Network Layer: IP Protocol

A/PROF. DUY NGO

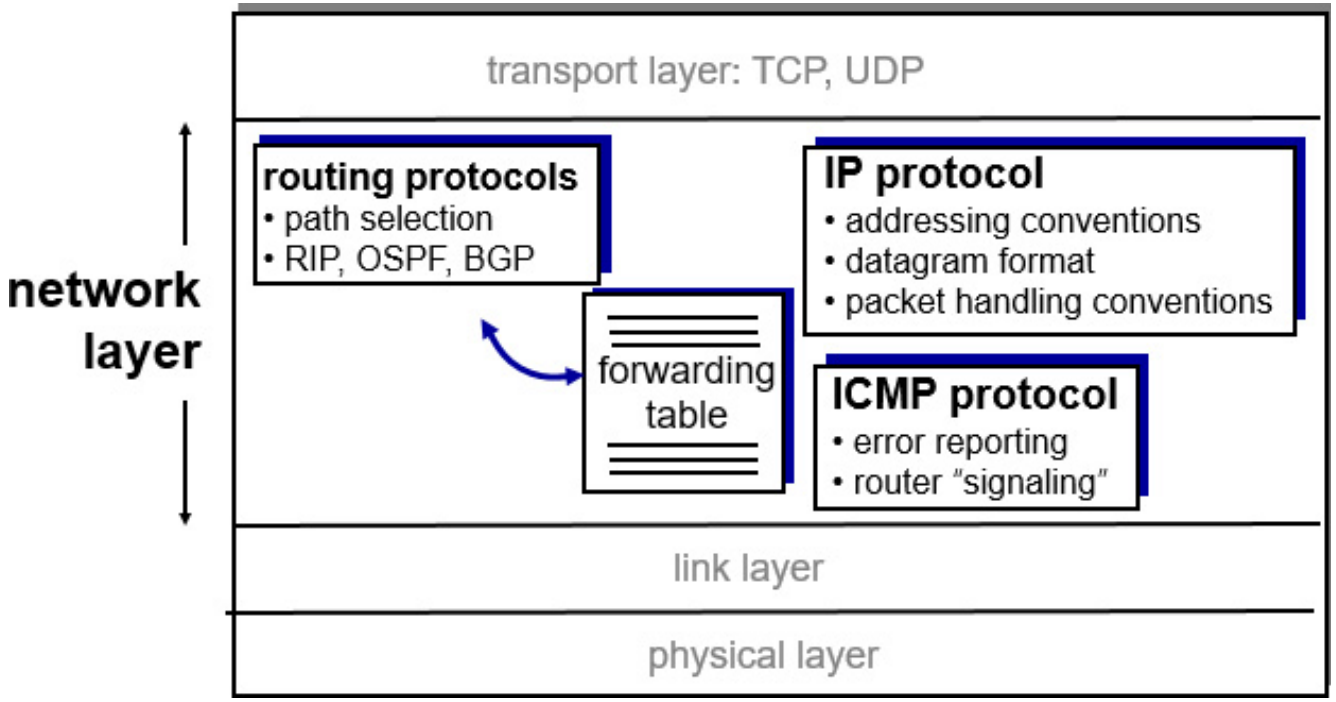
Learning Objectives

4.3 IP: Internet Protocol

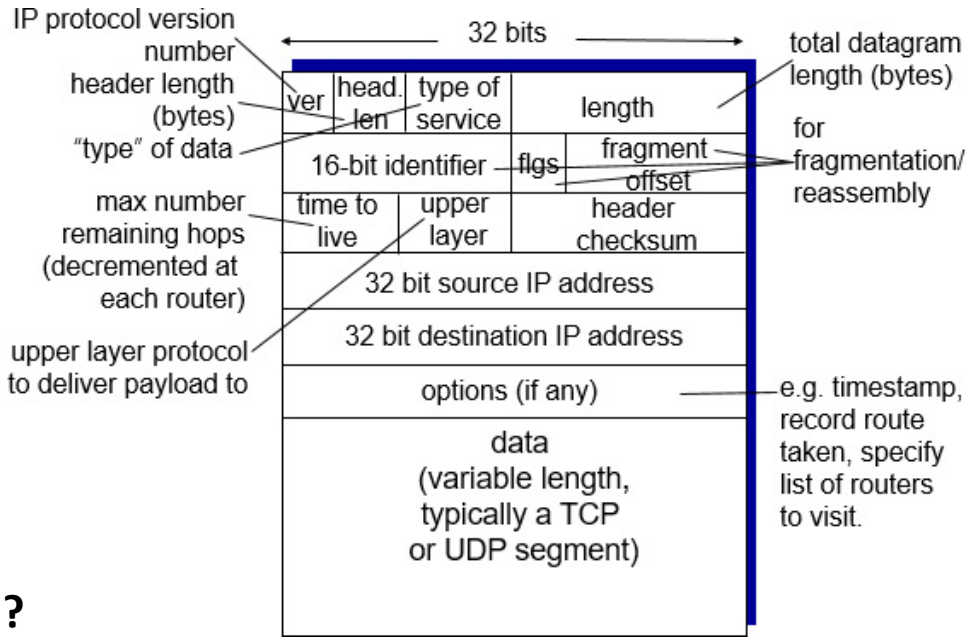
- **datagram format**
- **fragmentation**
- **IPv4 addressing**
- **network address translation**

The Internet Network Layer

host, router network layer functions:



IP Datagram Format



how much overhead?

