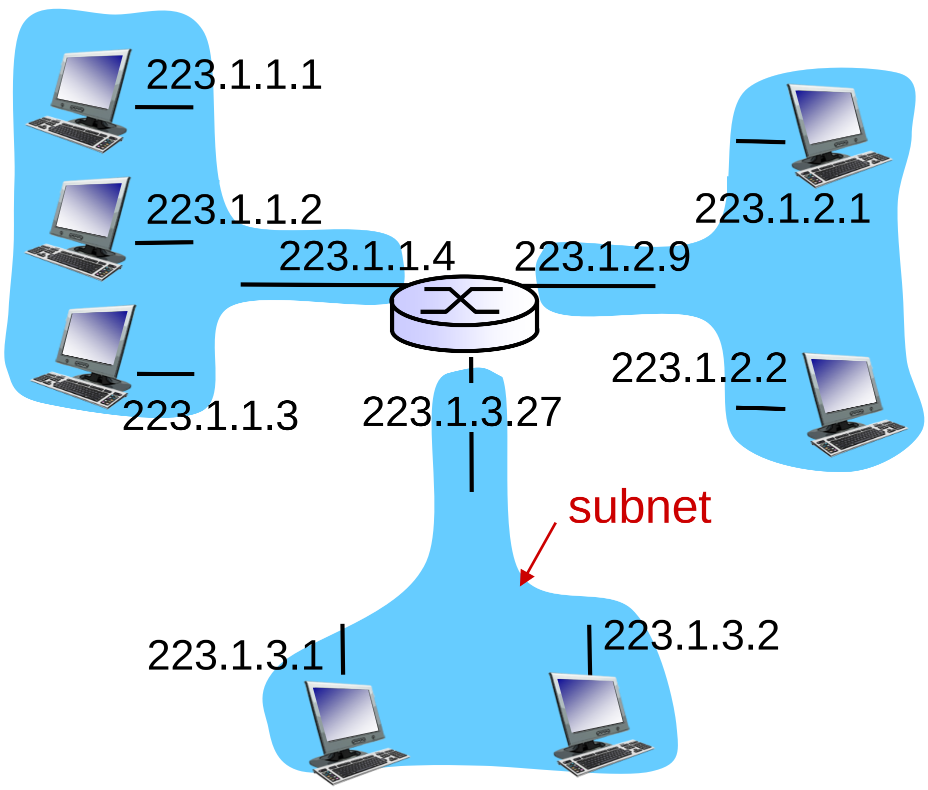

*A:* wired Ethernet interfaces connected by Ethernet switches

*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

# 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*

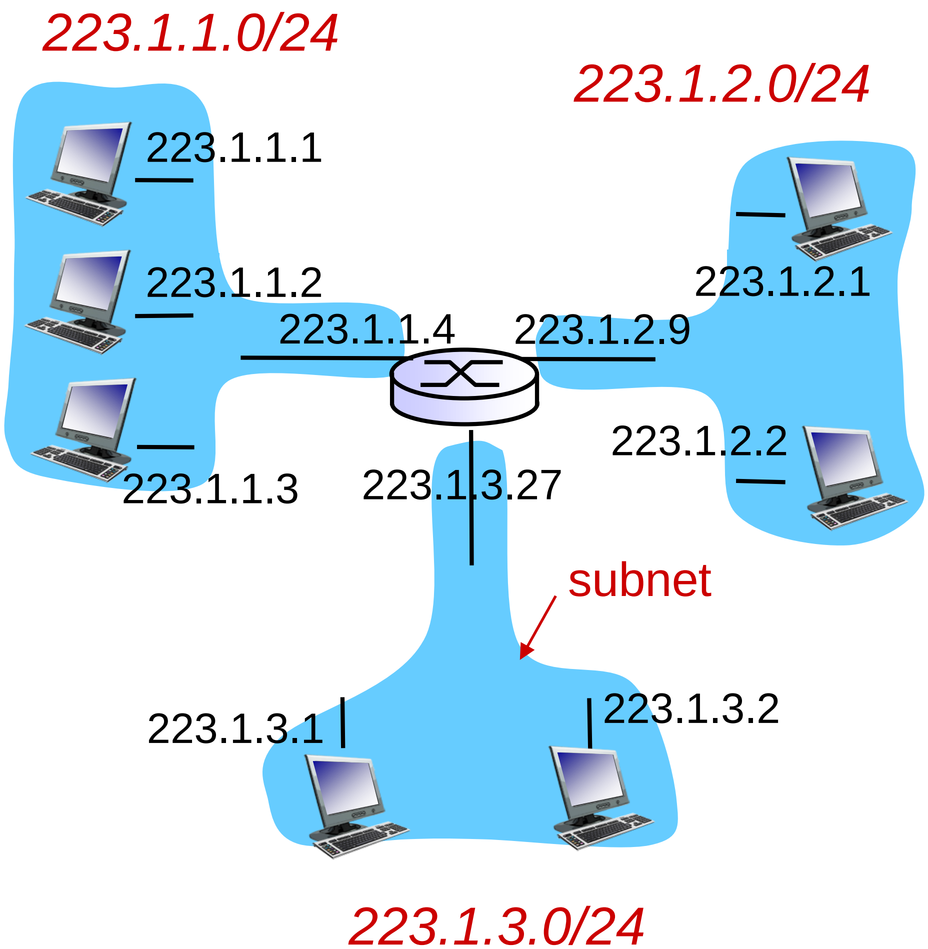



network consisting of 3 subnets

