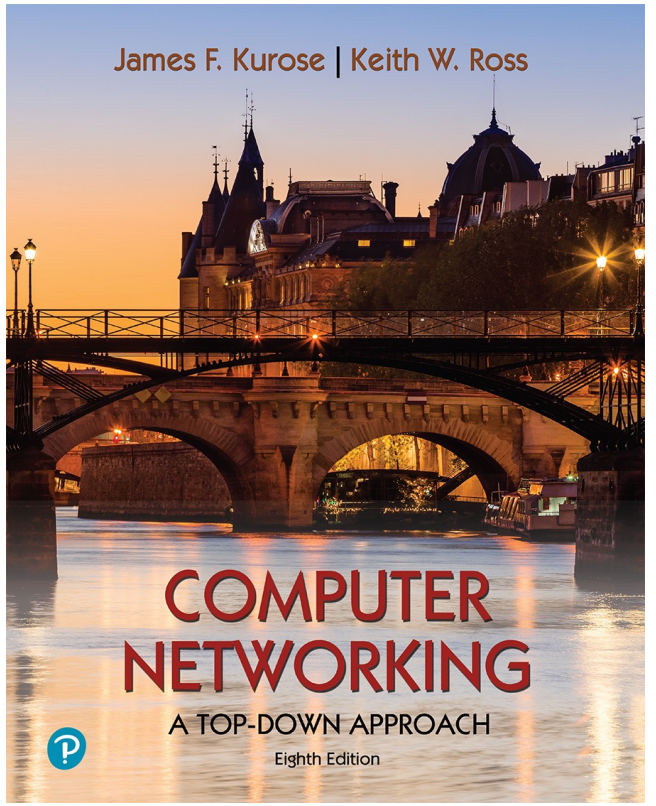
Chapter 4
Network Layer: Data Plane

Yaxiong Xie

Department of Computer Science and Engineering
University at Buffalo, SUNY

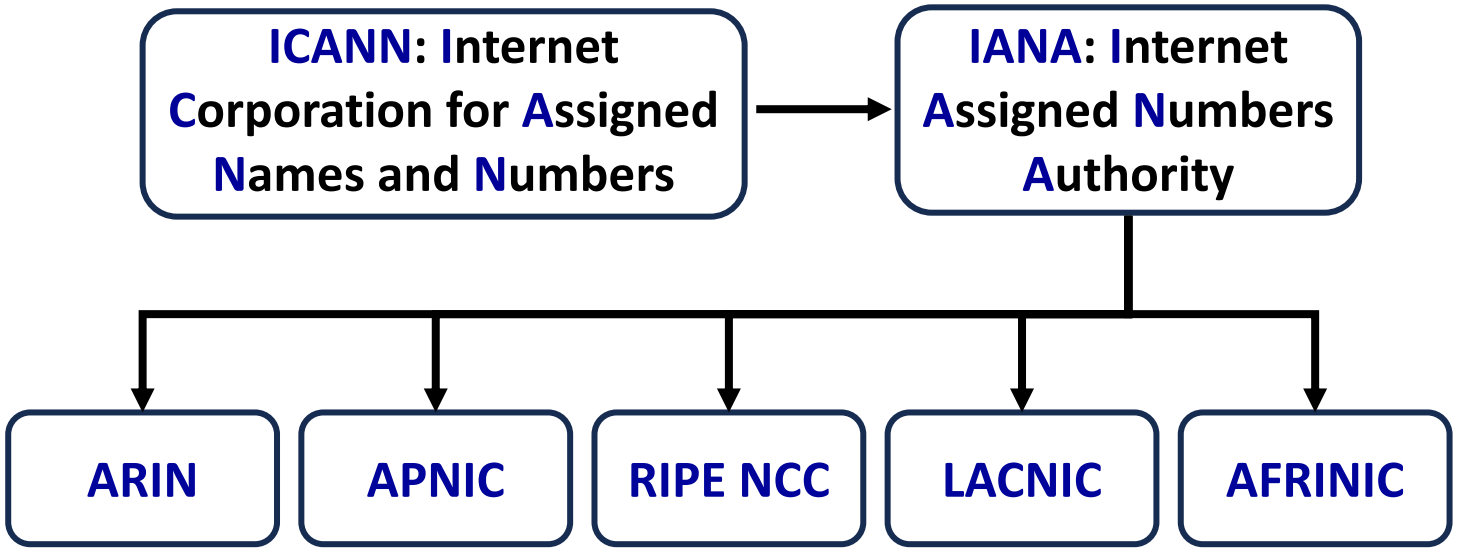


Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Adapted from the slides of the book's authors

How to obtain the IP address?

