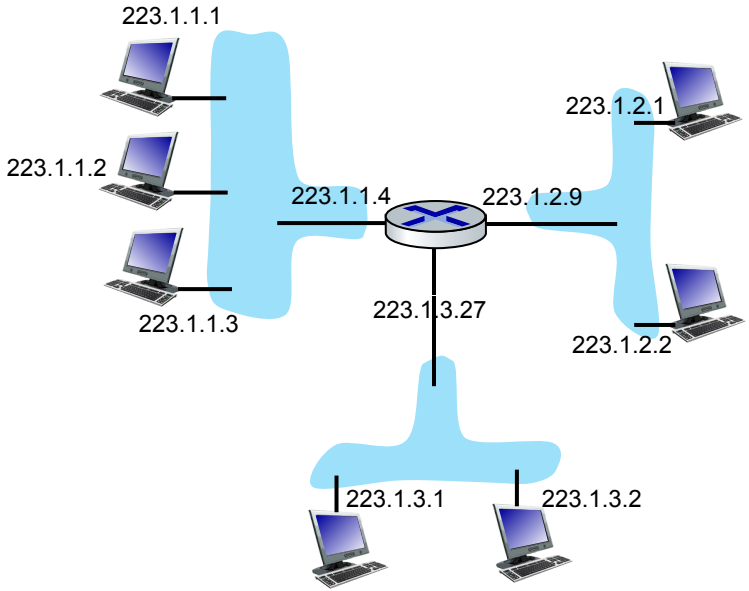
Internet Addressing

IP addressing: introduction

- IP address: 32-bit identifier 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)



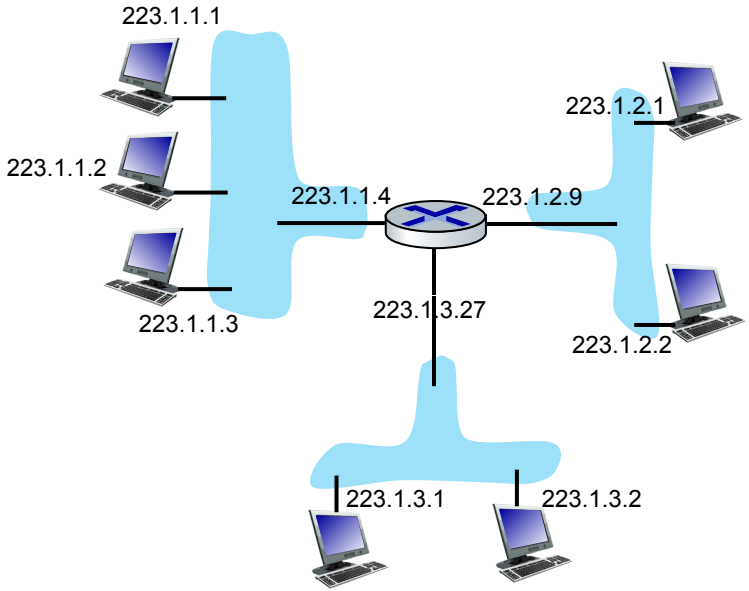
dotted-decimal IP address notation:

223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

IP addressing: introduction

- IP address: 32-bit identifier 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)



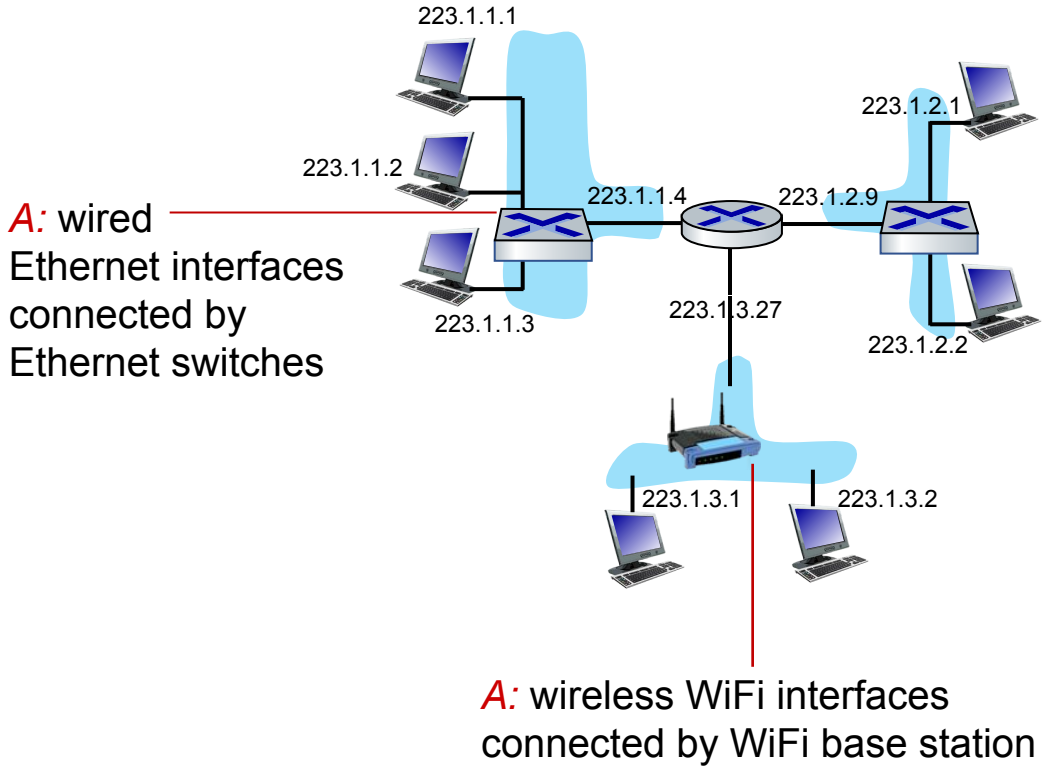
dotted-decimal IP address notation:

223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

IP addressing: introduction

Q: how are interfaces actually connected?



Grouping IP addresses by prefixes

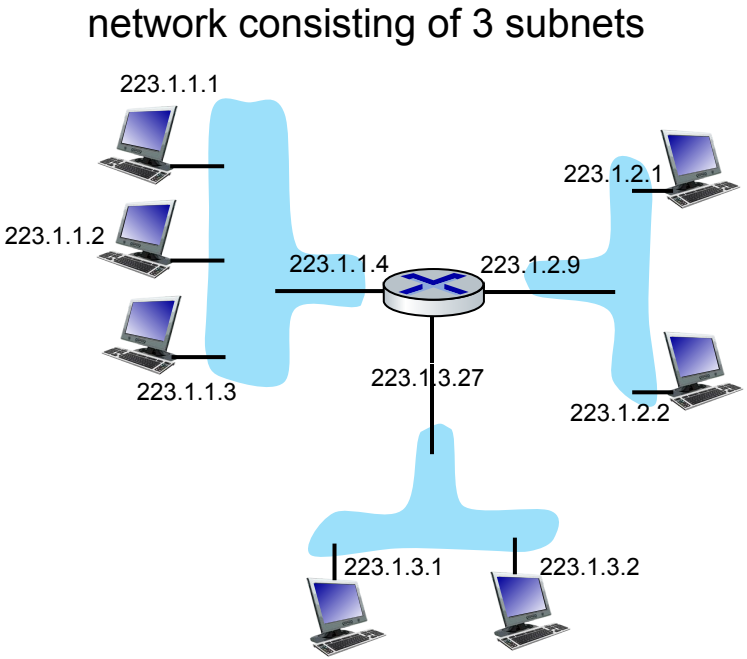
- IP addresses can be grouped based on a shared prefix of a specified length

- Example: consider two IP addresses:
  - 128.95.1.80 and 128.95.1.4
  - The addresses share a prefix of (bit) length 24: 128.95.1
  - The addresses have different suffixes of (bit) length 8

- IP addresses: prefix corresponds to the network component and the suffix to an endpoint/host component of the address

Subnets

- A subnet, or subnetwork, is a network inside a network.
- Subnets make networks more efficient.
- Through subnetting, network traffic can travel a shorter distance without passing through unnecessary routers to reach its destination

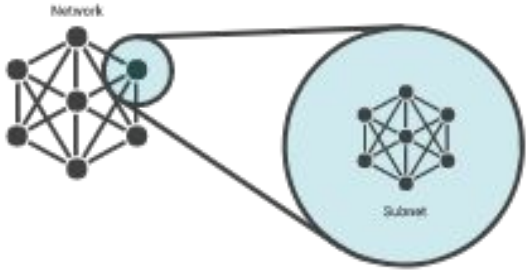


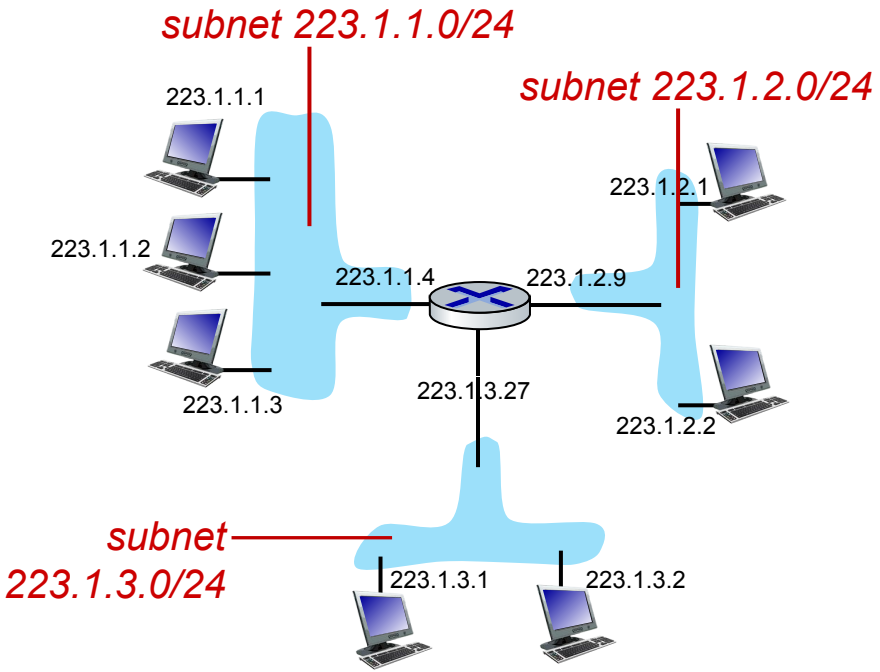
- IP addresses have structure:
  - subnet part: devices in same subnet have common high order bits
  - host part: remaining low order bits