# Subnets

*recipe*

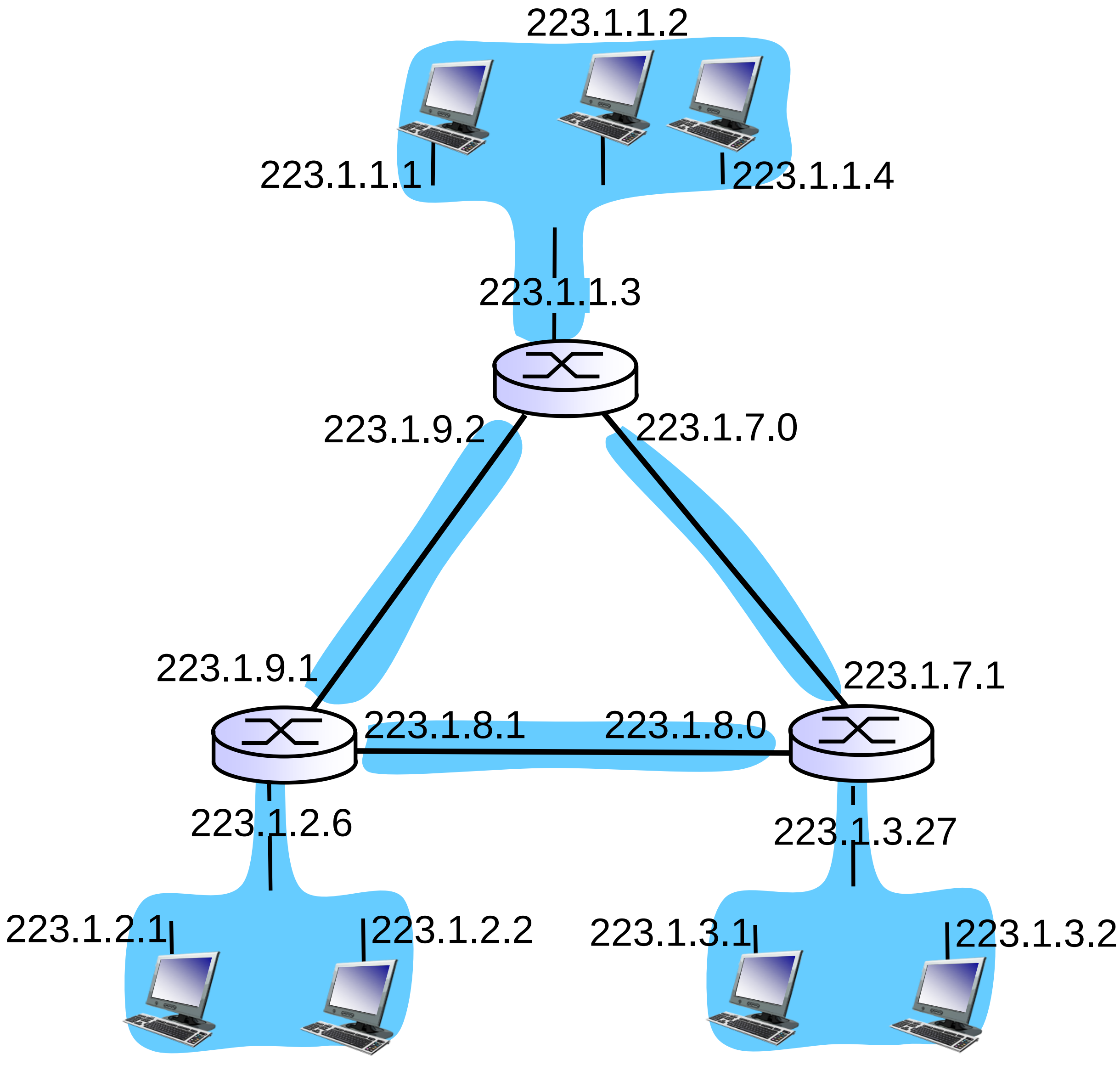
- to determine the subnets, detach each interface from its host or router, creating islands of isolated networks
- each isolated network is called a *subnet*




subnet mask: /24

# Subnets

how many?