20 bytes of TCP

20 bytes of IP

= 40 bytes + app layer overhead

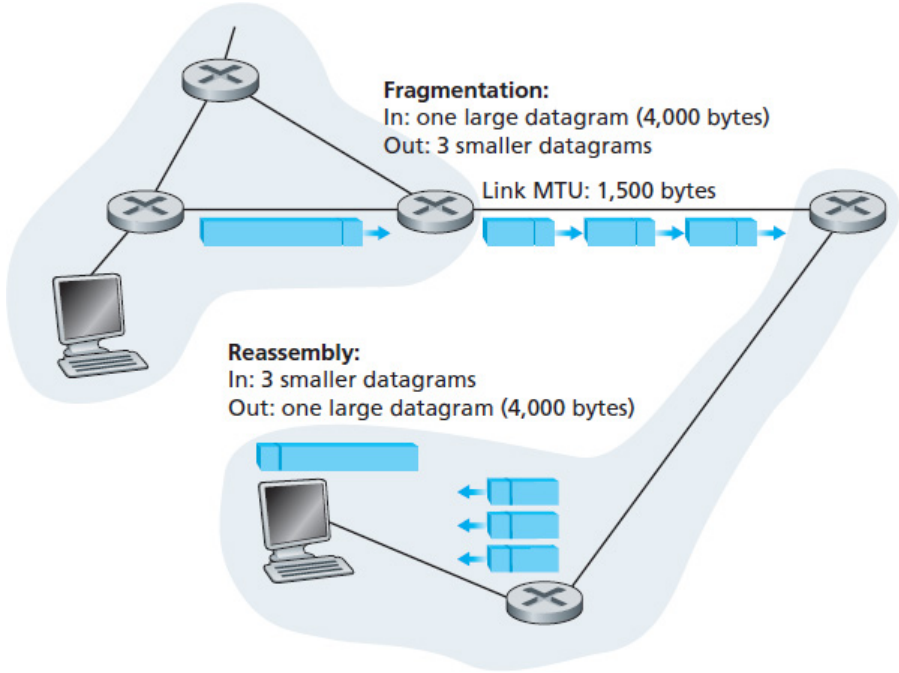
IP Fragmentation, Reassembly

network links have MTU (max.transfer size) - largest possible link-level frame

- different link types, different MTUs

large IP datagram divided ("fragmented") within net

- one datagram becomes several datagrams
- "reassembled" only at final destination
- IP header bits used to identify, order related fragments

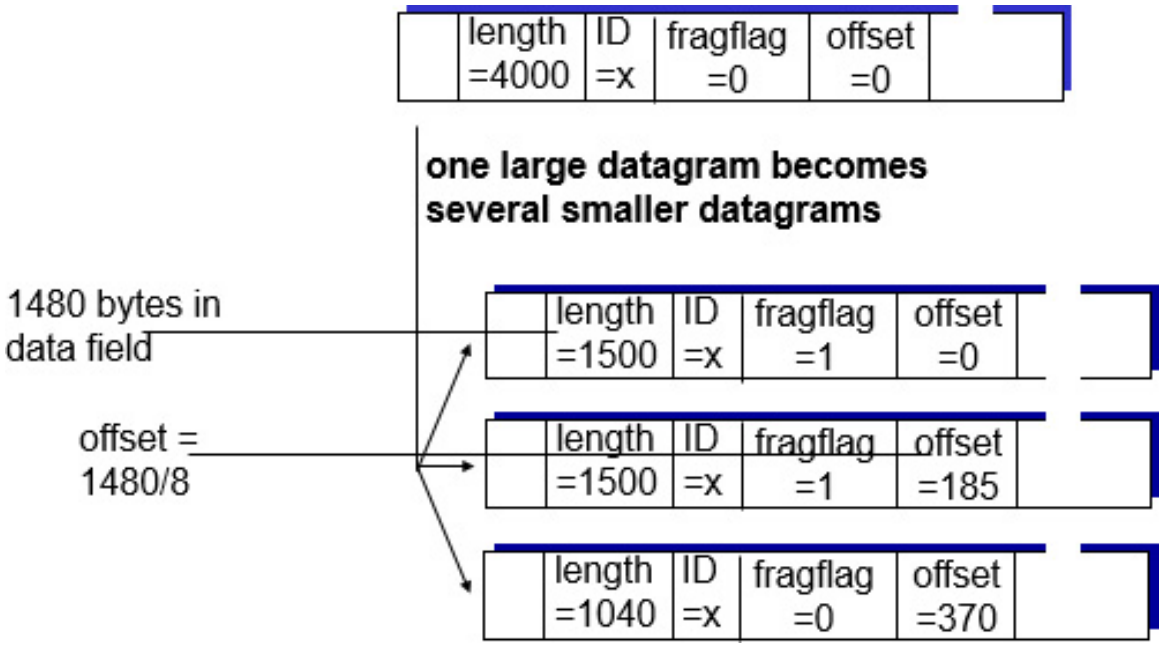


IP Fragmentation, Reassembly

example:

4000 byte datagram

MTU = 1500 bytes



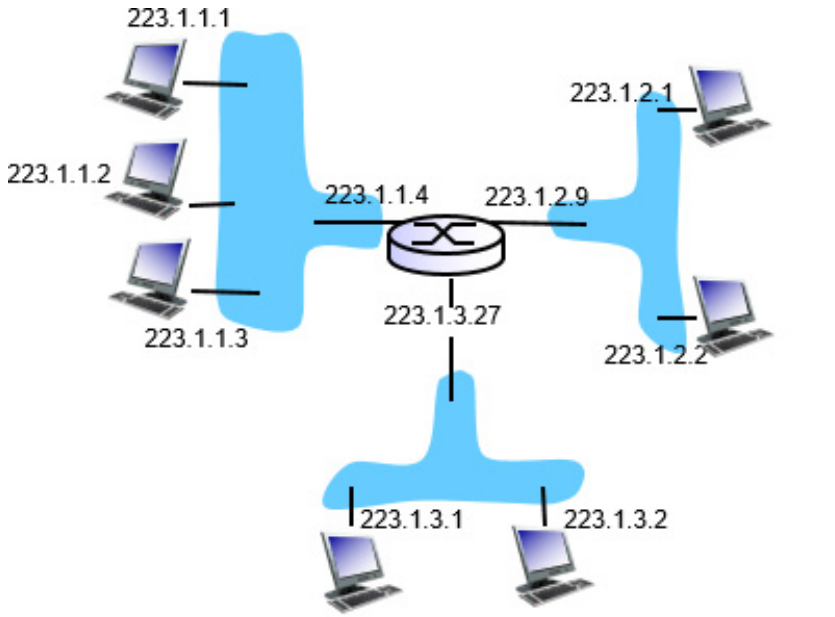
IP Addressing: Introduction (1 of 2)

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)