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*



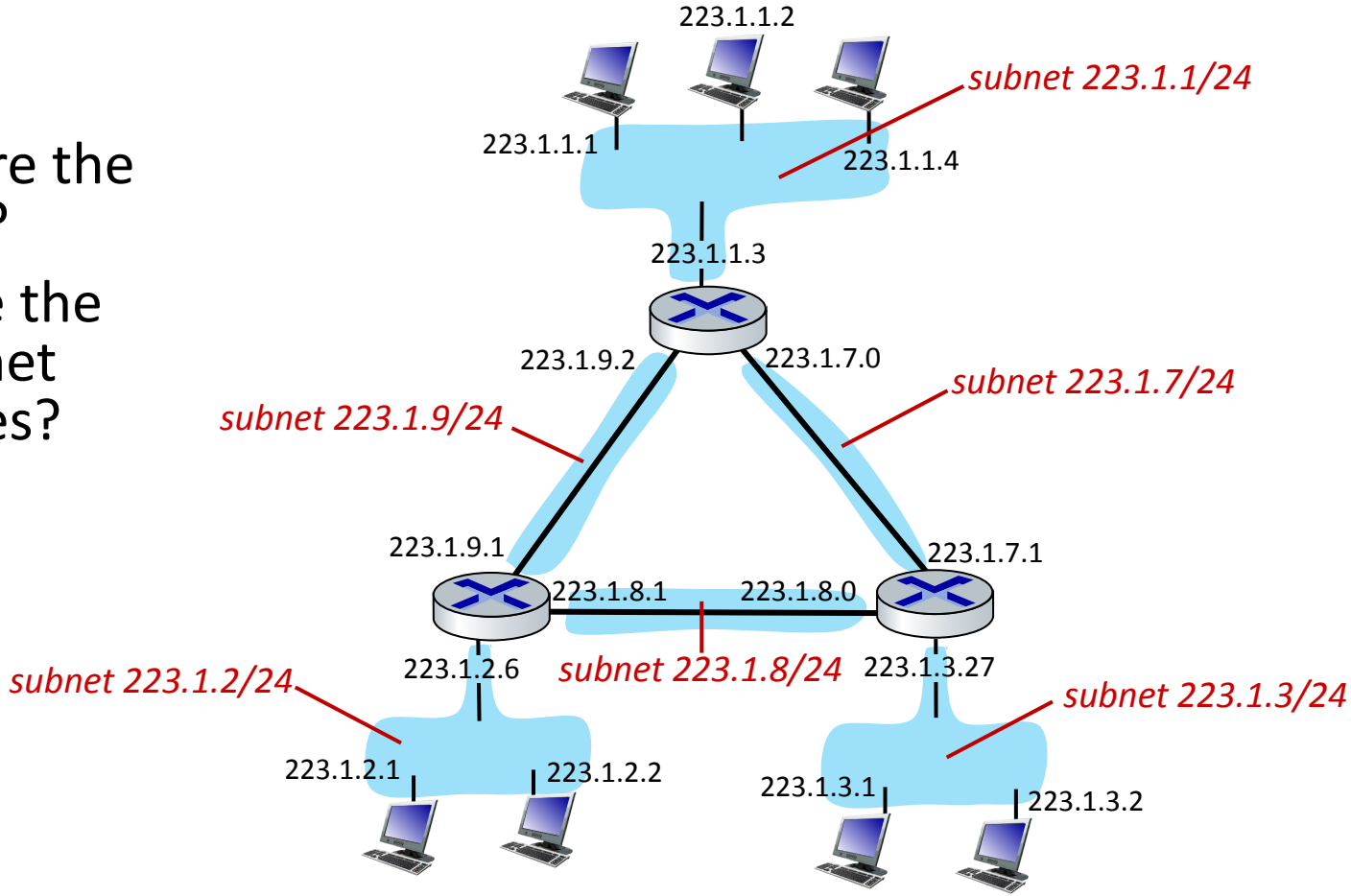


subnet mask: /24

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

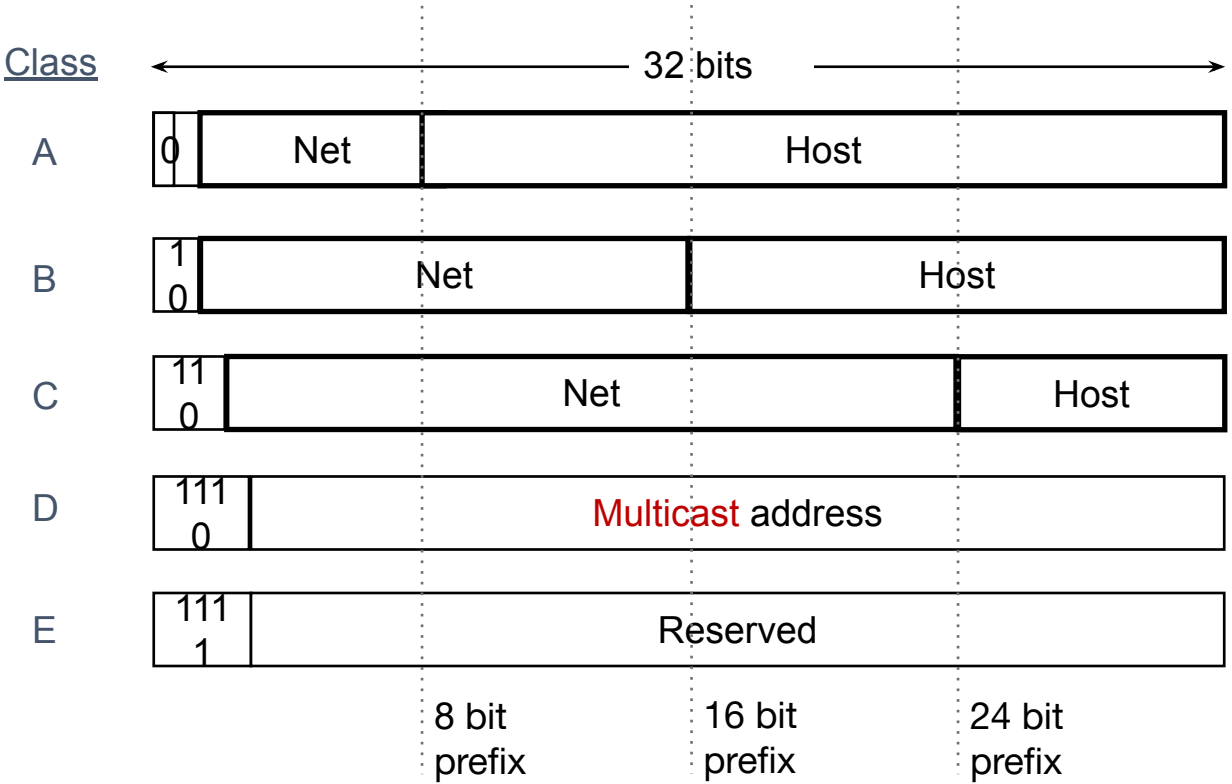
Subnets

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



Classful IPv4 addressing

Classful IPv4 addressing



0.x.x.x – 127.x.x.x

Unicast: single endpoint dest

128.x.x.x – 191.x.x.x

Unicast: single endpoint dest

192.x.x.x – 223.x.x.x

Unicast: single endpoint dest

224.x.x.x – 239.x.x.x

Destination is a group of hosts

240.x.x.x – 255.x.x.x

First octet of IP address gives you the prefix length.

Classful IPv4 addressing

- Class A:
  - For very large organizations
  - 2^{24} = 16 million hosts allowed
- Class B:
  - For large organizations
  - 2^{16} = 65 thousand hosts allowed
- Class C
  - For small organizations
  - 2^{8} = 255 hosts allowed
- Class D
  - Multicast addresses
  - No network/host hierarchy

Problems with classful addressing

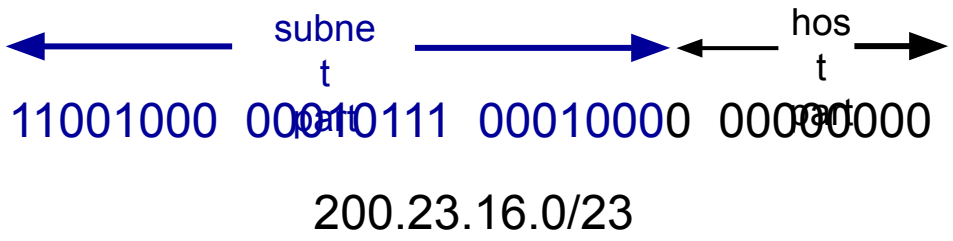
- IP prefixes are allocated to organizations (e.g., Rutgers) by Internet Registry organizations (e.g., ARIN, in North America)
- Many organizations required something bigger than class C address, but smaller than a class A (or even B) address
- However, the Internet was running out of class B addresses
- Too many networks required multiple class C addresses
- Not enough nets in class A for large + medium organizations
- Key issue: Classful addressing is too coarse-grained: The addressing strategy must allow for greater diversity of network sizes

Classless IPv4 addressing (CIDR)

Classless IPv4 addressing (also called CIDR)

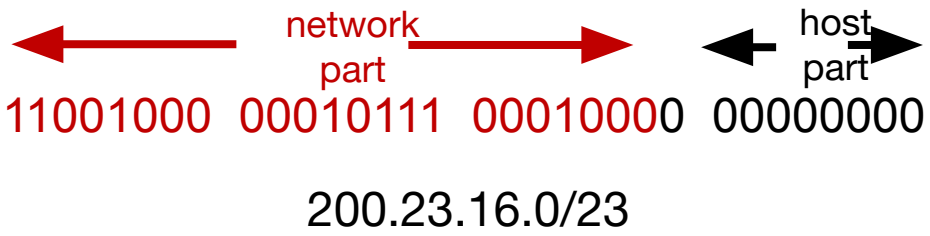
CIDR: Classless InterDomain Routing (pronounced “cider”)

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



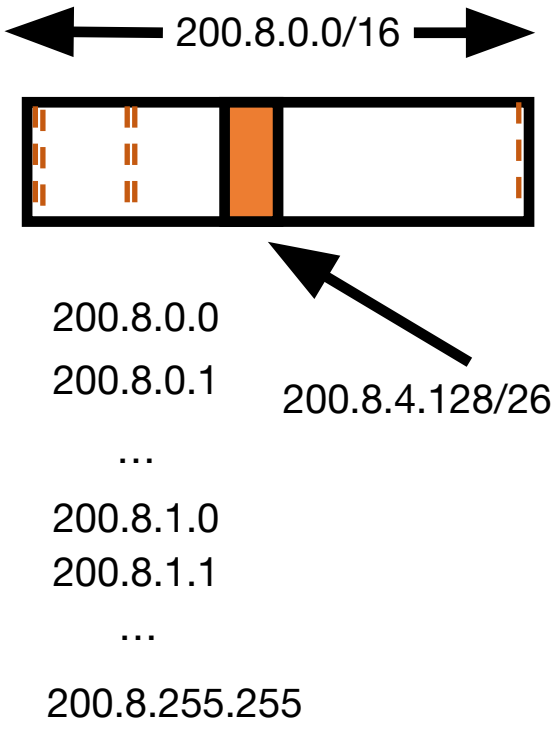
Classless IPv4 addressing

- Key idea: Network component of the address (ie: prefix) can have any length (usually from 8—32)
- Address format: a.b.c.d/x, where x is the prefix length
  - Customary to use 0s for all suffix bits



CIDR

- An ISP can obtain a block of addresses and partition this further to its customers
- Say an ISP has 200.8.0.0/16 address (65K addresses).
- The ISP has customer who needs only 64 addresses starting from 200.8.4.128
- Then that block can be specified as 200.8.4.128/26
- 200.8.4.128/26 is “inside” 200.8.0.0/16



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

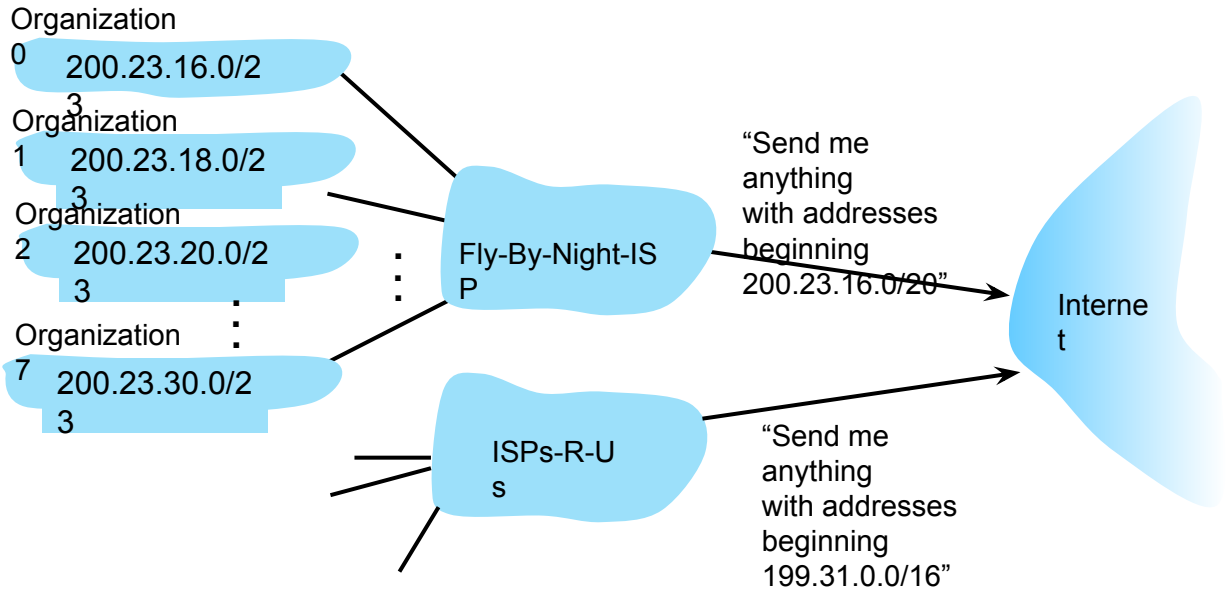
ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20

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

Organization 0	11001000 00010111 0001000**0**	00000000	200.23.16.0/23
Organization 1	11001000 00010111 0001001**0**	00000000	200.23.18.0/23
Organization 2	11001000 00010111 0001010**0**	00000000	200.23.20.0/23
...	.....	....	....
Organization 7	11001000 00010111 0001111**0**	00000000	200.23.30.0/23

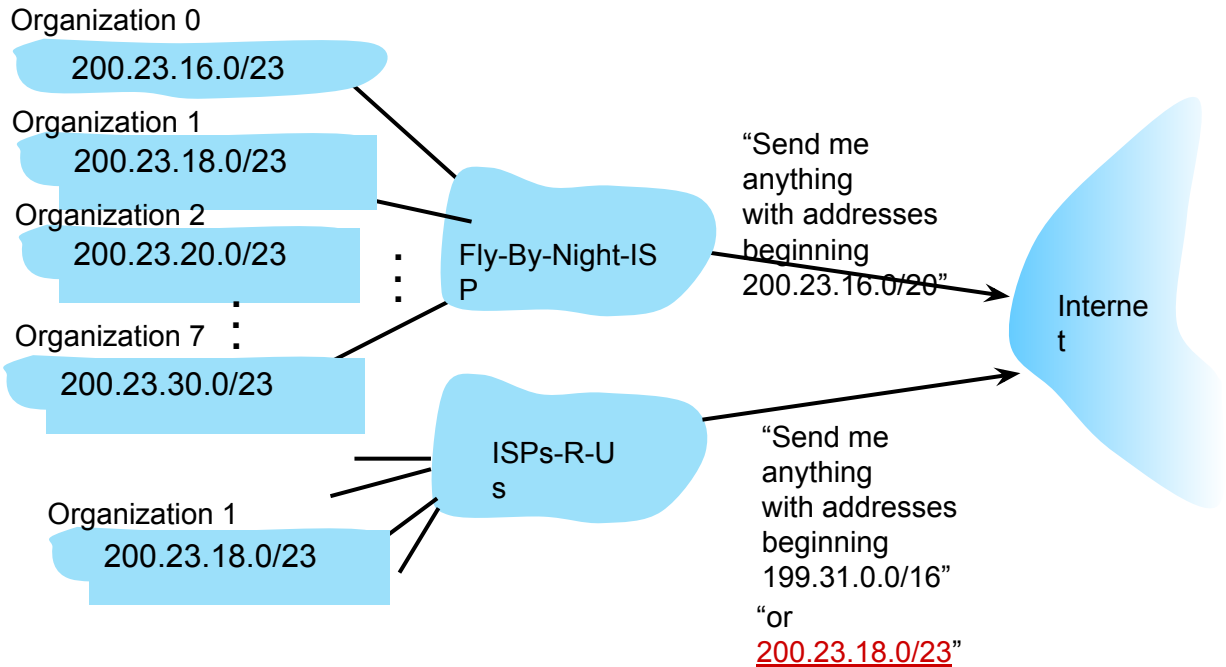
Hierarchical addressing: route aggregation

hierarchical addressing allows efficient advertisement of routing information:



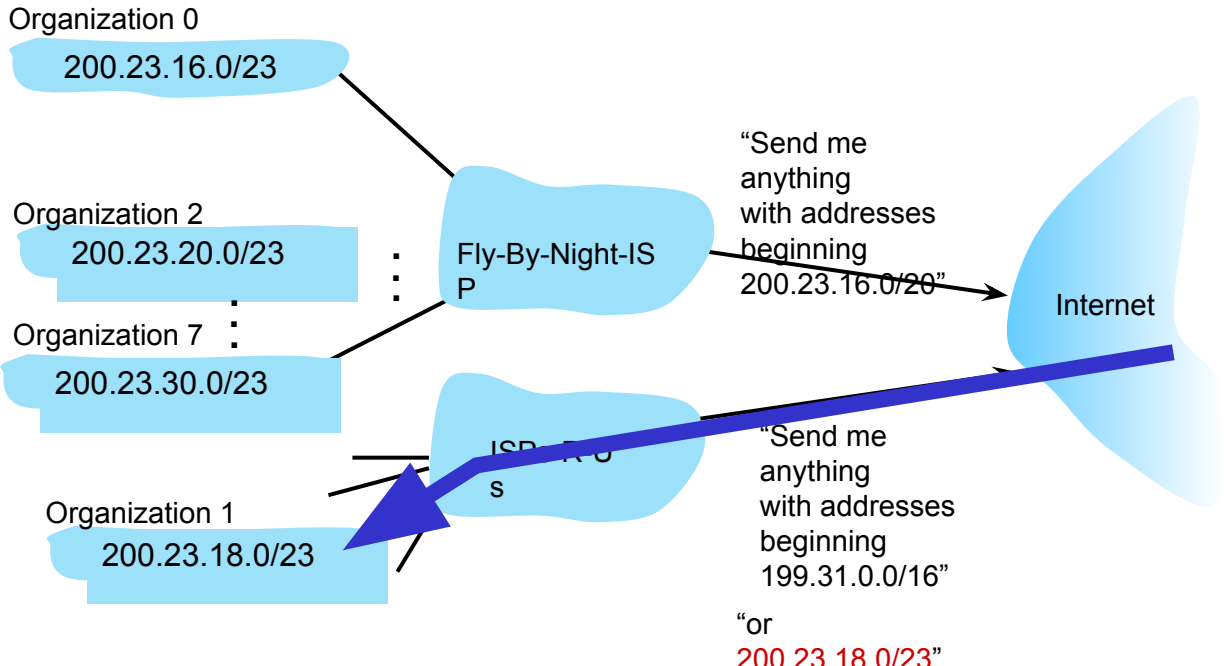
Hierarchical addressing: more specific routes

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



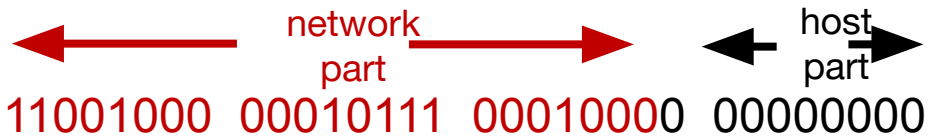
Hierarchical addressing: more specific routes

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



Netmask (or subnet mask)

- An alternative to denote the IP prefix length of an organization
- 32 bits: a 1-bit denotes a prefix bit position. 0 is the host part.



200.23.16.0/23

Netmask: 255.255.254.0

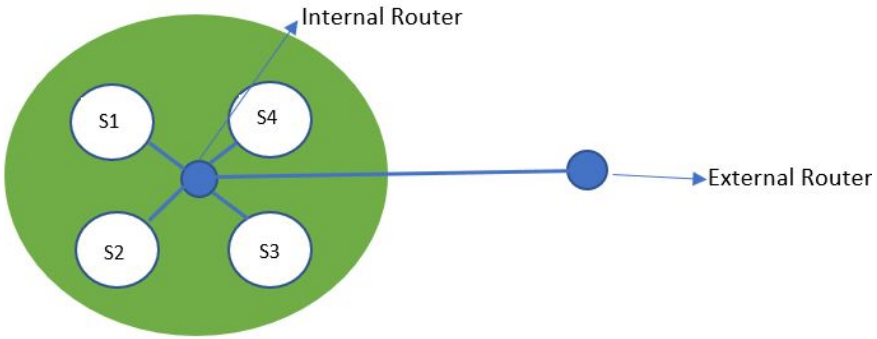
Detecting addresses from same network

- Given IP addresses A and B, and netmask M.
  1. Compute logical AND (A & M).
  2. Compute logical AND (B & M).
  3. If (A & M) == (B & M) then A and B are on the same subnet.

- Ex: A = 165.230.82.52, B = 165.230.24.93, M = 255.255.128.0

- A and B are in the same network according to the netmask
- A & M == B & M == 165.230.0.0

Actual View in reality:



The first and last IP addresses are reserved for **network ID** and **directed broadcast address** in every subnet

run command: **route -n**

Network ID	Subnet mask	Interface
200.1.2.0	255.255.255.192	A
200.1.2.64	255.255.255.192	B
200.1.2.128	255.255.255.192	C
200.1.2.192	255.255.255.192	D

Default entry has network id as 0.0.0.0.

Example

One of the addresses in a block is 167.199.170.82/27. Find the number of addresses in the network, the first address, and the last address.

Solution

The value of n is 27. The network mask has twenty-seven 1s and five 0s. It is 255.255.255.224.

a. The number of addresses in the network is $2^{32 - n} = 32$.

b. We use the AND operation to find the first address (network address). The first address is 167.199.170.64/27.

Address in binary:	10100111 11000111 10101010 01010010
Network mask:	11111111 11111111 11111111 11100000
First address:	10100111 11000111 10101010 01000000

Cont..

c. To find the last address, we first find the complement of the network mask and then OR it with the given address: The last address is 167.199.170.95/27.

Address in binary:	10100111	11000111	10101010	01010010
Complement of network mask:	00000000	00000000	00000000	00011111
Last address:	10100111	11000111	10101010	01011111

Example

One of the addresses in a block is 17.63.110.114/24. Find the number of addresses, the first address, and the last address in the block.

Solution

The network mask is 255.255.255.0.

a. The number of addresses in the network is $2^{32-24} = 256$.

b. To find the first address, we use the short cut methods discussed early in the chapter. The first address is 17.63.110.0/24.

Address:	17	.	63	.	110	.	114
Network mask:	255	.	255	.	255	.	0
First address (AND):	17	.	63	.	110	.	0

Cont..

c. To find the last address, we use the complement of the network mask and the first short cut method we discussed before. The last address is 17.63.110.255/24.

Example

One of the addresses in a block is 110.23.120.14/20. Find the number of addresses, the first address, and the last address in the block.

Solution

The network mask is 255.255.240.0.

a. The number of addresses in the network is $2^{32 - 20} = 4096$.

b. To find the first address, we apply the first short cut to bytes 1, 2, and 4 and the second short cut to byte 3. The first address is 110.23.112.0/20.

Address:	110	.	23	.	120	.	14
Network mask:	255	.	255	.	240	.	0
First address (AND):	110	.	23	.	112	.	0

Example

c. To find the last address, we apply the first short cut to bytes 1, 2, and 4 and the second short cut to byte 3. The OR operation is applied to the complement of the mask. The last address is 110.23.127.255/20.

Address:	110	.	23	.	120	.	14
Network mask:	0	.	0	.	15	.	255
Last address (OR):	110	.	23	.	127	.	255

Example

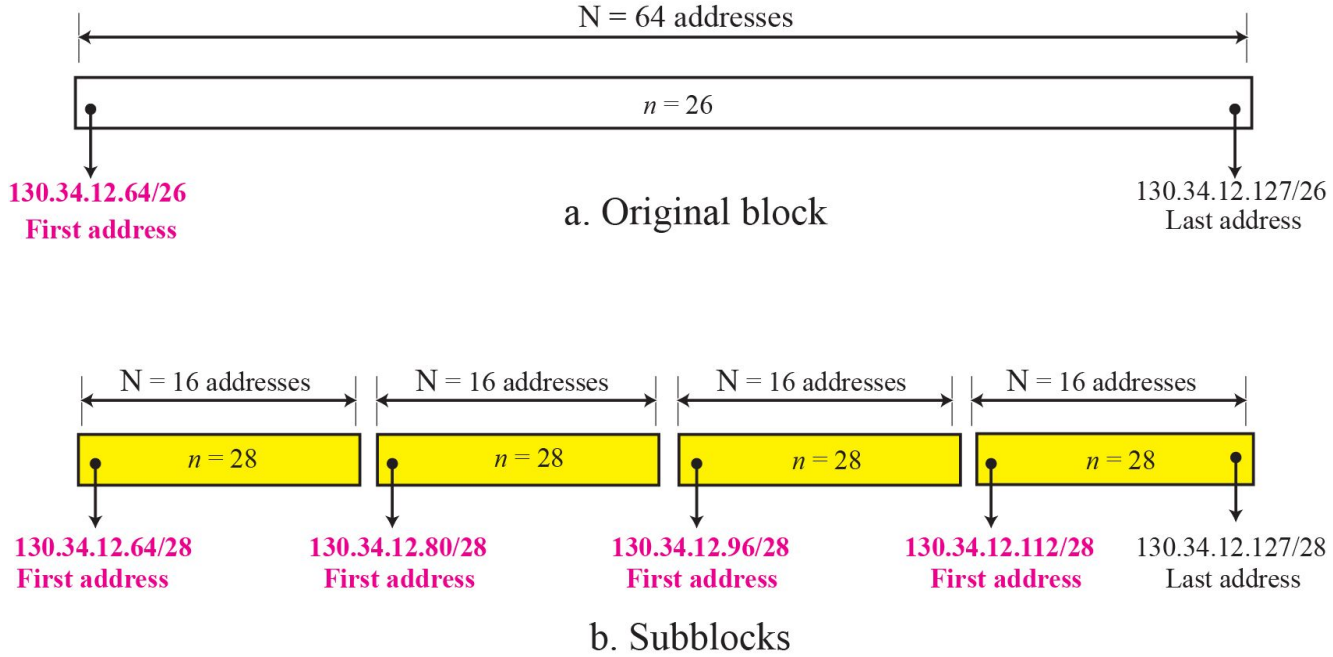
An organization is granted the block 130.34.12.64/26. The organization needs four subnetworks, each with an equal number of hosts. Design the subnetworks and find the information about each network.

Solution

The number of addresses for the whole network can be found as $N = 2^{32-26} = 64$. The first address in the network is 130.34.12.64/26 and the last address is 130.34.12.127/26. We now design the subnetworks:

1. We grant 16 addresses for each subnetwork to meet the first requirement (64/16 is a power of 2).
2. The subnetwork mask for each subnetwork is:

3. We grant 16 addresses to each subnet starting from the first available address. Figure shows the subblock for each subnet. Note that the starting address in each subnetwork is divisible by the number of addresses in that subnetwork.



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 as shown below:

- One subblock of 120 addresses.
- One subblock of 60 addresses.
- One subblock of 10 addresses.

Solution

There are $2^{32 - 24} = 256$ addresses in this block. The first address is 14.24.74.0/24; the last address is 14.24.74.255/24.

a. The number of addresses in the first subblock is not a power of 2. We allocate 128 addresses. The subnet mask is 25. The first address is 14.24.74.0/25; the last address is 14.24.74.127/25.