# IP addressing: CIDR

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

- Replaces older “class” based system (A, B, C, D)
- 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

# CIDR Subnets and masks

## 192.168.1.0/24

- For this particular address, the first 24 bits are the network prefix used for routing:
  - 192.168.1 =
    11000000 10101000 00000001
- We write this prefix like this:
  - 192.168.1.0 =
    11000000 10101000 00000001 00000000
- The remaining 8 bits are used for *host addressing*:
  - 192.168.1.1 - 192.168.1.254
  - .0 and .255 are special reserved addresses.

# CIDR Subnets and masks

- The subnet mask is a binary number that, when applied by a bitwise AND operation to *any* IP address in the network, yields the routing prefix.
- 192.168.1.0/24 Network prefix has the corresponding subnet mask of 255.255.255.0.
- Example:
  - 192 . 168 . 1 . 10
    11000000 10101000 00000001 00001010
  - 255 . 255 . 255 . 0
    11111111 11111111 11111111 00000000
  - AND
    11000000 10101000 00000001 00001010
    11111111 11111111 11111111 00000000

    11000000 10101000 00000001 00000000 = 192.168.1.0

# Subnets and masks

| | | | |
|---|---|---|---|
| 255.255.255.255 | 11111111.11111111.11111111.11111111 | /32 | **Host (single addr)** |
| 255.255.255.254 | 11111111.11111111.11111111.11111110 | /31 | **Unuseable** |
| 255.255.255.252 | 11111111.11111111.11111111.11111100 | /30 | 2 useable |
| 255.255.255.248 | 11111111.11111111.11111111.11111000 | /29 | 6 useable |
| 255.255.255.240 | 11111111.11111111.11111111.11110000 | /28 | 14 useable |
| 255.255.255.224 | 11111111.11111111.11111111.11100000 | /27 | 30 useable |
| 255.255.255.192 | 11111111.11111111.11111111.11000000 | /26 | 62 useable |
| 255.255.255.128 | 11111111.11111111.11111111.10000000 | /25 | 126 useable |
| 255.255.255.0 | 11111111.11111111.11111111.00000000 | /24 | "Class C" **254 useable** |
| 255.255.254.0 | 11111111.11111111.11111110.00000000 | /23 | 2 Class C's |
| 255.255.252.0 | 11111111.11111111.11111100.00000000 | /22 | 4 Class C's |
| 255.255.248.0 | 11111111.11111111.11111000.00000000 | /21 | 8 Class C's |
| 255.255.240.0 | 11111111.11111111.11110000.00000000 | /20 | 16 Class C's |
| 255.255.224.0 | 11111111.11111111.11100000.00000000 | /19 | 32 Class C's |
| 255.255.192.0 | 11111111.11111111.11000000.00000000 | /18 | 64 Class C's |
| 255.255.128.0 | 11111111.11111111.10000000.00000000 | /17 | 128 Class C's |
| 255.255.0.0 | 11111111.11111111.00000000.00000000 | /16 | "Class B" |
| 255.254.0.0 | 11111111.11111110.00000000.00000000 | /15 | 2 Class B's |
| 255.252.0.0 | 11111111.11111100.00000000.00000000 | /14 | 4 Class B's |
| 255.248.0.0 | 11111111.11111000.00000000.00000000 | /13 | 8 Class B's |
| 255.240.0.0 | 11111111.11110000.00000000.00000000 | /12 | 16 Class B's |
| 255.224.0.0 | 11111111.11100000.00000000.00000000 | /11 | 32 Class B's |
| 255.192.0.0 | 11111111.11000000.00000000.00000000 | /10 | 64 Class B's |
| 255.128.0.0 | 11111111.10000000.00000000.00000000 | /9 | 128 Class B's |
| 255.0.0.0 | 11111111.00000000.00000000.00000000 | /8 | "Class A" |
| 254.0.0.0 | 11111110.00000000.00000000.00000000 | /7 | |
| 252.0.0.0 | 11111100.00000000.00000000.00000000 | /6 | |
| 248.0.0.0 | 11111000.00000000.00000000.00000000 | /5 | |
| 240.0.0.0 | 11110000.00000000.00000000.00000000 | /4 | |
| 224.0.0.0 | 11100000.00000000.00000000.00000000 | /3 | |
| 192.0.0.0 | 11000000.00000000.00000000.00000000 | /2 | |
| 128.0.0.0 | 10000000.00000000.00000000.00000000 | /1 | |
| 0.0.0.0 | 00000000.00000000.00000000.00000000 | /0 | **IP space** |

# IP addresses: how to get one?

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

- Statically assigned - hard-coded by system admin:
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- Dynamically assigned:
  - DHCP: Dynamic Host Configuration Protocol: dynamically get address from a server
  - "plug-and-play"