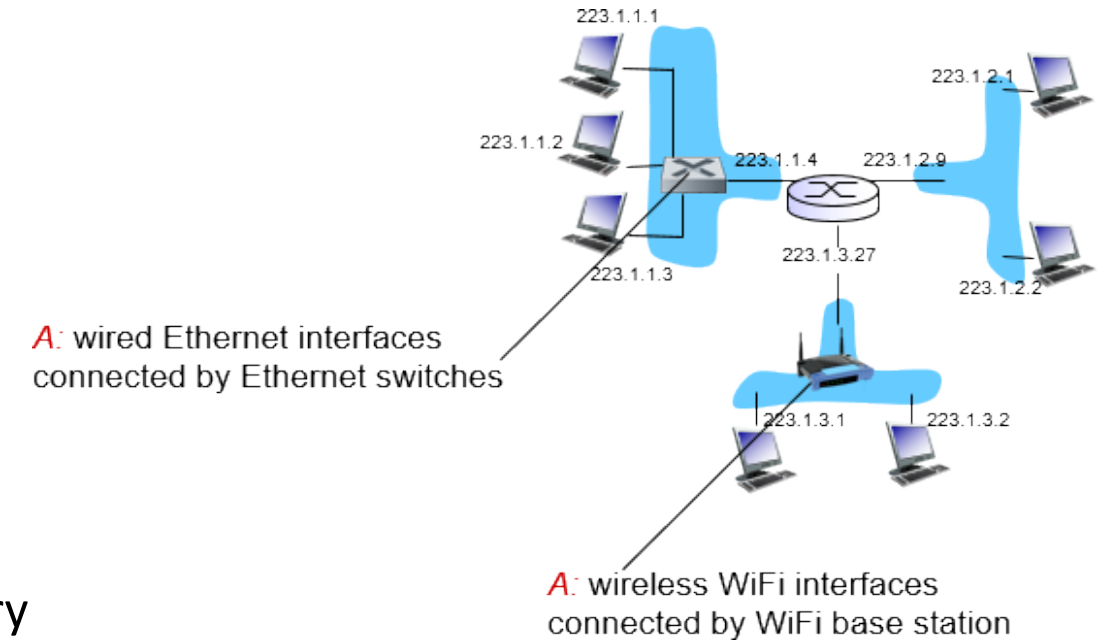
IP addresses associated with each interface



IP Addressing: Introduction (2 of 2)

Q: how are interfaces actually connected?

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



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

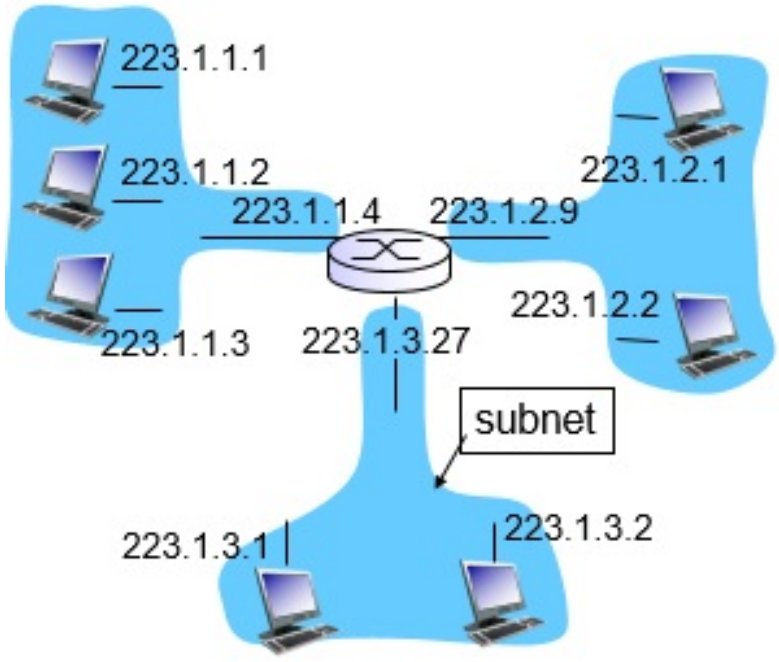
Subnets (1 of 3)

IP address:

- subnet part - high order bits
- host part - low order bits

What's subnet ?

- device interfaces with same subnet part of IP address
- can physically reach each other **without intervening router**



network consisting of 3 subnets

IP Addressing

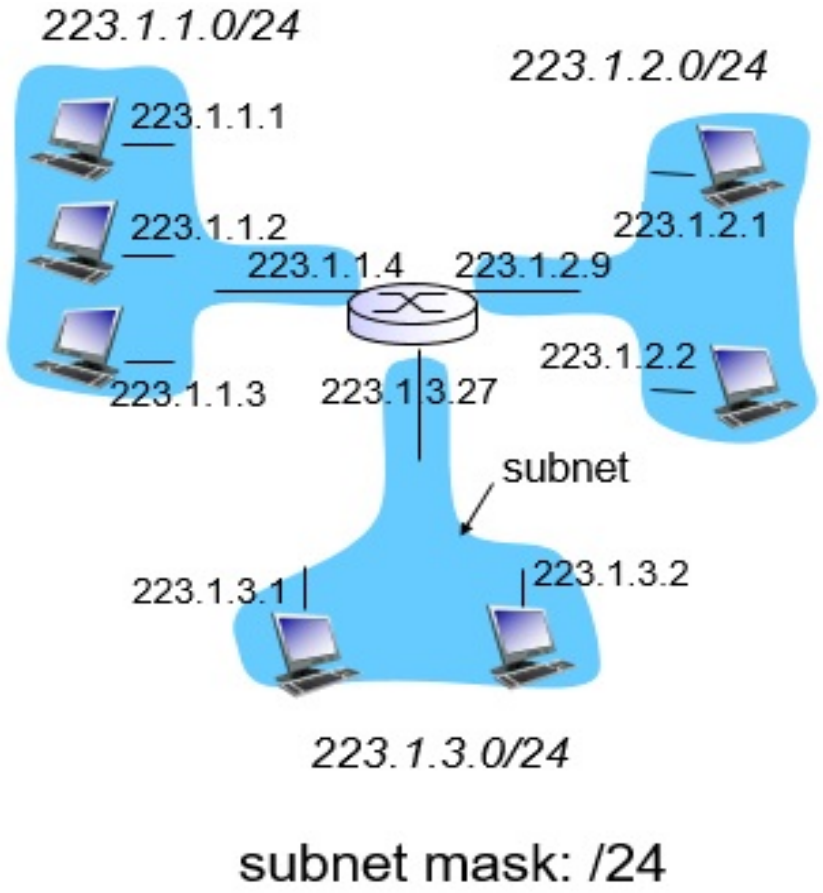
- Here we will use the IPv4 addresses.
- The IP header is 32 bit long, used as a unique identifier to locate a device on the IP network
- To make networks scalable, the address structure is subdivided into the fields: the **Network(Subnet) ID** and the **Host ID**
- The **Network/subnet ID** identifies a network where the device with the IP address is connected
- The **Host ID** identifies the device within the networl
- The IP address is subdivided into five classes as below:
  - Class A: fewer networks, but large networks, 7 bit Net/subnet ID, 24 bit Host ID
  - Class B: more networks than class A, moderate number of hosts/network, 14 bit Net/subnet ID, 16 bit Host ID
  - Class C: Very large no. of networks, few hosts/network, 21 Net/subnet ID, 8 bit Host ID
  - Class D: Multicast addressing
  - Class E: Reserved for experiments