Example

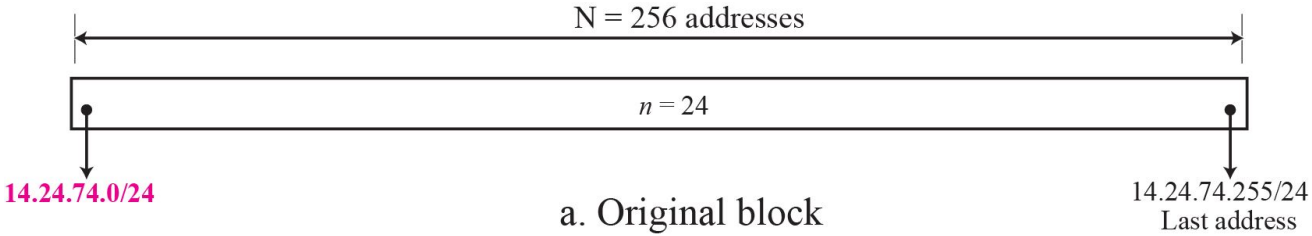
b. The number of addresses in the second subblock is not a power of 2 either. We allocate 64 addresses. The subnet mask is 26. The first address in this block is 14.24.74.128/26; the last address is 14.24.74.191/26.

c. The number of addresses in the third subblock is not a power of 2 either. We allocate 16 addresses. The subnet mask is 28. The first address in this block is 14.24.74.192/28; the last address is 14.24.74.207/28.

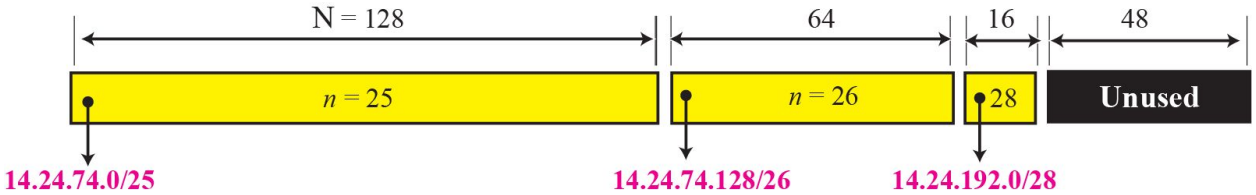
d. If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.209. The last address is 14.24.74.255.

e. Figure shows the configuration of blocks. We have shown the first address in each block.

Solution to Example



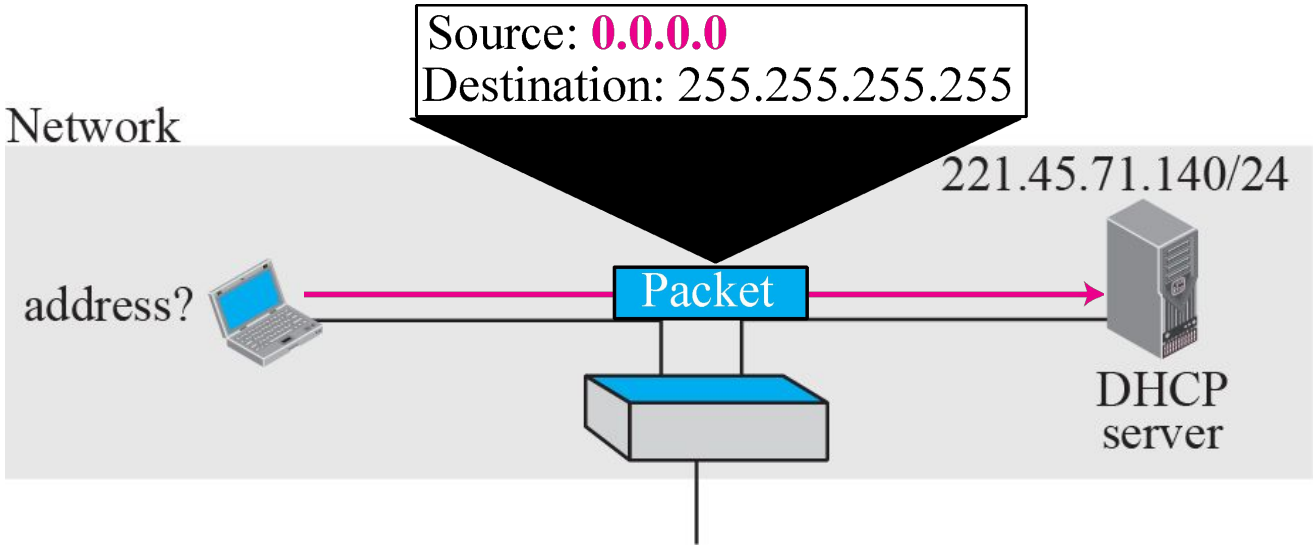
a. Original block



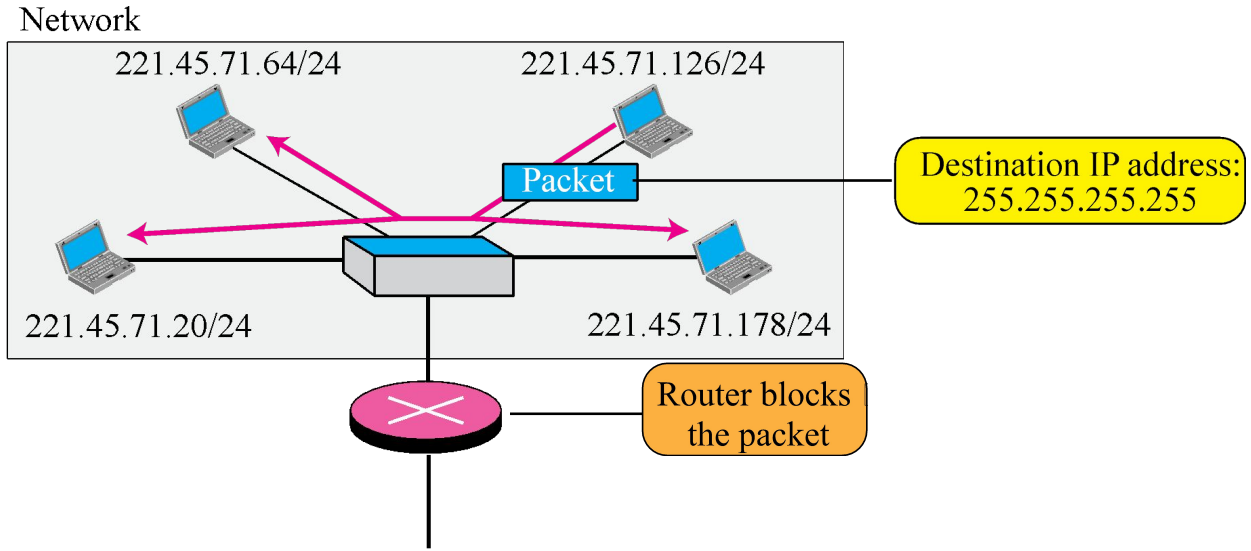
b. Subblocks

Special Address

Example of using the all-zero address



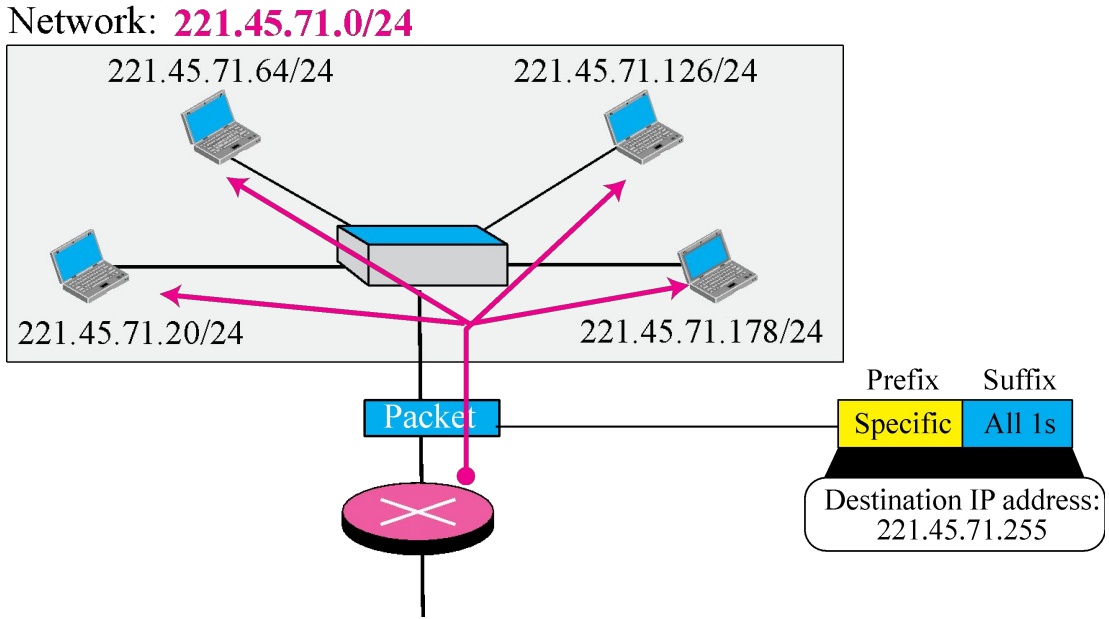
Example of limited broadcast address



Example of loopback address

- 127.0.0.0 to 127.255.255.255 or 127.0.0.0/8 is loopback address block.
- With loopback address you can send packet to yourself.
- Used for troubleshooting
- Check if TCP/IP protocol suite is working properly.
- Generally, we use ping 127.0.0.1 to test it.

Example of a directed broadcast address



IP addresses: how to get one?

That's actually two questions:

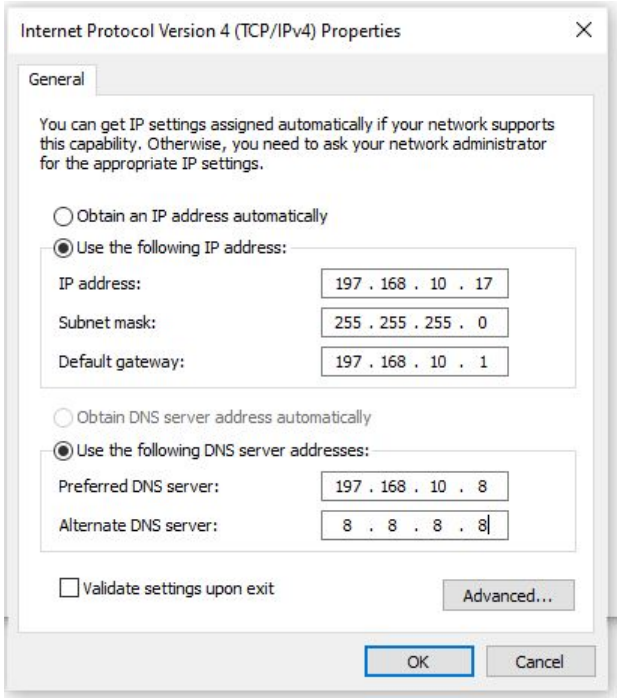
1. Q: How does a *host* get IP address within its network (host part of address)?
2. Q: How does a *network* get IP address for itself (network part of address)

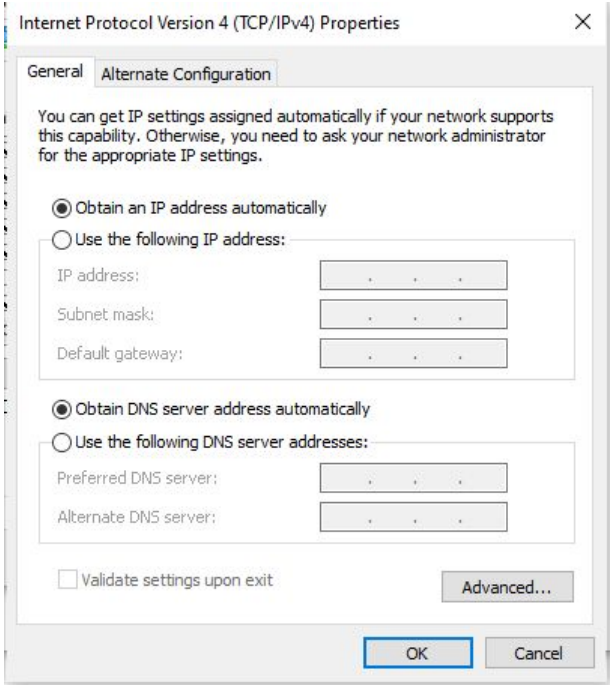
How does *host* get IP address?

- hard-coded by sysadmin in config file (e.g., /etc/rc.config in UNIX)
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - "plug-and-play"

DHCP: Dynamic Host Configuration Protocol

IP addresses: how to get one?