# DHCP: Dynamic Host Configuration Protocol

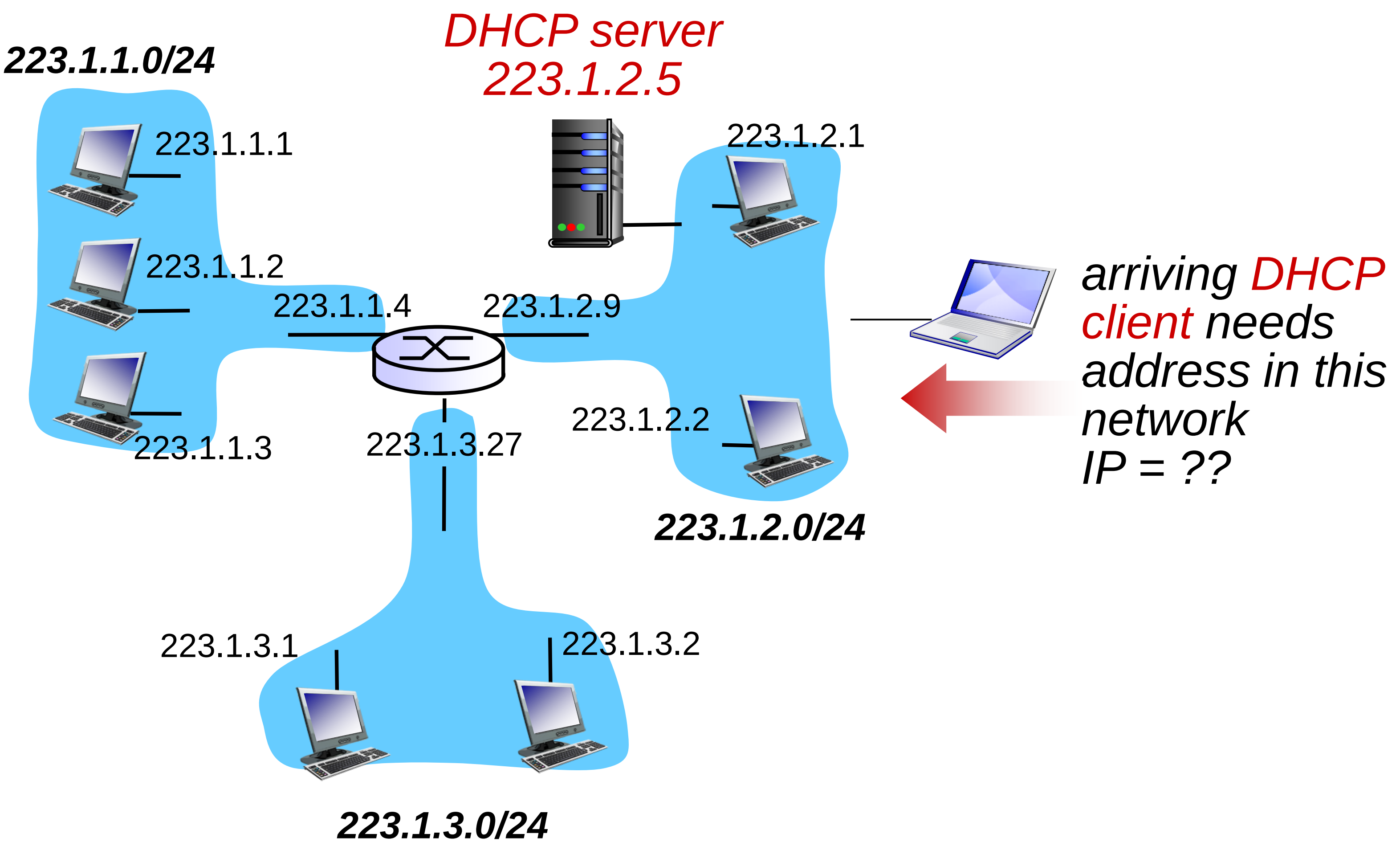
*goal:* allow host to *dynamically* obtain its IP address from network server when it joins network, and keep it for a time (a "lease")