Subnets (2 of 3)

recipe

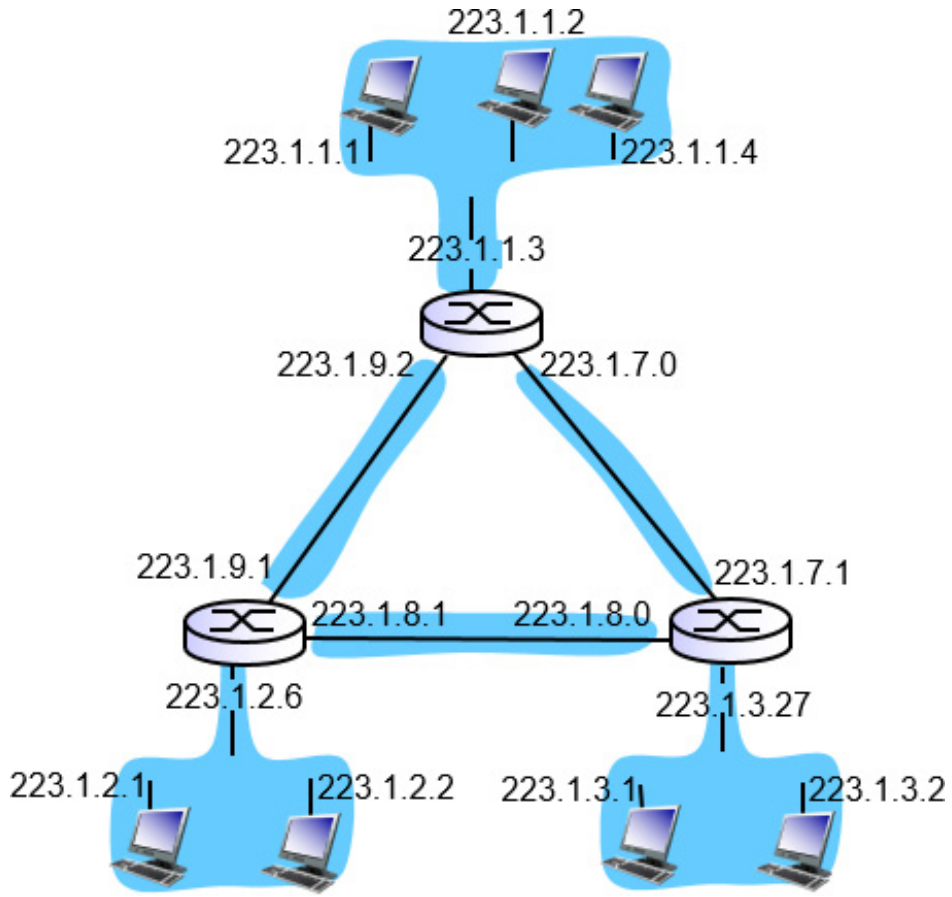
to determine the subnets, detach each interface from its host or router, creating islands of isolated networks

each isolated network is called a **subnet**



Subnets (3 of 3)

how many?



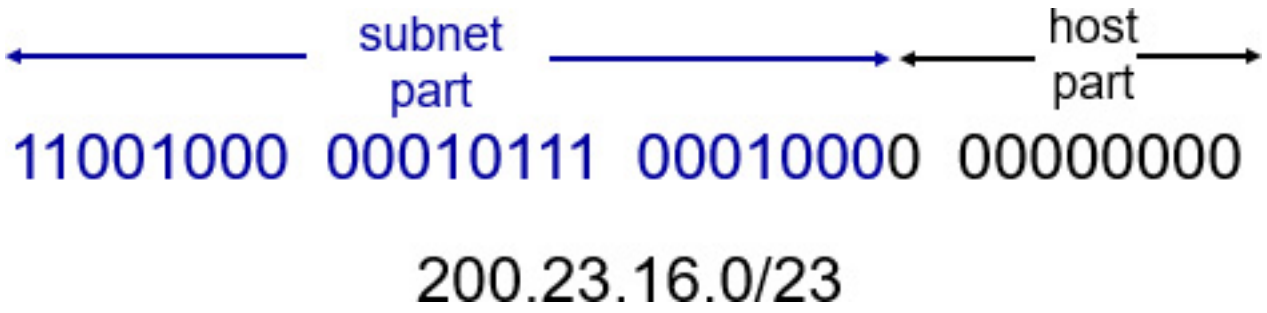
Classless Interdomain Routing (CIDR)

- A certain C class address space to an organisation doesn't guarantee that all addresses with the space can be used and therefor some addresses may be wasted
- This kind of situation is inflexible and would exhaust the IP address space
- The classful addressing scheme consists of class A, B, C, D and E results in an inefficient use of the address space for certain gateway routers
- Motivation:
  - A new scheme with no restriction on the classes emerged as the CIDR which is more flexible, allowing:
    - A variable length ***prefix field*** to represent the network
    - The remaining bits of the 32 field address to represent the hosts (normally don't care field for a gateway or router)
  - Example: One organisation may choose a 20 bit network ID, whereas another organisation may choose a 21 bit network ID, with first 20 bits of these two network IDs being identical. That means the address space of one organisation contains that of another one