DHCP: Dynamic Host Configuration Protocol

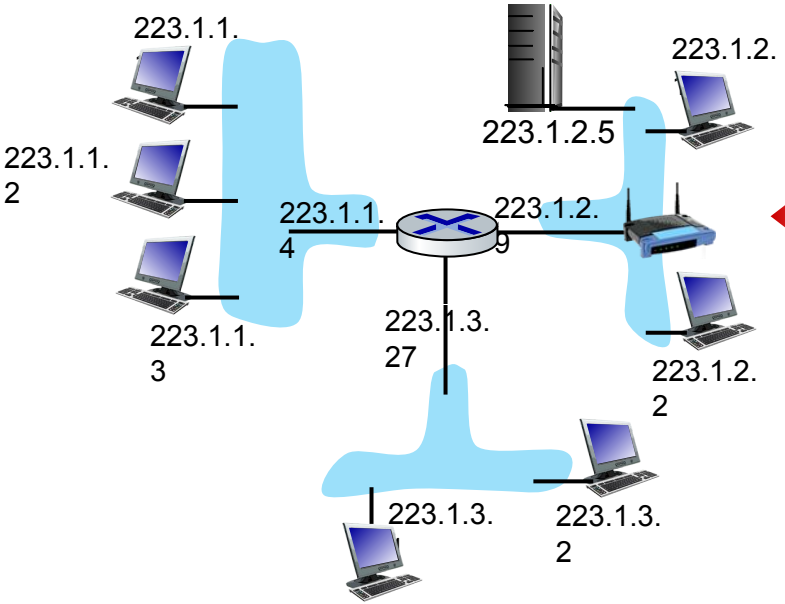
goal: host *dynamically* obtains IP address from network server when it "joins" network

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/on)
- support for mobile users who join/leave network

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

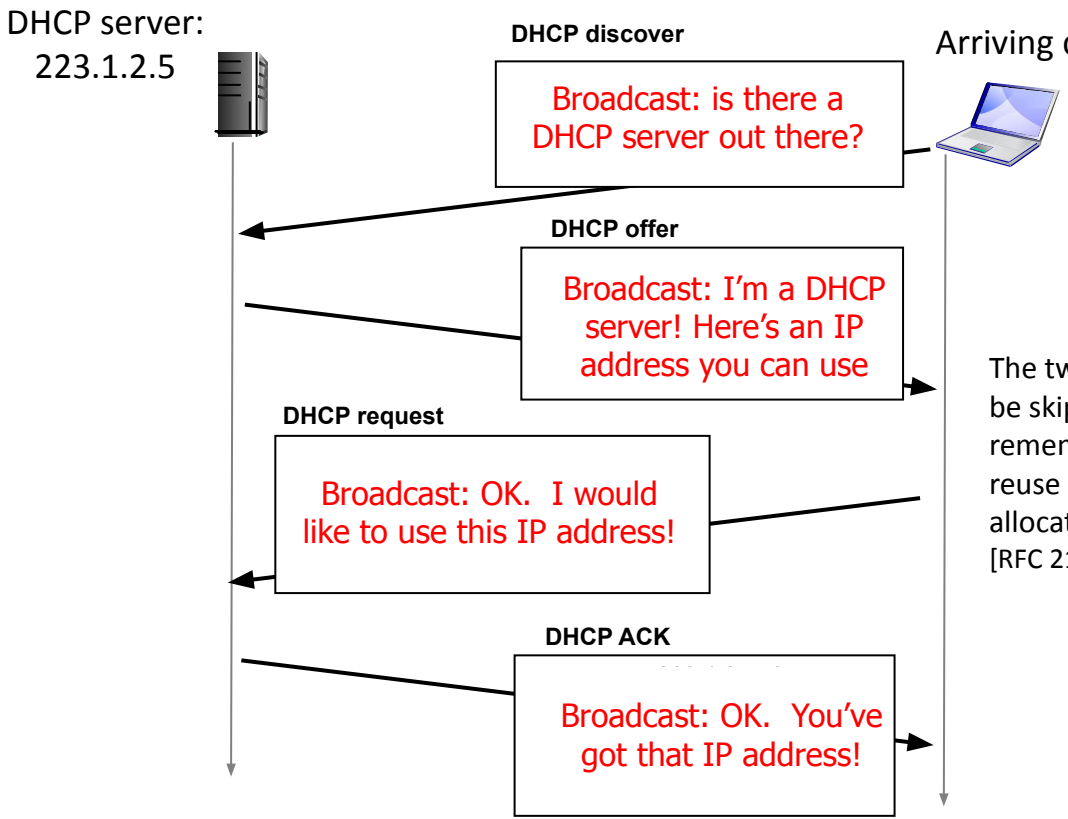
DHCP client-server scenario



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



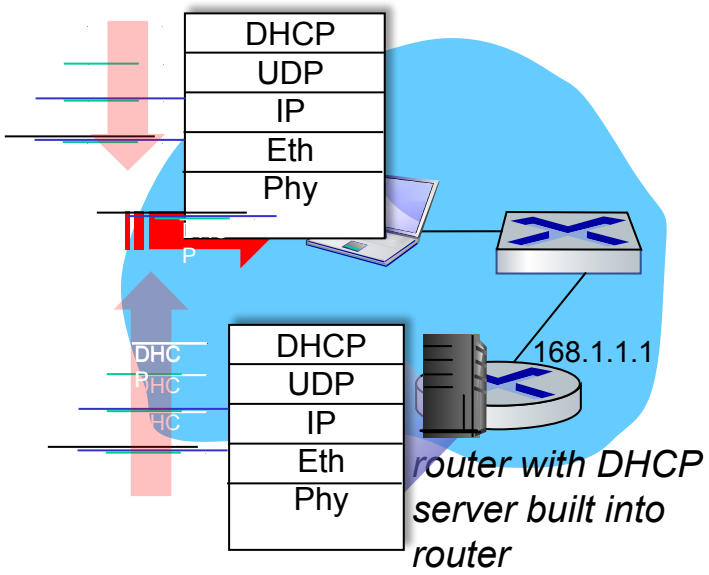
The two steps above can be skipped "if a client remembers and wishes to reuse a previously allocated network address" [RFC 2131]

DHCP: more than IP addresses

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

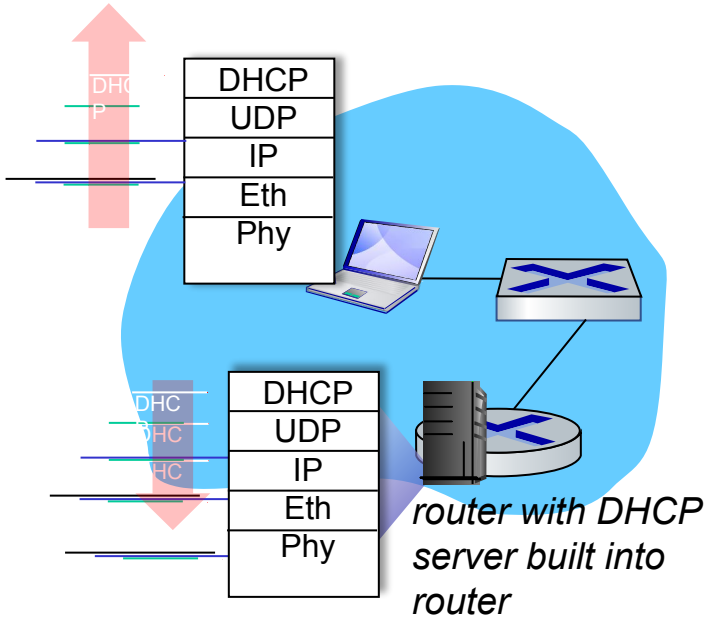
- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)

DHCP: example



- Connecting laptop will use DHCP to get **IP address, address of first-hop router, address of DNS server**.
- DHCP REQUEST message encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet de-mux'ed to IP de-mux'ed, UDP de-mux'ed to DHCP

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
- encapsulated DHCP server reply forwarded to client, de-muxing up to DHCP at client
- client now knows its IP address, name and IP address of DNS server, IP address of its first-hop router

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

Public Internet addresses are regulated by five Regional Internet Registries (RIRs):

- ARIN
- RIPE
- APNIC
- LACNIC
- AfriNIC

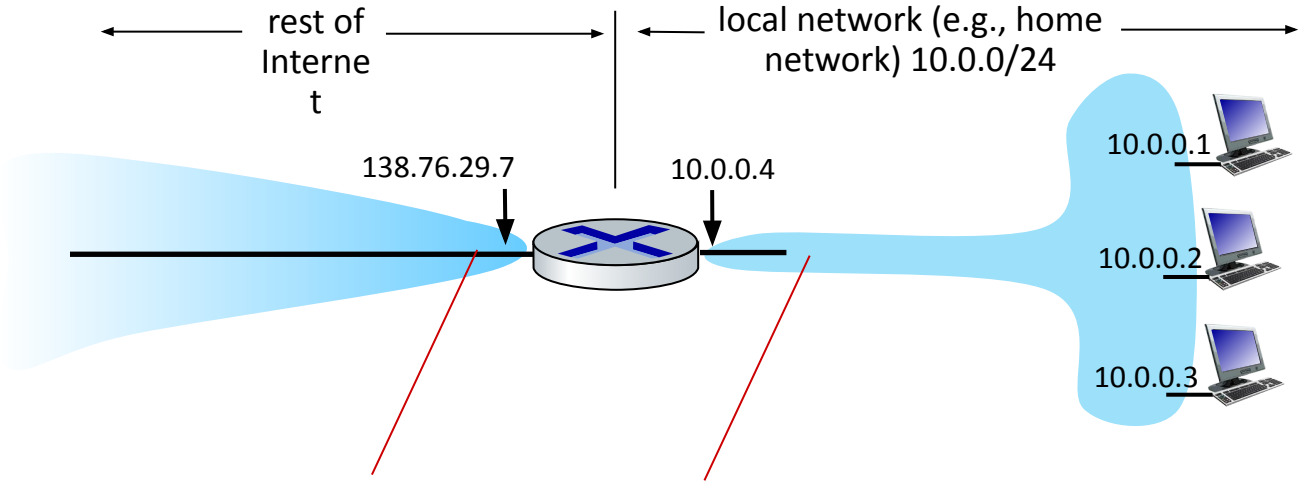
Network layer: “data plane” roadmap

- Network layer: overview
  - data plane
  - control plane
- What’s inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation

- Generalized Forwarding, SDN
  - match+action
  - OpenFlow: match+action in action
- Middleboxes

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)

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.
  - Generally, private networks use addresses from the following experimental address ranges (**Free and *non-routable addresses***): **[RFC 1918]**
    - 10.0.0.0 – 10.255.255.255
    - 172.16.0.0 – 172.31.255.255
    - 192.168.0.0 – 192.168.255.255
- 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

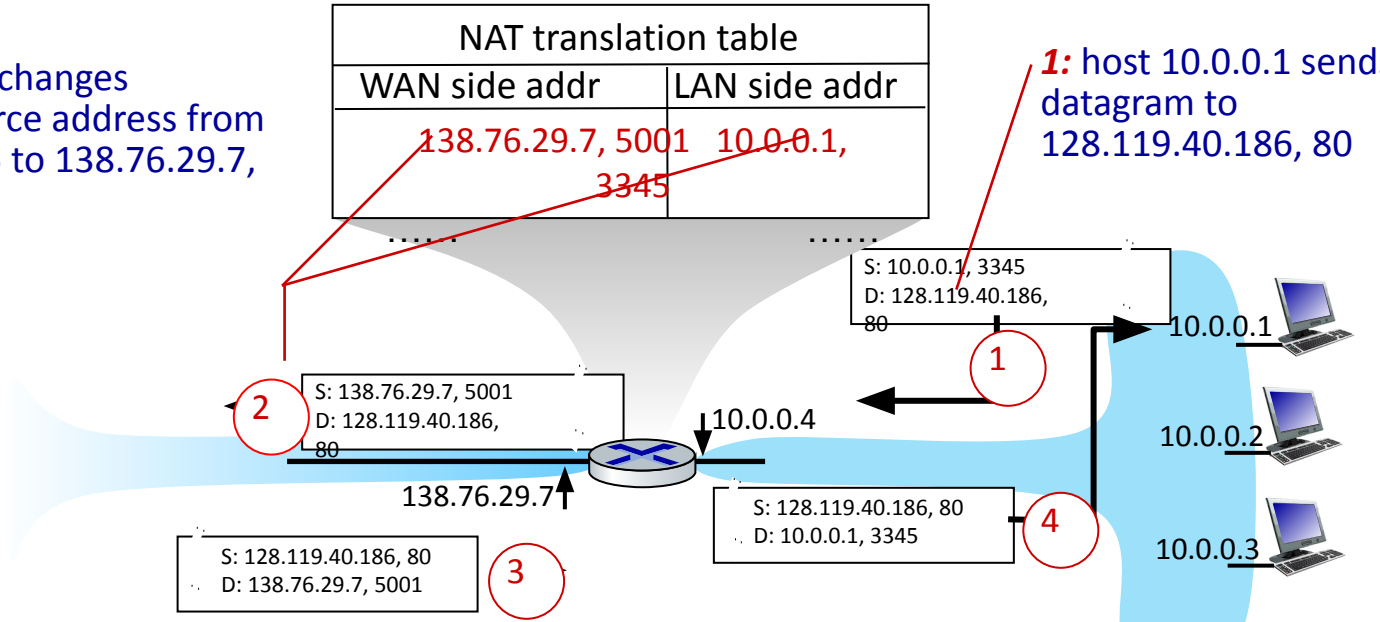
NAT: network address translation

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



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

- A short-term solution to the problem of the shortage of IP addresses
  - Long term solution is IP v6
  - CIDR (Classless Inter Domain Routing) is a possible short-term solution
  - NAT is another