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

*DHCP overview:*

- host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

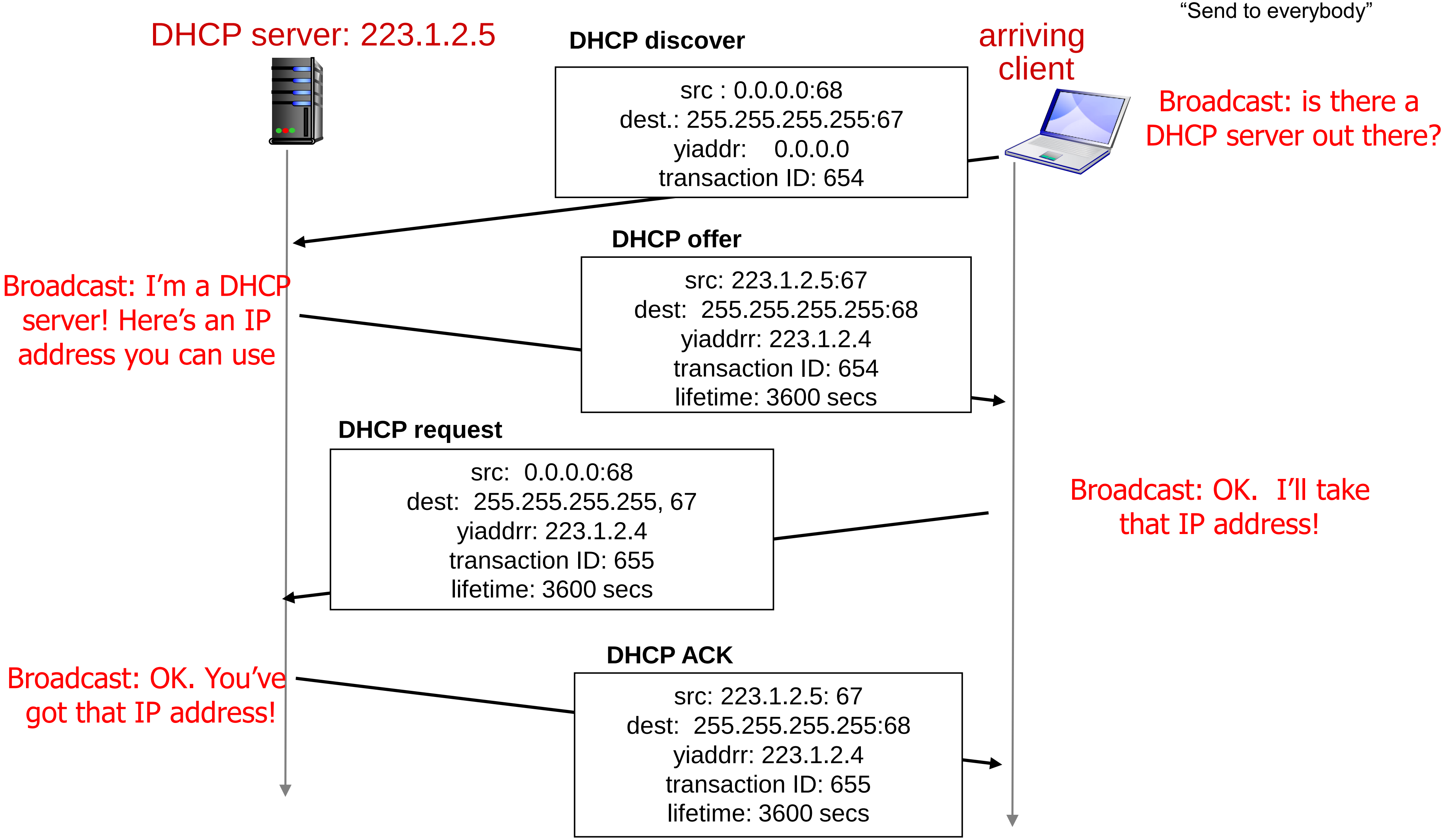
# DHCP client-server scenario



# DHCP client-server scenario

0.0.0.0 =
“I don’t have an address”

255.255.255.255 =
“Send to everybody”

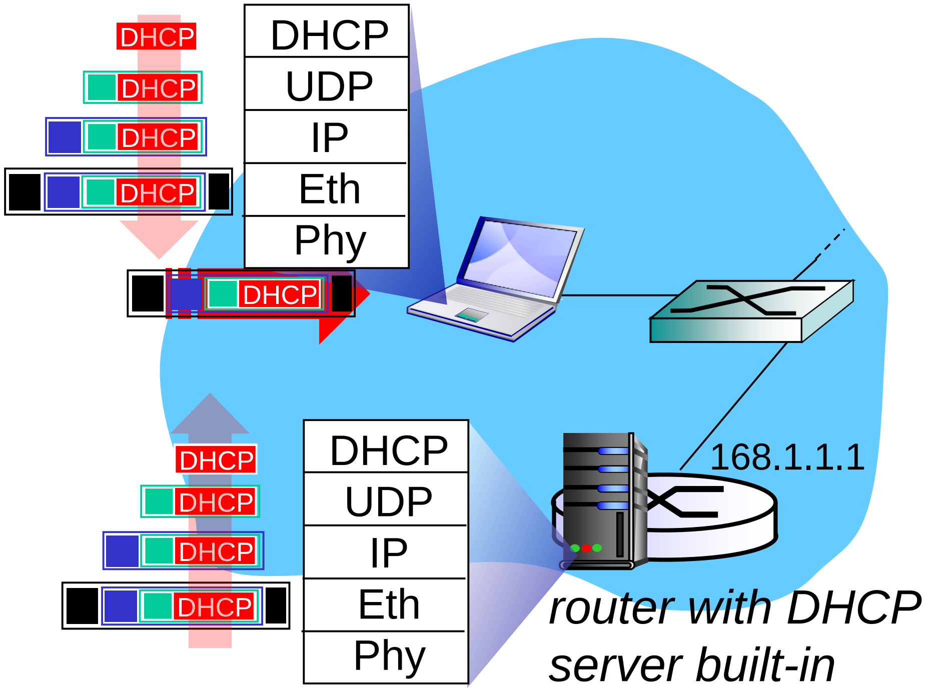


# DHCP: more than IP addresses

DHCP can return more than just allocated Iv4 address on subnet:

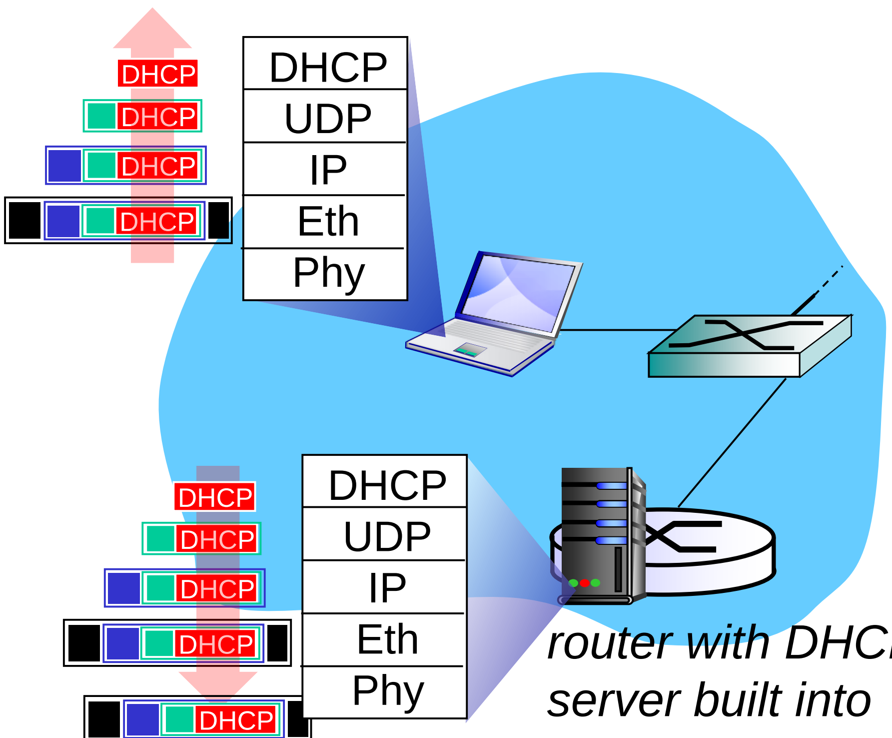
- address of first-hop router for client (default gateway)
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)
- Windows Domain name
- IPv6 addressing information

# 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

# DHCP: example




*router with DHCP server built into router*

- 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

# DHCP: Wireshark output (home LAN)

request

Message type: **Boot Request (1)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
**Transaction ID: 0x6b3a11b7**
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
**Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)**
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) **DHCP Message Type = DHCP Request**
Option: (61) Client identifier
  Length: 7; Value: 010016D323688A;
  Hardware type: Ethernet
  Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
**Option: (55) Parameter Request List**
  Length: 11; Value: 010F03062C2E2F1F21F92B
  **1 = Subnet Mask; 15 = Domain Name**
  **3 = Router; 6 = Domain Name Server**
  44 = NetBIOS over TCP/IP Name Server
  ……

reply

Message type: **Boot Reply (2)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
**Transaction ID: 0x6b3a11b7**
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
**Client IP address: 192.168.1.101 (192.168.1.101)**
Your (client) IP address: 0.0.0.0 (0.0.0.0)
**Next server IP address: 192.168.1.1 (192.168.1.1)**
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
**Option: (t=53,l=1) DHCP Message Type = DHCP ACK**
**Option: (t=54,l=4) Server Identifier = 192.168.1.1**
**Option: (t=1,l=4) Subnet Mask = 255.255.255.0**
**Option: (t=3,l=4) Router = 192.168.1.1**
**Option: (6) Domain Name Server**
  **Length: 12; Value: 445747E2445749F244574092;**
  **IP Address: 68.87.71.226;**
  **IP Address: 68.87.73.242;**
  **IP Address: 68.87.64.146**
**Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."**