IP Address Allocation: A Top Down View



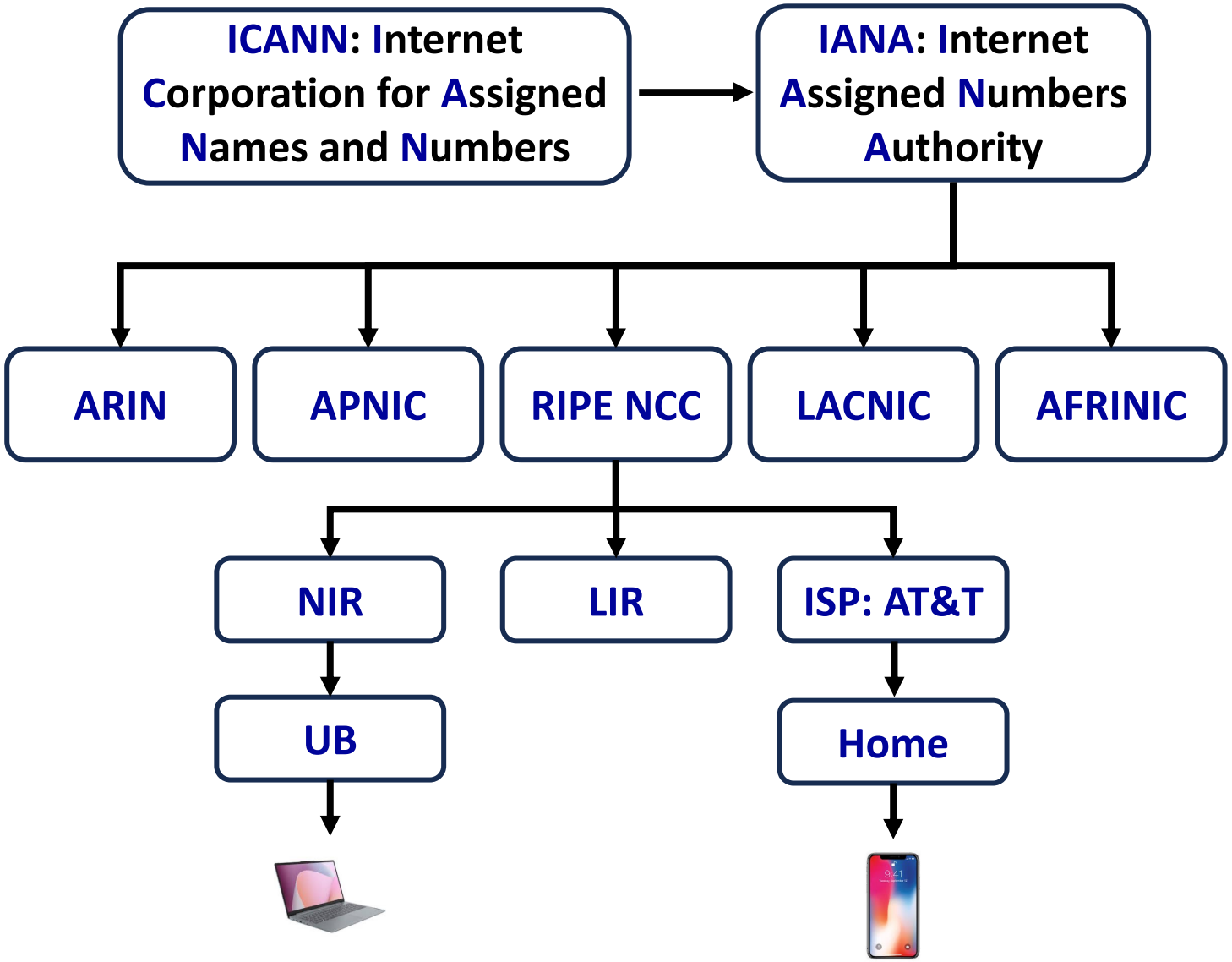
ICANN: non-profit organization that coordinates internet policies and oversees the domain system

IANA: a department *within* ICANN that manages IP address

RIR: Regional Internet Registry

- **ARIN** serving North America founded in 1997
- **APNIC** serving the Asia Pacific region founded in 1993
- **RIPE NCC** serving Europe, Central Asia and the Middle East founded in 1992
- **LACNIC** serving South America and the Caribbean founded in 2001
- **AFRINIC** Serving Africa Founded in 2005