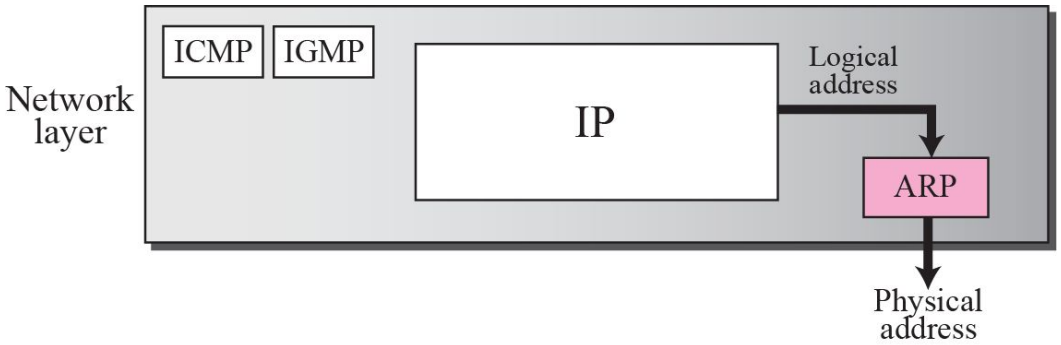
Concerns about NAT

- Performance:
  - Modifying the IP header by changing the IP address requires that NAT boxes recalculate the IP header checksum
  - Modifying port number requires that NAT boxes recalculate TCP checksum
- End-to-end connectivity:
  - NAT destroys universal end-to-end reachability of hosts on the Internet.
  - A host in the public Internet often cannot initiate communication to a host in a private network.
  - The problem is worse, when two hosts that are in a private network need to communicate with each other.
- but NAT is here to stay:
  - extensively used in home and institutional nets, 4G/5G cellular nets

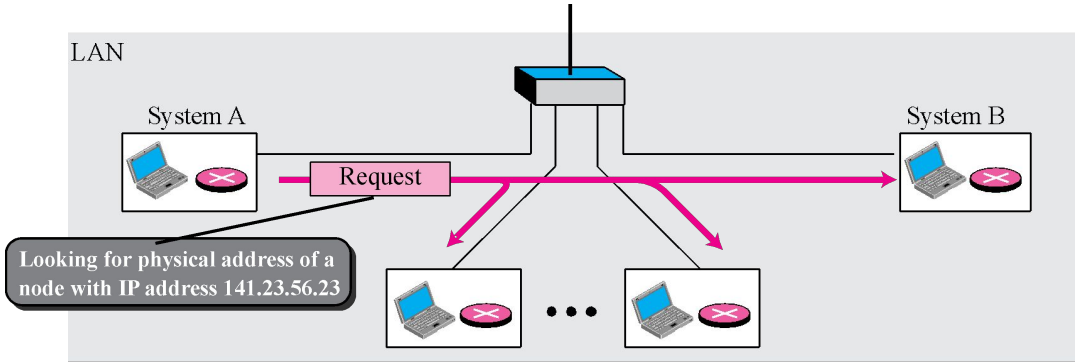
Address Resolution Protocol (ARP)

- **The delivery of a packet to a host or a router requires two levels of addressing: *logical* and *physical*.**
- **We need to be able to map a logical address to its corresponding physical address and vice versa.**
- **Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver.**
- **But the IP datagram must be encapsulated in a frame to be able to pass through the physical network.**
- **This means that the sender needs the physical address of the receiver.**
- **A mapping corresponds a logical address to a physical address. ARP accepts a logical address from the IP protocol, maps the address to the corresponding physical address and pass it to the data link layer.**

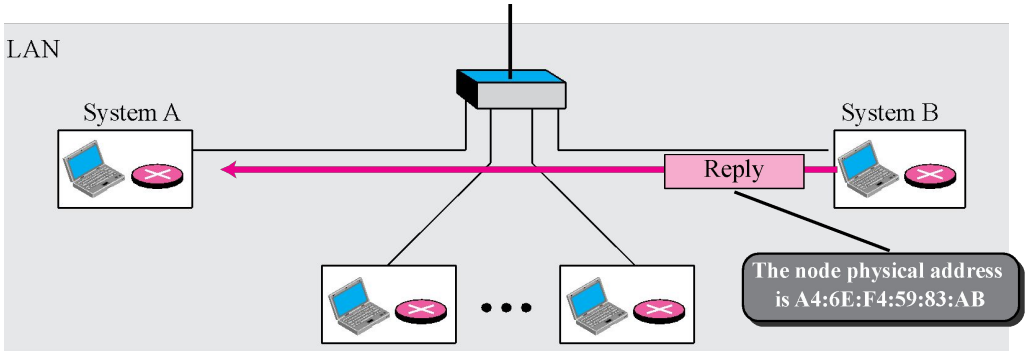
Position of ARP in TCP/IP protocol suite



ARP operation



a. ARP request is **broadcast**

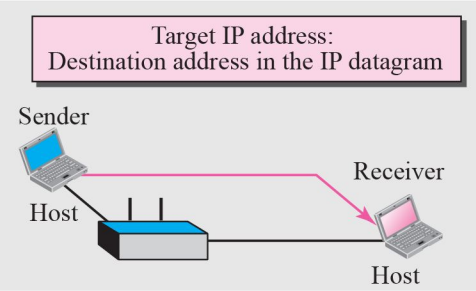


b. ARP reply is unicast

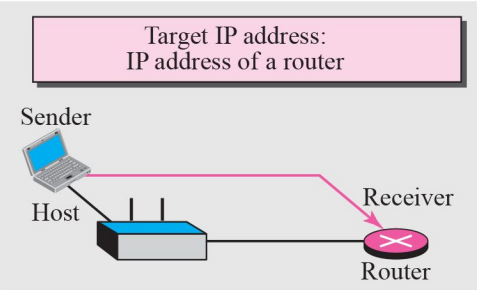
An ARP request is broadcast; an ARP reply is unicast.

Four cases using ARP

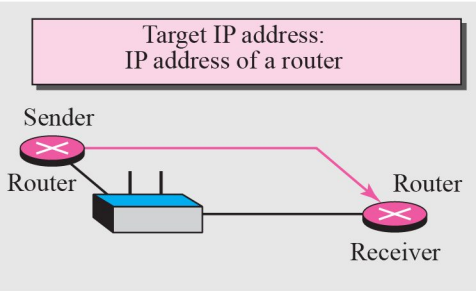
Case 1: A host has a packet to send to a host on the same network.



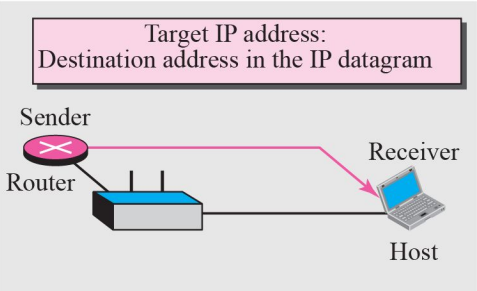
Case 2: A host has a packet to send to a host on another network.



Case 3: A router has a packet to send to a host on another network.

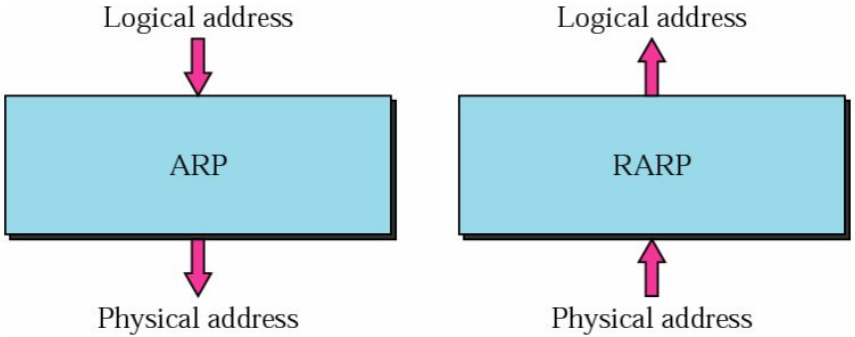


Case 4: A router has a packet to send to a host on the same network.

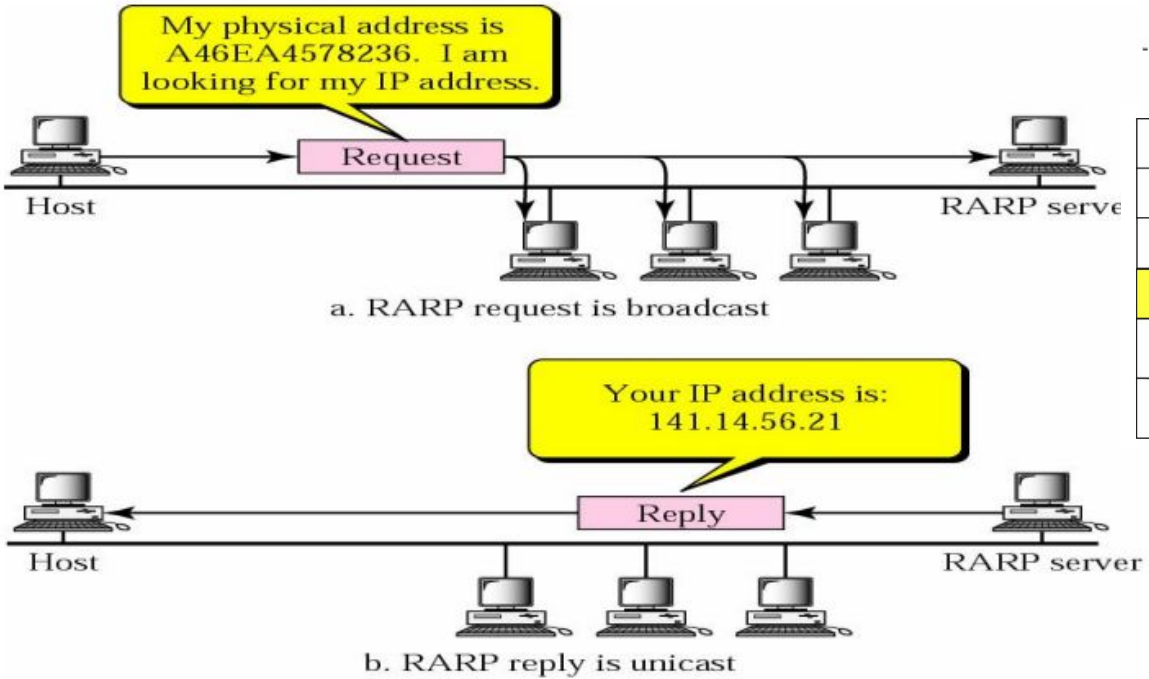


RARP

- Physical address to logical address.
- A diskless machine is usually booted from ROM. RARP is used for diskless machine which can not store the IP address.
- Request Broadcast, reply unicast by RARP server.



RARP Operation



Hardware type		Protocol type
Hardware length	Protocol length	Operation Request 3, Reply 4
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP) (It is not filled for request)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled for request)		
Target protocol address (For example, 4 bytes for IP) (It is not filled for request)		

The format of the RARP packet is the same as the ARP packet, Except that the operation field is three for RARP request message and Four for RARP reply message

Alternative Solution to RARP

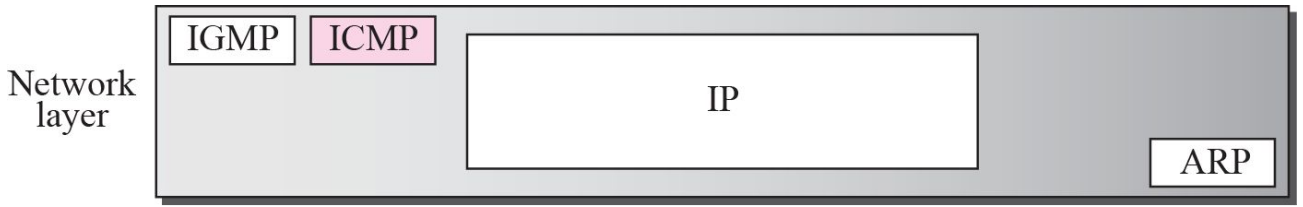
- When a diskless computer is booted, it needs more information in addition to its IP address
  - like subnet mask,
  - default gateway/router,
  - DNS server,
- RARP cannot provide this extra information
- Hence, we need something more than RARP i.e., DHCP.

Internet Control Message Protocol (ICMP)

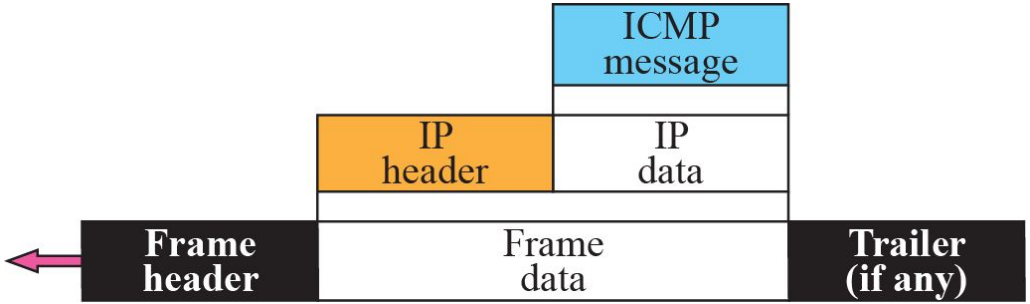
Introduction

- The IP protocol has no error-reporting or error correcting mechanism.
- What happens if something goes wrong? What happens if a router must discard a datagram because it cannot find a router to the final destination, or because the time-to-live field has a zero value?
- These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.