IP Addressing: CIDR

CIDR: Classless **I**nter **D**omain **R**outing

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



IP Addresses: How to Get One? (1 of 2)

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

hard-coded by system admin in a file

- Windows: control-panel->network->configuration->tcp/ip->properties
- UNIX: /etc/rc.config

DHCP: Dynamic **H**ost **C**onfiguration **P**rotocol: dynamically get address from as server

- “plug-and-play”

DHCP: Dynamic Host Configuration Protocol

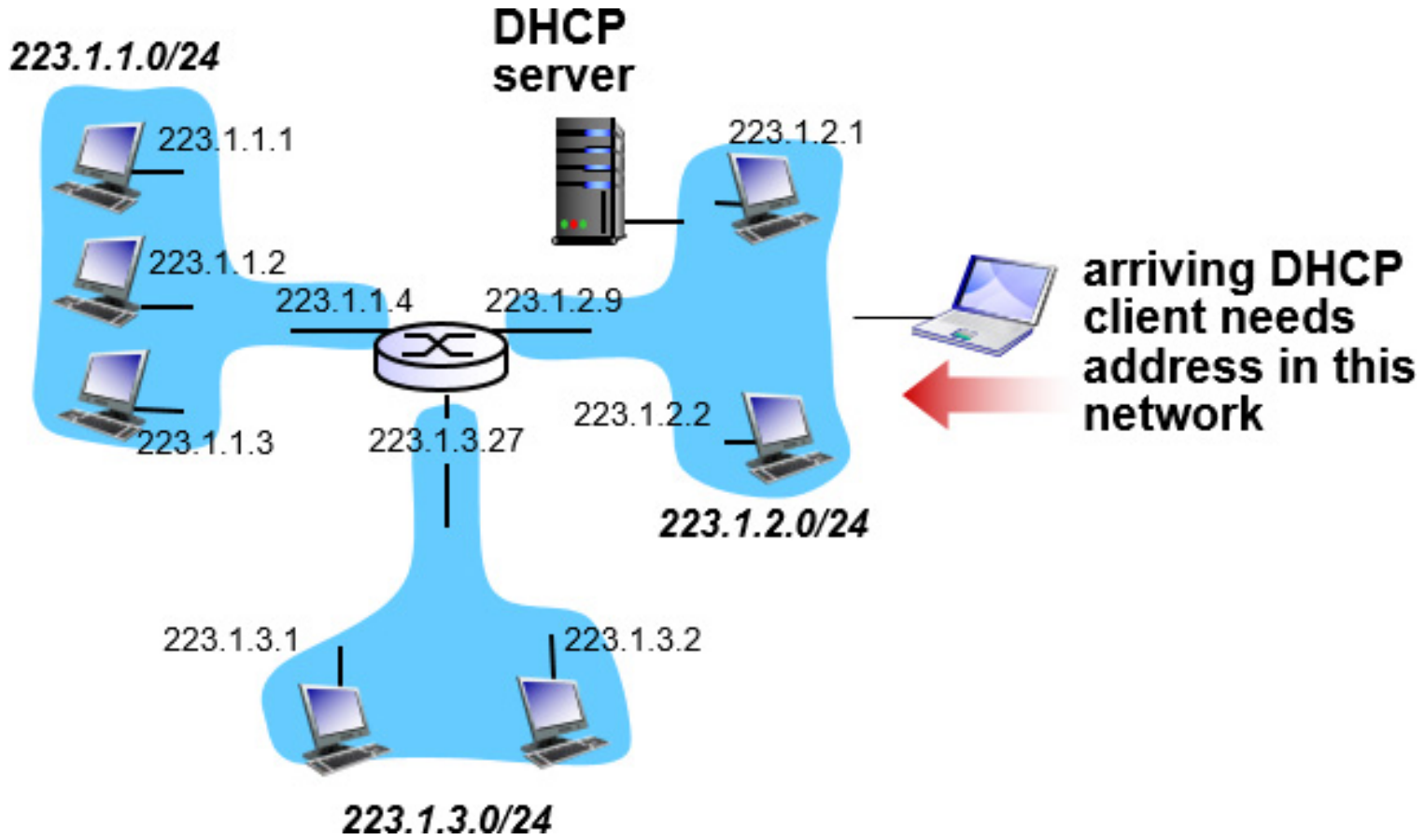
goal: allow host to **dynamically** obtain its IP address from network server when it joins network

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

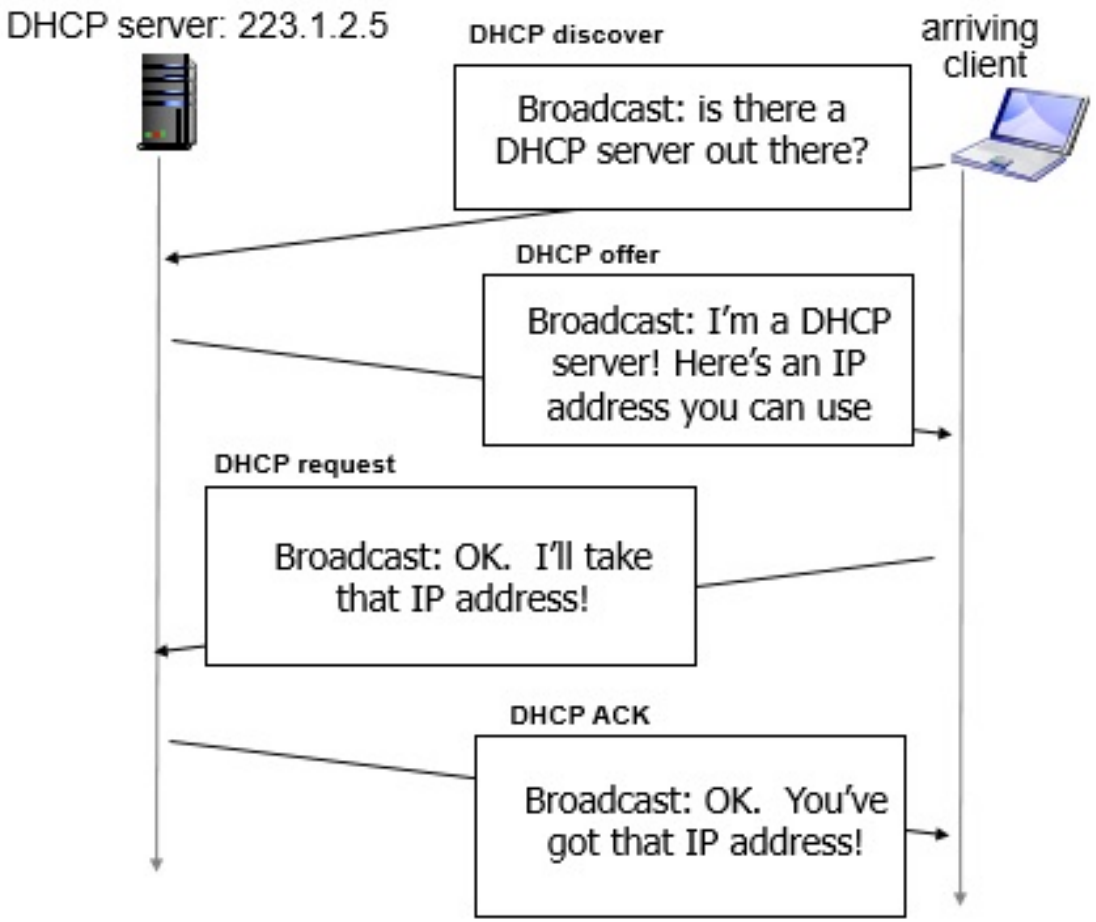
DHCP overview:

- 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 (1 of 2)



DHCP Client-Server Scenario (2 of 2)



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)

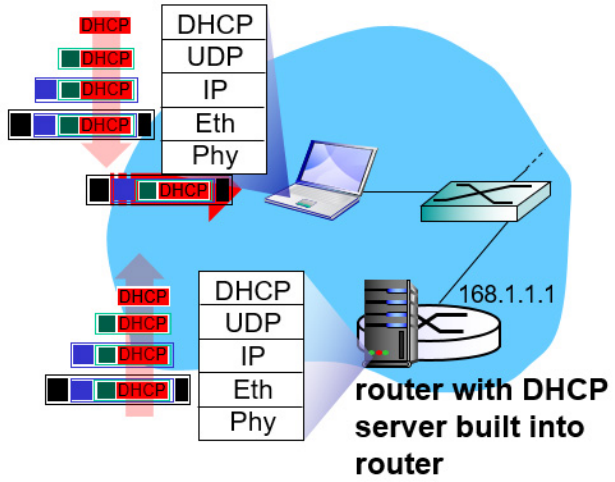
DHCP: Example (1 of 2)

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

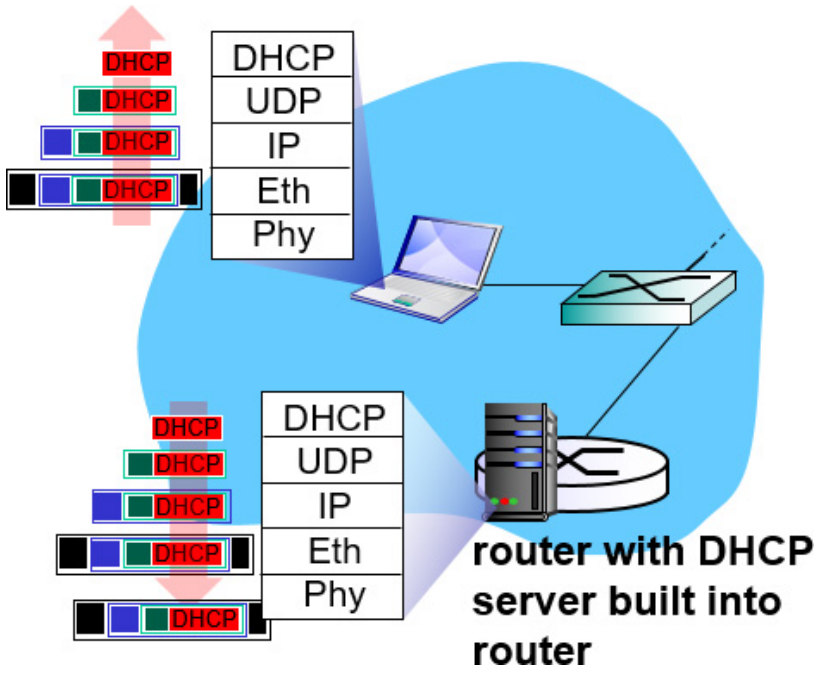


DHCP: Example (2 of 2)

DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DSN 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? (2 of 2)