# IP addresses: how to get one?

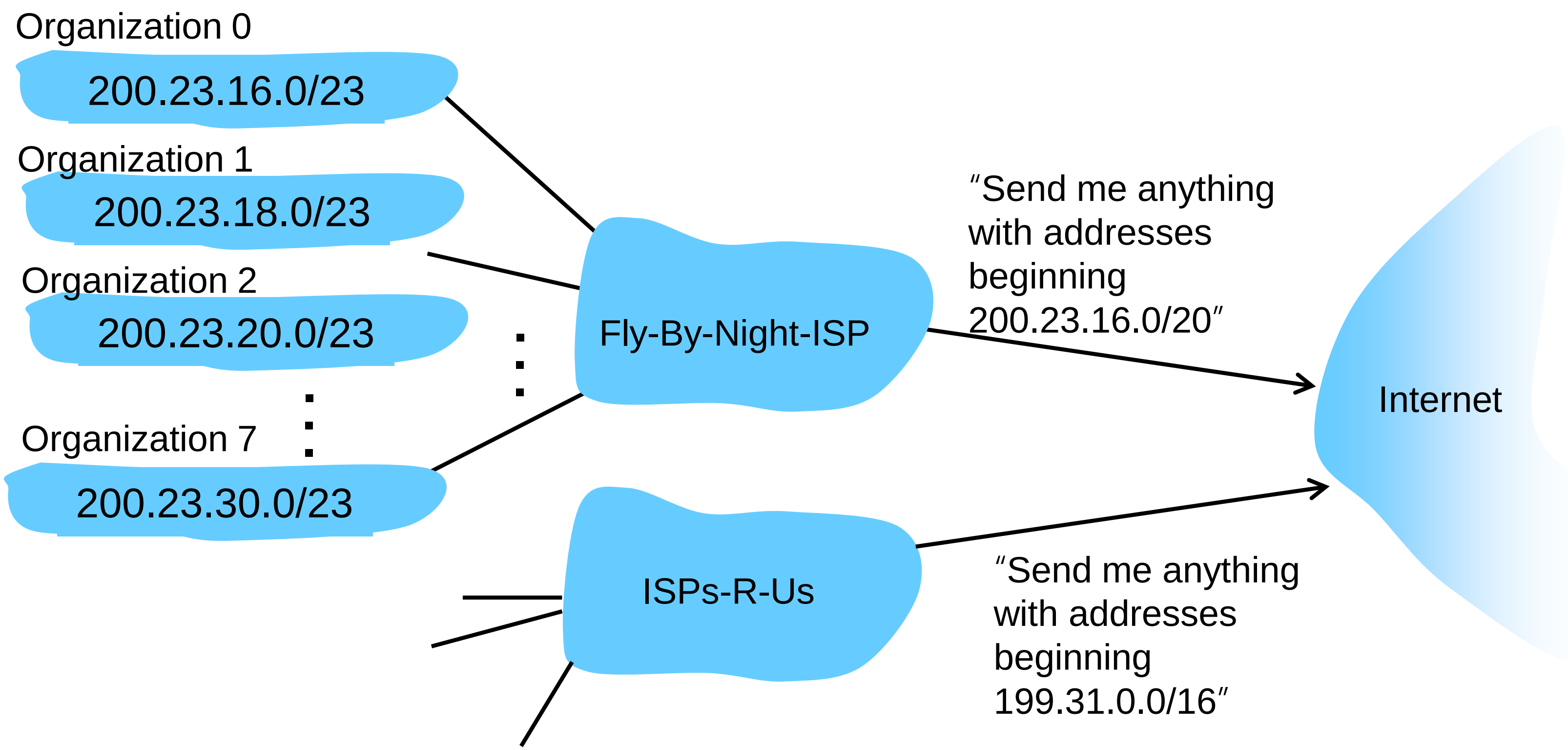
*Q:* how does *network* get subnet part of IP addr?