Position of ICMP in the network layer



ICMP encapsulation



ICMP

- ICMP messages are divided into two broad categories:
  - error-reporting messages
  - query messages.

- The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.

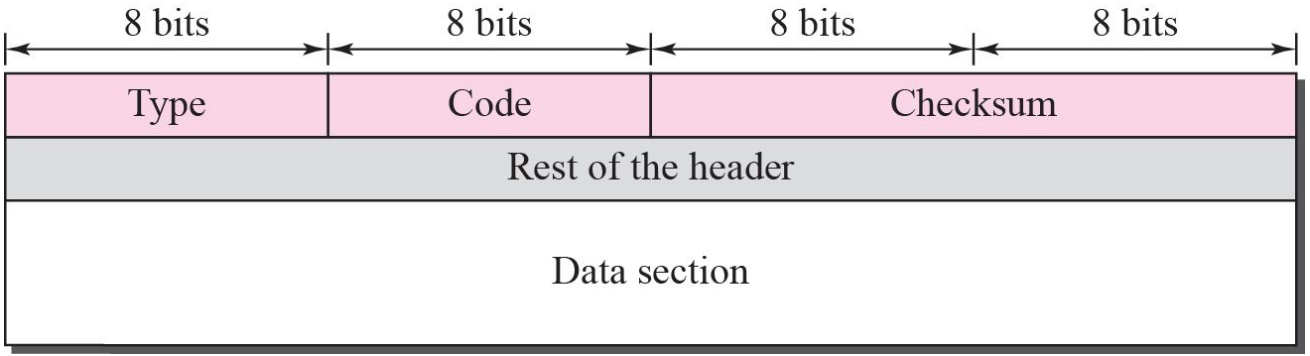
- The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host.

Table 9.1 *ICMP messages*

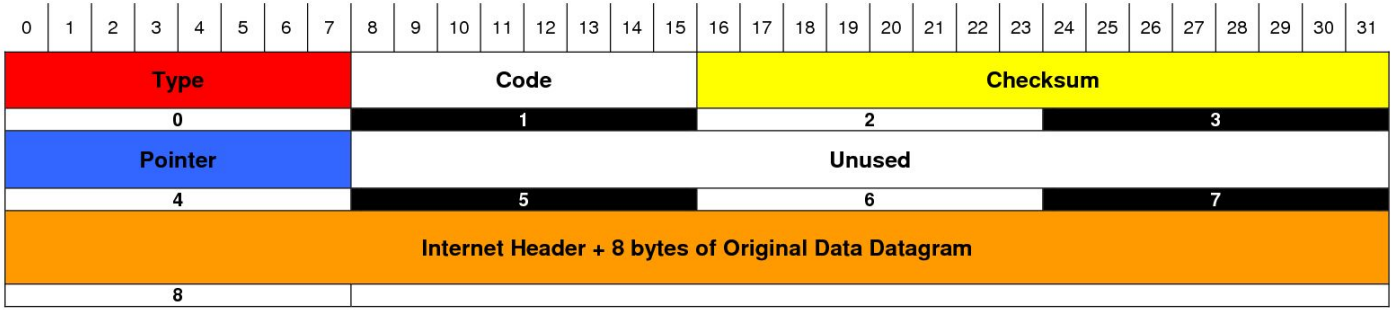
Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml

General format of ICMP messages



ICMP Parameter Message Format

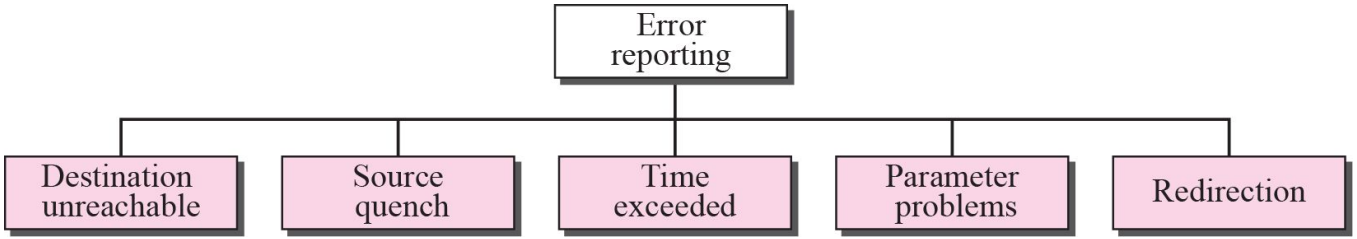


Type	Code	Meaning
0	0	Echo Reply
3	0	Net Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Port Unreachable
	4	Frag needed and DF set
	5	Source route failed
	6	Dest network unknown
	7	Dest host unknown
	8	Source host isolated
	9	Network admin prohibited
	10	Host admin prohibited
	11	Network unreachable for TOS
	12	Host unreachable for TOS
	13	Communication admin prohibited
4	0	Source Quench (Slow down/Shut up)

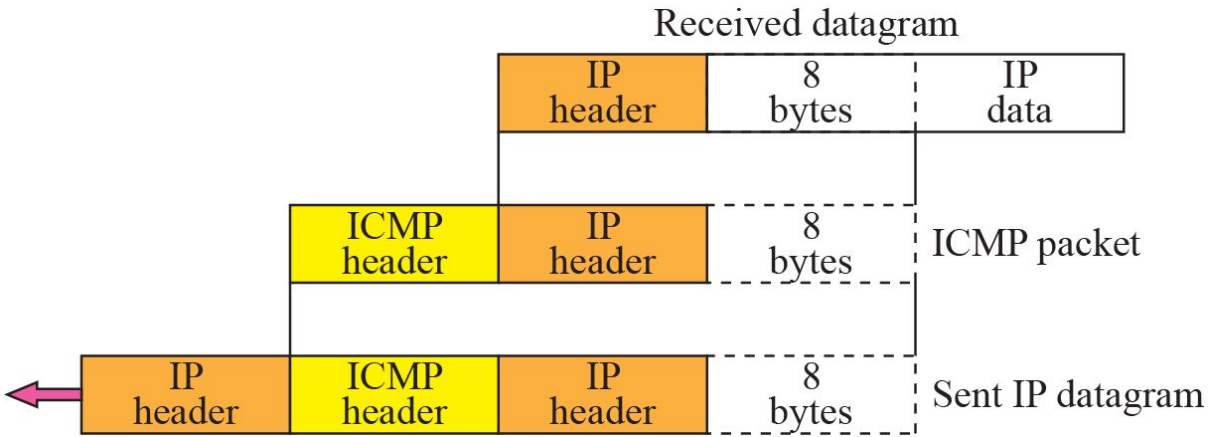
Type	Code	Meaning
5	0	Redirect datagram for the network
	1	Redirect datagram for the host
	2	Redirect datagram for the TOS & Network
	3	Redirect datagram for the TOS & Host
8	0	Echo
9	0	Router advertisement
10	0	Router selection
11	0	Time To Live exceeded in transit
	1	Fragment reassemble time exceeded
12	0	Pointer indicates the error (Parameter Problem)
	1	Missing a required option (Parameter Problem)
	2	Bad length (Parameter Problem)
13	0	Time Stamp
14	0	Time Stamp Reply
15	0	Information Request
16	0	Informaiton Reply
17	0	Address Mask Request
18	0	Address Mask Reply
30	0	Traceroute (Tracert)

ICMP always reports error messages to the original source.

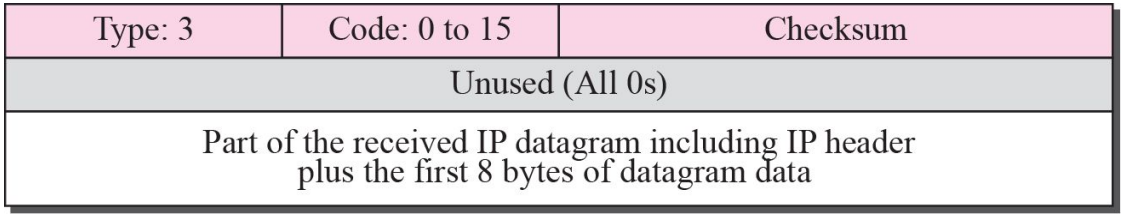
Error-reporting messages



Contents of data field for the error message



Destination-unreachable format

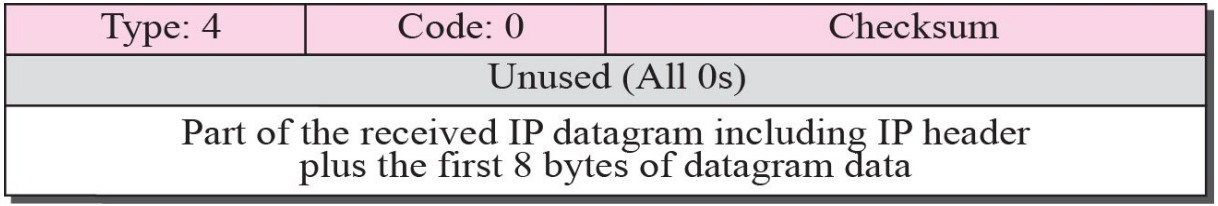


Note

Destination-unreachable messages with codes 2 or 3 can be created only by the destination host.

Other destination-unreachable messages can be created only by routers.

Source-quench format



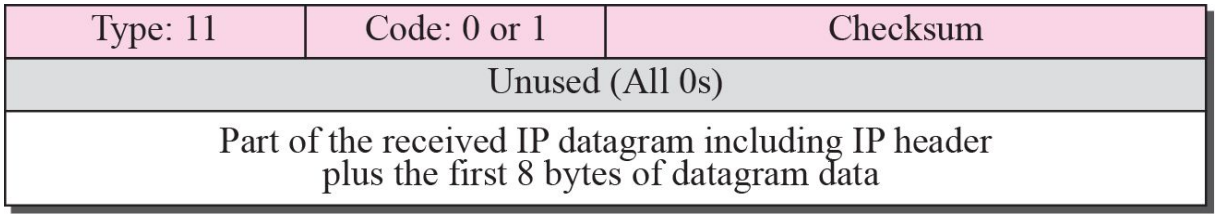
Note

A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host.

The source must slow down the sending of datagrams until the congestion is relieved.

One source-quench message is sent for each datagram that is discarded due to congestion.

Time-exceeded message format



Note

Whenever a router decrements a datagram with a time-to-live value to zero, it discards the datagram and sends a time-exceeded message to the original source.

Note

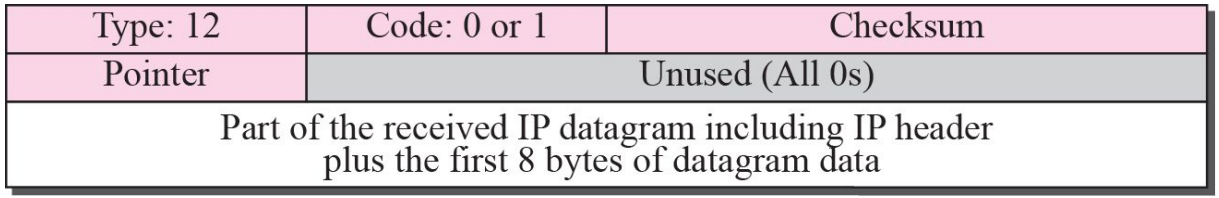
When the final destination does not receive all of the fragments in a set time, it discards the received fragments and sends a time-exceeded message to the original source.

Note

In a time-exceeded message, code 0 is used only by routers to show that the value of the time-to-live field is zero.

Code 1 is used only by the destination host to show that not all of the fragments have arrived within a set time.

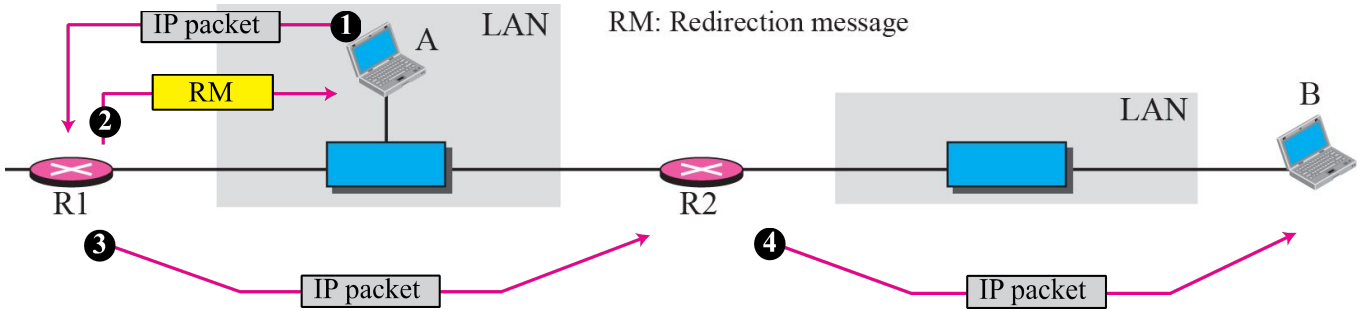
Parameter-problem message format



Code Value	Message Subtype	Description
0	*Pointer Indicates The Error*	This is the normal use of the *Parameter Problem* message; when this *Code* value is used, the *Pointer* field indicates the location of the problem.
1	*Missing A Required Option*	The IP datagram needed to have an option in it that was missing. Since the option was missing there is no way to point to it, of course. ☺
2	*Bad Length*	The length of the datagram overall was incorrect, indicating a general problem with the message as a whole. Again, the *Pointer* field makes no sense here.