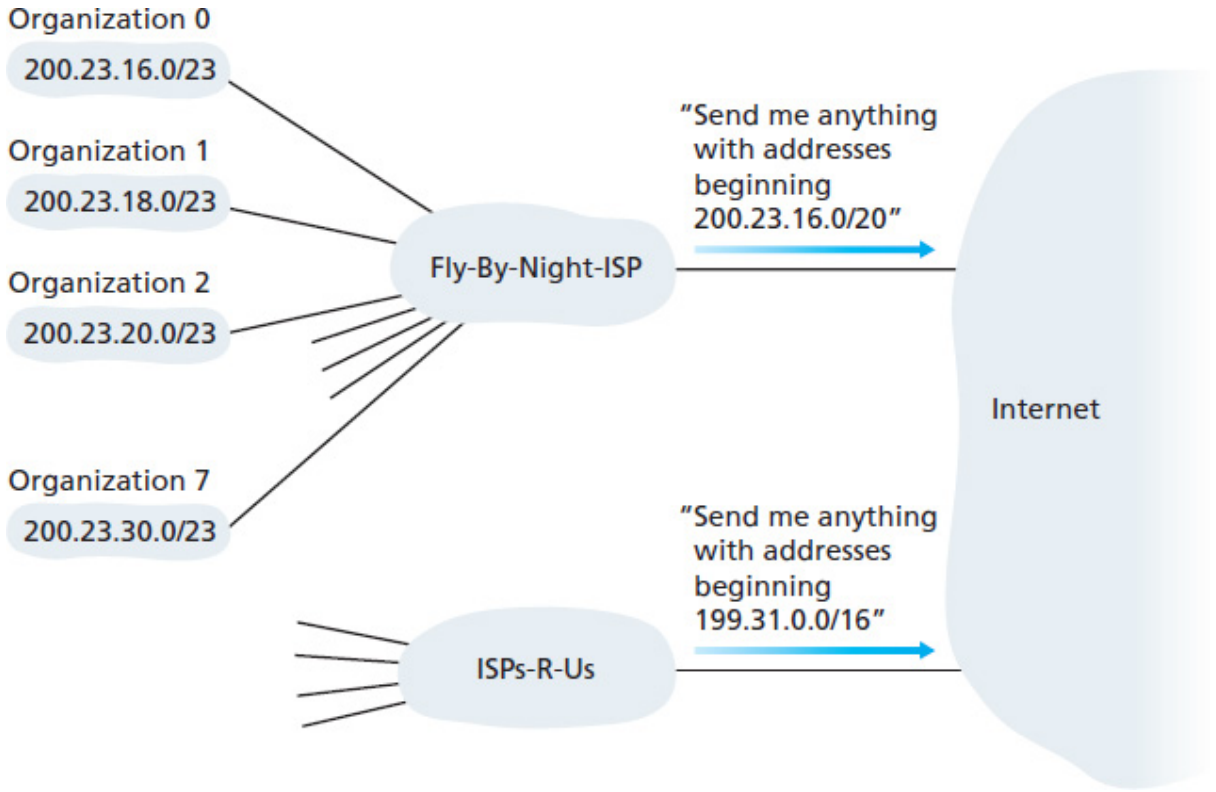
Q: how does **network** get subnet part of IP addr?

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

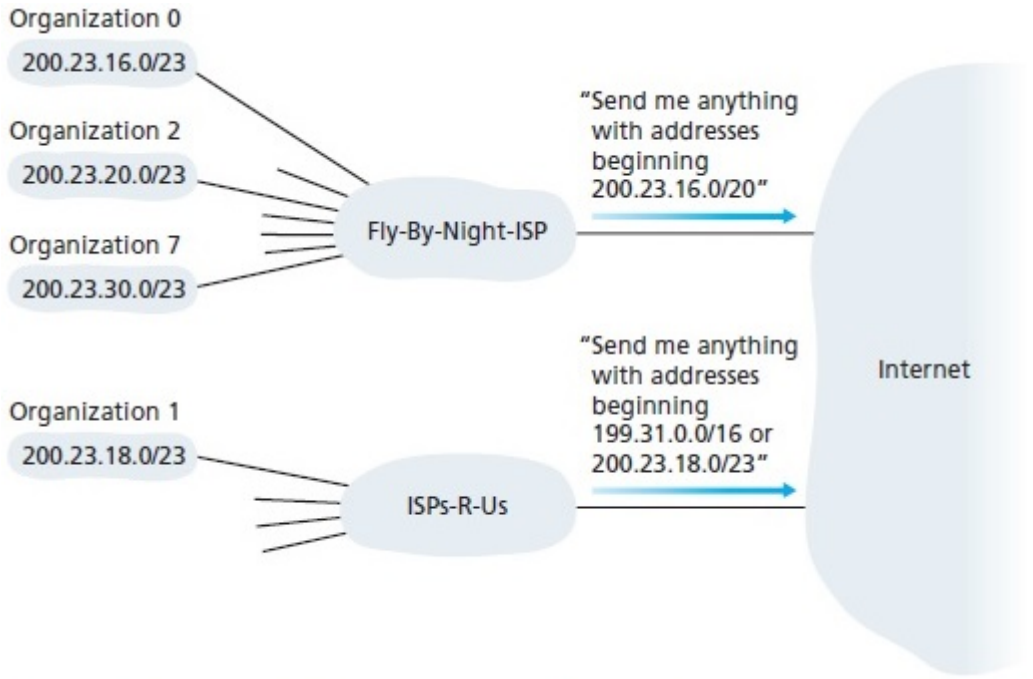
Hierarchical Addressing: Route Aggregation

hierarchical addressing allows efficient advertisement of routing information:



Hierarchical Addressing: More Specific Routes

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



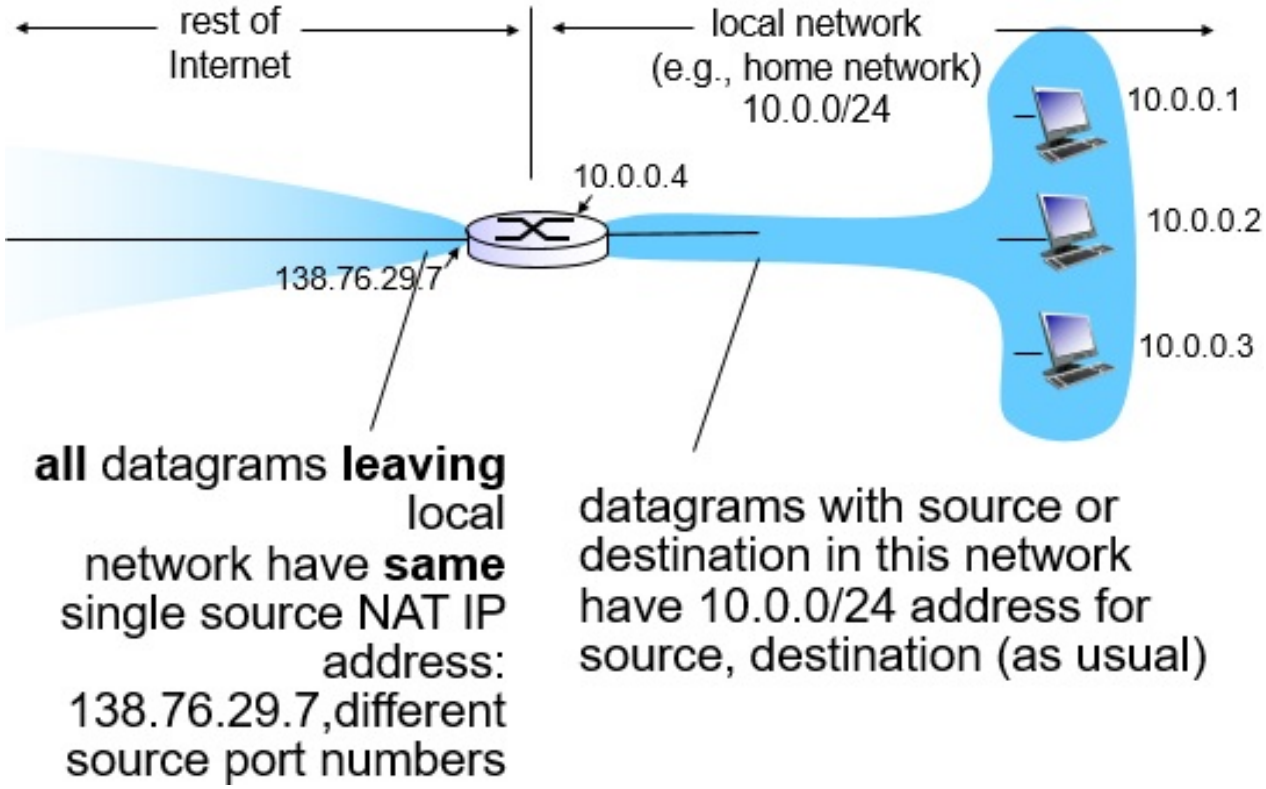
IP Addressing: The Last Word

Q: how does an ISP get block of addresses?

A: ICANN: Internet **C**orporation for **A**ssigned **N**ames and **N**umbers
http://www.icann.org/

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

NAT: Network Address Translation (1 of 5)



NAT: Network Address Translation (2 of 5)

motivation: local network uses just one IP address as far as outside world is concerned:

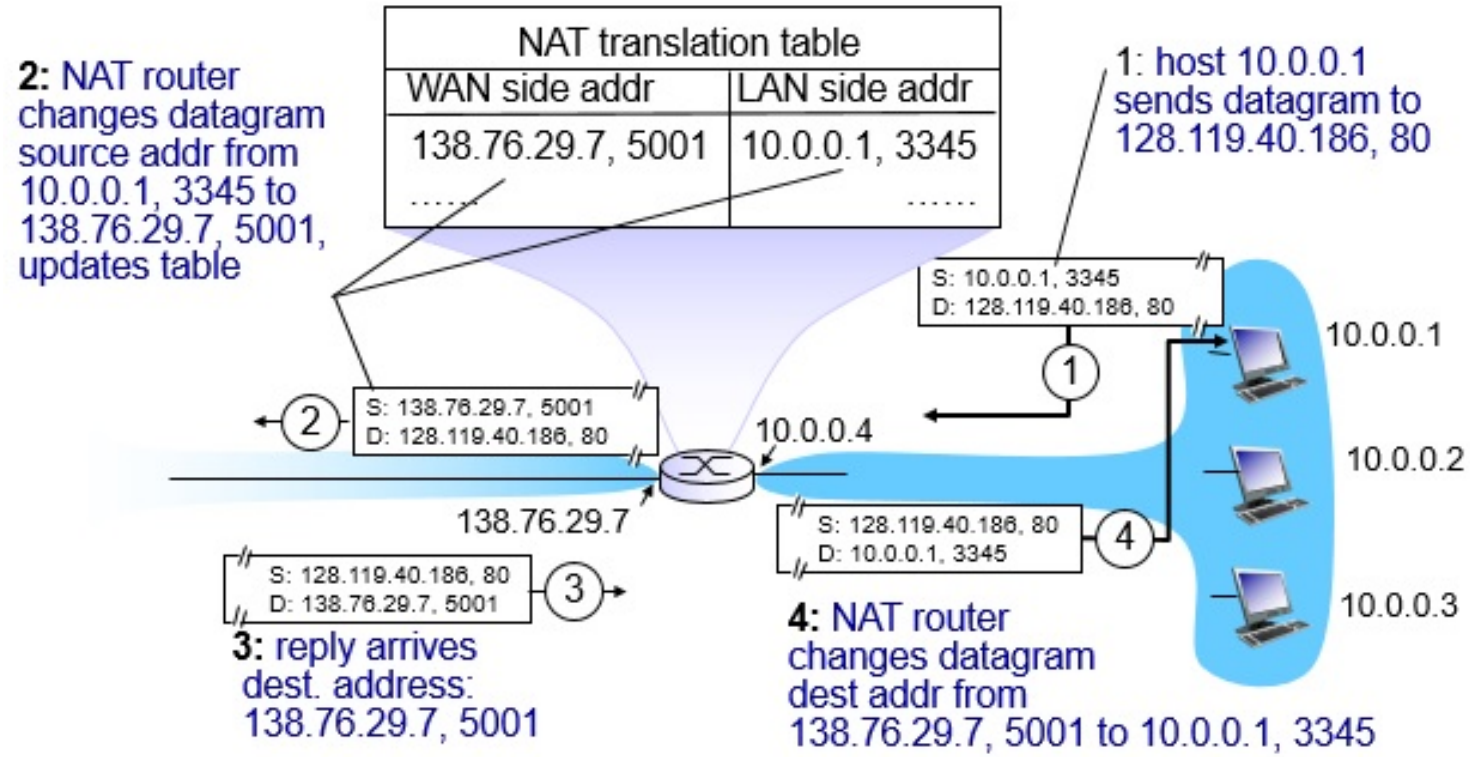
- range of addresses not needed from ISP: just one IP address for all devices
- can change addresses of devices in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- devices inside local net not explicitly addressable, visible by outside world (a security plus)

NAT: Network Address Translation (3 of 5)

implementation: NAT router must:

- **outgoing datagrams: replace** (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #) . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr
- **remember (in NAT translation table)** every (source IP address, port #) to (NAT IP address, new port #) translation pair
- **incoming datagrams: replace** (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: Network Address Translation (4 of 5)



NAT: Network Address Translation (5 of 5)

16-bit port-number field:

- 60,000 simultaneous connections with a single LAN-side address!

NAT is controversial:

- routers should only process up to layer 3
- address shortage should be solved by IPv6
- violates end-to-end argument
  - NAT possibility must be taken into account by app designers, e.g., P2P applications
- NAT traversal: what if client wants to connect to server behind NAT?