*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 |

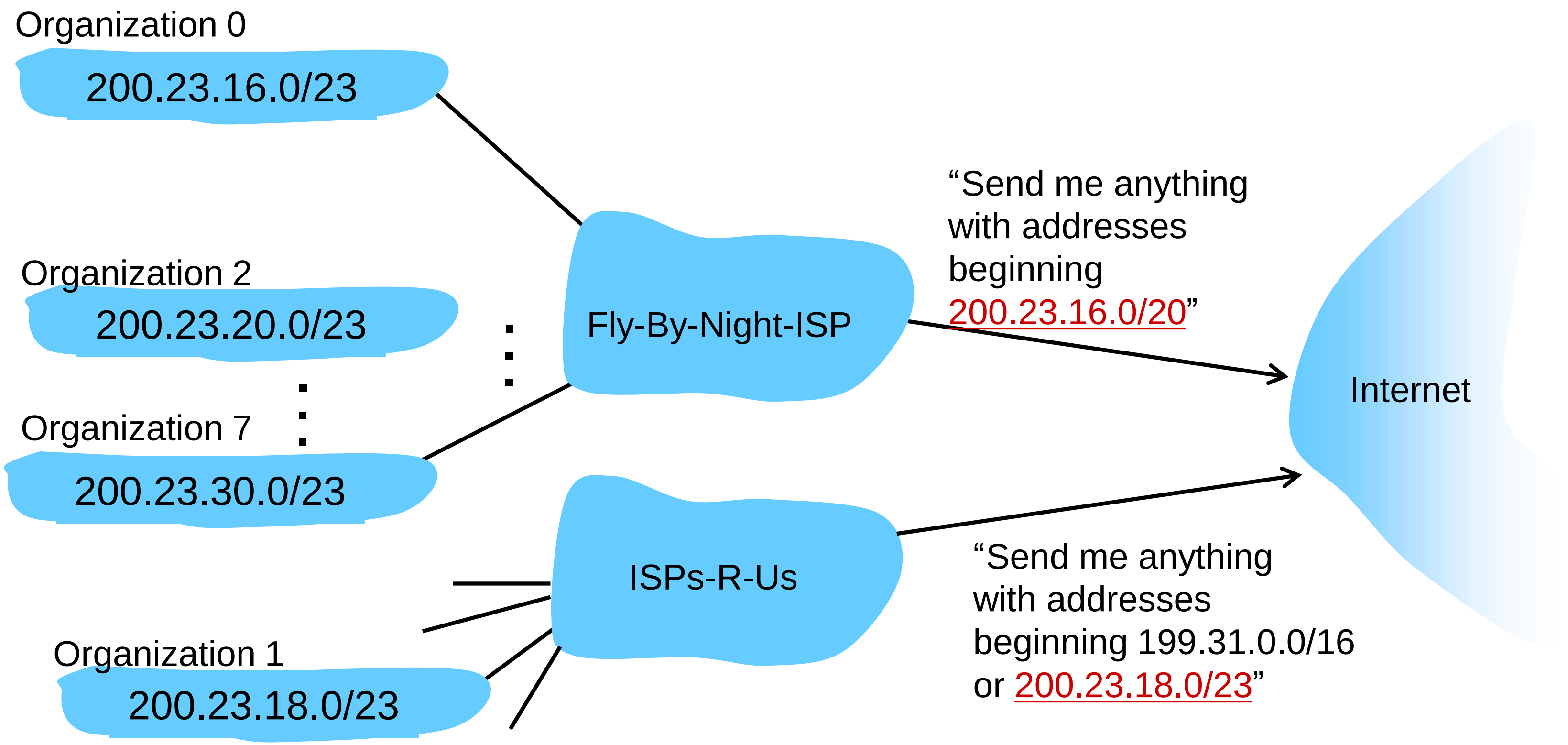
# Hierarchical addressing: route aggregation

hierarchical addressing allows efficient advertisement of routing information:



# Hierarchical addressing: more specific routes

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



# 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 addresses
- manages DNS
- assigns domain names, resolves disputes
- There are no more IPv4 addresses to give out!

# Who's got them…

**Companies orgs with IPv4 /8 blocks from IANA – Internet Assigned Numbers Authority (a dept. of ICANN)**

| **Owner** | **/8 Blocks** | **~IP addresses** |
|---|---|---|
| US Military (Department of Defense etc.) | 12 | 201 million |
| Level 3 Communications, Inc. | 2 | 33 million |
| Hewlett-Packard | 2 | 33 million |
| AT&T Bell Laboratories (Alcatel-Lucent) | 1 | 16 million |
| AT&T Global Network Services | 1 | 16 million |
| Bell-Northern Research (Nortel Networks) | 1 | 16 million |
| Amateur Radio Digital Communications | 1 | 16 million |
| Apple Computer Inc. | 1 | 16 million |
| Cap Debis CCS (Mercedes-Benz) | 1 | 16 million |
| Computer Sciences Corporation | 1 | 16 million |
| Deparment of Social Security of UK | 1 | 16 million |
| E.I. duPont de Nemours and Co., Inc. | 1 | 16 million |
| Eli Lily and Company | 1 | 16 million |
| Ford Motor Company | 1 | 16 million |
| General Electric Company | 1 | 16 million |
| Halliburton Company | 1 | 16 million |
| IBM | 1 | 16 million |
| Interop Show Network | 1 | 16 million |
| Merck and Co., Inc. | 1 | 16 million |
| MERIT Computer Network | 1 | 16 million |
| Massachusetts Institute of Technology | 1 | 16 million |
| Performance Systems International (Cogent) | 1 | 16 million |
| Prudential Equity Group, LLC | 1 | 16 million |
| Société Internationale De Telecommunications Aero. | 1 | 16 million |
| U.S. Postal Service | 1 | 16 million |
| UK Ministry of Defence | 1 | 16 million |
| Xerox Corporation | 1 | 16 million |

http://royal.pingdom.com/2008/02/13/where-did-all-the-ip-numbers-go-the-us-department-of-defense-has-them/