A parameter-problem message can be created by a router or the destination host.

Redirection concept

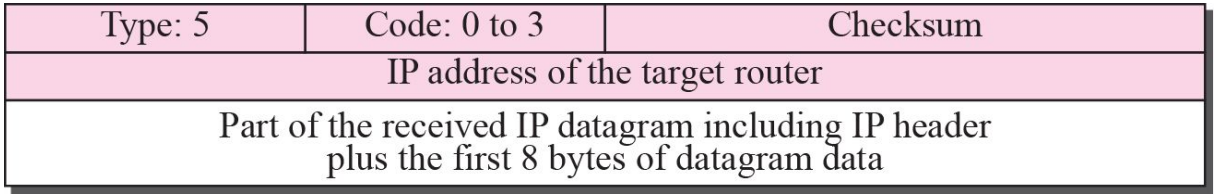


Note

A host usually starts with a small routing table that is gradually augmented and updated.

One of the tools to accomplish this is the redirection message.

Redirection message format



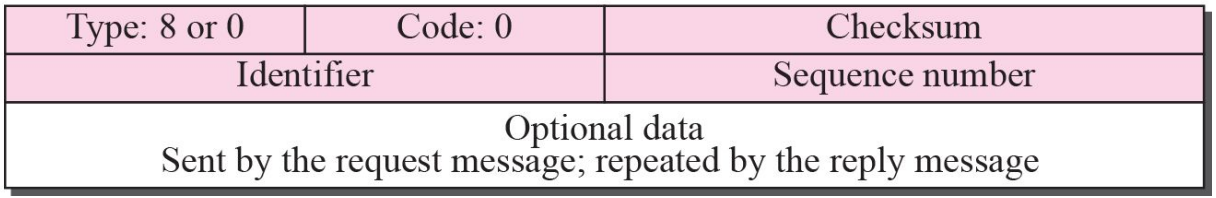
ICMP Code	Meaning
0	Redirect IPv4 datagram for the network (or subnet)
1	Redirect IPv4 datagram for the host
2	Redirect IPv4 datagram for Type of Service and the network
3	Redirect IPv4 datagram for Type of Service and the host

Note

A redirection message is sent from a router to a host on the same local network.

Echo-request and echo-reply message

Type 8: Echo request
Type 0: Echo reply



Note

An echo-request message can be sent by a host or router.

An echo-reply message is sent by the host or router that receives an echo-request message.

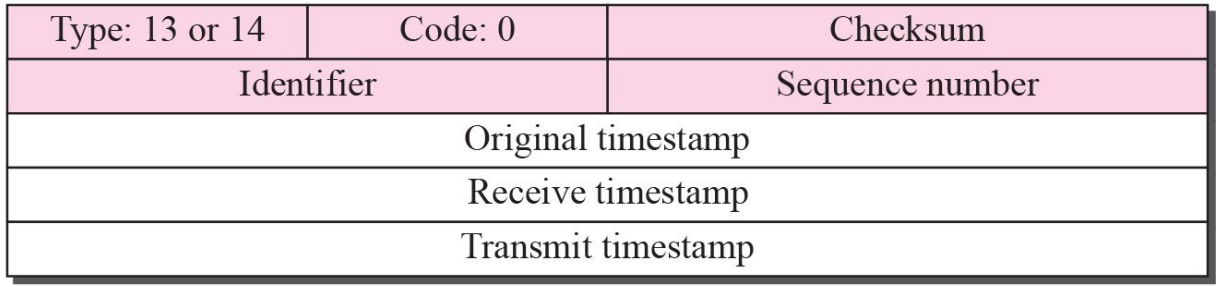
Note

Echo-request and echo-reply messages can test the reachability of a host.

This is usually done by invoking the ping command.

Timestamp-request and timestamp-reply message format

Type 13: request
Type 14: reply



Note

Timestamp-request and timestamp-reply messages can be used to calculate the round-trip time between a source and a destination machine even if their clocks are not synchronized.

Note

The timestamp-request and timestamp-reply messages can be used to synchronize two clocks in two machines if the exact one-way time duration is known.

Problems

Problem:

Consider a router that interconnects three subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix 223.1.17/24. Also suppose that Subnet 1 is required to support at least 62 interfaces, Subnet 2 is to support at least 106 interfaces, and Subnet 3 is to support at least 15 interfaces. Provide three network addresses (of the form a.b.c.d/x) that satisfy these constraints.

Step 1 1 of 4

For *Subnet 1* 62 interface is required. The nearest range that includes 62 addresses is **64**. When we translate 64 to bits it is **6 bits**. We give the last 6 bits to hosts from 223.1.17/24. That's why, the network address for *Subnet 1* will be **223.1.17.0/26**. And the range will be **223.1.17.0-223.1.17.63**.

Step 2 2 of 4

For *Subnet 2* 106 interface is required. The nearest range that includes 106 addresses is **128**. When we translate 128 to bits it is **7 bits**. We give the last 7 bits to hosts from 223.1.17/24. That's why, the network address for *Subnet 2* will be **223.1.17.128/25**. And the range will be **223.1.17.128-223.1.17.255**.

Step 3 3 of 4

For *Subnet 3* 15 interface is required. The nearest range that includes 15 addresses is **16**. When we translate 16 to bits it is **4 bits**. We give the last 4 bits to hosts from 223.1.17/24. Because of *Subnet 1* and *Subnet 2* taking wide range from 223.1.17/24, There are left range from **223.1.17.64-223.1.17.127**. That's why we will take from 223.1.17.64/26. The network address for *Subnet 3* will be **223.1.17.64/28**. And the range will be **223.1.17.64-223.1.17.79**.

Network address for *Subnet 1* is **223.1.17.0/26**.

Network address for *Subnet 2* is **223.1.17.128/25**.

Network address for *Subnet 3* is **223.1.17.64/28**.

Problem:

Consider a subnet with prefix 192.168.56.128/26. Give an example of one IP address (of form xxx.xxx.xxx.xxx) that can be assigned to this network.

Suppose an ISP owns the block of addresses of the form 192.168.56.32/26. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

Step 1 1 of 3

If prefix of a subnet is 192.168.56.128/26, the address range is **192.168.56.128-192.168.56.191**. For example, **192.168.56.155** is assigned to this network.

When we calculate the range for 192.168.56.32/26, it will be **192.168.56.0-192.168.56.63** Here is **64** address. When we want to create 4 subnets with the same number of IP addresses, we should divide 64 addresses to 4 parts. 64/4=16. Each subnet will have **16 addresses**. It means we should give last 4 bits to hosts and 2 bits will be subnet mask.

Step 2 2 of 3

That's why, each subnet network address will be like this:

First subnet: **192.168.56.0/28 (192.168.56.0-192.168.56.15)**

Second subnet: **192.168.56.16/28 (192.168.56.16-192.168.56.31)**

Third subnet: **192.168.56.32/28 (192.168.56.32-192.168.56.47)**

Fourth subnet: **192.168.56.48/28 (192.168.56.48-192.168.56.63)**

a) **192.168.56.155**

b) First subnet: **192.168.56.0/28**

Second subnet: **192.168.56.16/28**

Third subnet: **192.168.56.32/28**

Fourth subnet: **192.168.56.48/28**

P17. Consider the topology shown in Figure 4.17. Denote the three subnets with hosts (starting clockwise at 12:00) as Networks A, B, and C. Denote the subnets without hosts as Networks D, E, and F.

a. Assign network addresses to each of these six subnets, with the following constraints: All addresses must be allocated from 214.97.254/23; Subnet A should have enough addresses to support 250 interfaces; Subnet B should have enough addresses to support 120 interfaces; and Subnet C should have enough addresses to support 120 interfaces. Of course, subnets D, E and F should each be able to support two interfaces. For each subnet, the assignment should take the form a.b.c.d/x or a.b.c.d/x – e.f.g.h/y.

b. Using your answer to part (a), provide the forwarding tables (using longest prefix matching) for each of the three routers.

Step 1 1 of 5

a) For assigning 250 interfaces for $Subnet\ A$ we should take **8 bits** for it from 214.97.254/23. The network address for $Subnet\ A$ is $\mathbf{214.97.254.0/24}$. The network range is **214.97.254.0-214.97.254.255**.

The nearest range for $Subnet\ B$ is **128** and it is **7 bits**. The network address of $Subnet\ B$ is $\mathbf{214.97.255.0/25}$. And the range is **214.97.255.0-214.97.255.127**.

Step 2 2 of 5

It is the same situation for $Subnet\ C$ as $Subnet\ B$. And the network address for $Subnet\ C$ is $\mathbf{214.97.255.128/25}$. The network range is **214.97.255.128-214.97.255.255**.

For taking two interfaces for each of $Subnet\ D$, $Subnet\ E$ and $Subnet\ F$, only **1 bit** is assigned as host bit for each of them. That's why, their network address will be like this: network address for $Subnet\ D$ is $\mathbf{214.97.255.0/31}$, for $Subnet\ E$ is $\mathbf{214.97.255.2/31}$ and for $Subnet\ F$ is $\mathbf{214.97.255.4/31}$.

Step 3

b) Before creating forwarding tables for router, we can give *Subnet D* to the link between **R1** and **R2**, *Subnet E* to the link between **R1** and **R3** and *Subnet F* to the link between **R2** and **R3**. I also give *Subnet A* to the hosts of **R1**, *Subnet B* to the hosts of **R2** and *Subnet C* to the hosts of **R3**. The forwarding tables will be like this:

Forwarding table of R1	
Prefix	Interface
11010110 01100001 11111110	Interface for Subnet A
11010110 01100001 11111111 00000000	Interface for Subnet D
11010110 01100001 11111111 0000001	Interface for Subnet E

Step 4

Forwarding table of R2	
Prefix	Interface
11010110 01100001 11111111 0	Interface for Subnet B
11010110 01100001 11111111 0000000	Interface for Subnet D
11010110 01100001 11111111 000001	Interface for Subnet F

Step 5

Forwarding table of R3	
Prefix	Interface
11010110 01100001 11111111 1	Interface for Subnet C
11010110 01100001 11111111 0000001	Interface for Subnet E
11010110 01100001 11111111 000001	Interface for Subnet F

Problem:

Consider sending a 2400-byte datagram into a link that has an MTU of 700 bytes. Suppose the original datagram is stamped with the identification number 422. How many fragments are generated? What are the values in the various fields in the IP datagram(s) generated related to fragmentation?

Step 1 1 of 4

Firstly, we can see that **MTU** is 700 bytes. It means that we can use maximum 700 bytes for each fragmentation including header. We have 2400-byte datagram. In the first stage we should subtract **IP header** from it. IP header is 20 bytes. 2400-20=2380. Now he have **2380-byte** datagram without header. Our MTU is 700 bytes but we should also substract 20 from it. Because in every fragment 20 bytes will be for IP header. 700-20=**680**. Then, we should divide this datagram with MTU for finding the number of fragments. 2380/680=3.5. It means that there are **4** fragments.

Step 2 2 of 4

The various field in the IP datagrams generated related to fragmentation are **Fragment**, **Bytes**, **ID**, **Offset** and **Flag**. Fragment will be **4**. First of three fragments will be **700 bytes** but the last one will be **360 bytes** (2380-680*3=340+20(IP header)). ID or identification number will be **422** for each fragment. Offset determines the beginning byte for each fragment. But we should divide each beginning byte with 8 for adding to offset. And offset for each fragment will be **0**, **85 (680/8=85)**, **170 (1360/8=170)**, **255 (2040/8=255)**. Finally, flag number determines if there is another fragment after it or not. That's why, for the first there fragment it will be **1**, but for the last one will be **0**.

Step 3 3 of 4

Fragment	Bytes (with IP header)	ID	Offset	Flag
1st fragment	700 bytes	422	0	1
2nd fragment	700 bytes	422	85	1
3rd fragment	700 bytes	422	170	1
4th fragment	360 bytes	422	255	0

Result 4 of 4

a) **4** fragments are generated.

b)

Fragment	Bytes (with IP header)	ID	Offset	Flag
1st fragment	700 bytes	422	0	1
2nd fragment	700 bytes	422	85	1
3rd fragment	700 bytes	422	170	1
4th fragment	360 bytes	422	255	0

Problem: Suppose datagrams are limited to 1,500 bytes (including header) between source Host A and destination Host B. Assuming a 20-byte IP header, how many datagrams would be required to send an MP3 consisting of 5 million bytes? Explain how you computed your answer.

When one datagram is transferred in TCP/IP environment, it takes **20-byte TCP header** and **20-byte IP header**. That's why, we should subtract that size of header from datagram size limit. 1500-40=1460. It is **1460** now. Then, we divide size of our MP3 file with that limited datagram size for finding the number of datagrams for transmission form *Host A* to *Host B*. 5000000/1460=3424.66. We need **3425 datagrams** to transmission of 5 million-byte MP3 file from *Host A* to *Host B*.