IP Address Allocation: A Top Down View



ICANN: non-profit organization that coordinates internet policies and oversees the domain system

IANA: a department *within* ICANN that manages IP address

RIR: Regional Internet Registry

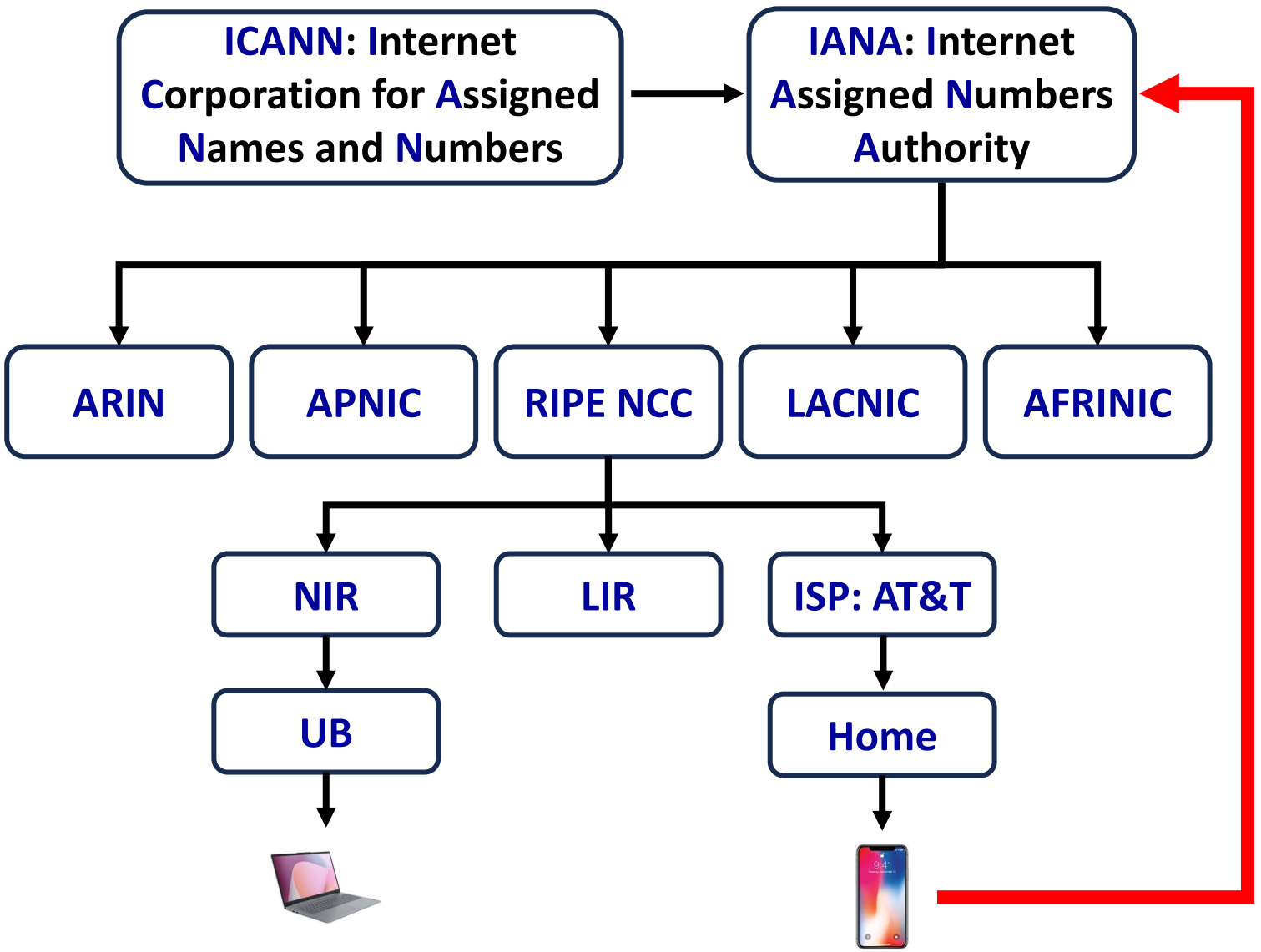
NIR: National Internet Registry

LIR: Local Internet Registry

ISP: Internet Service Provider

IANA is a top-level coordinator, not the entity that hands out addresses to end devices

IP Address Allocation: A Top Down View

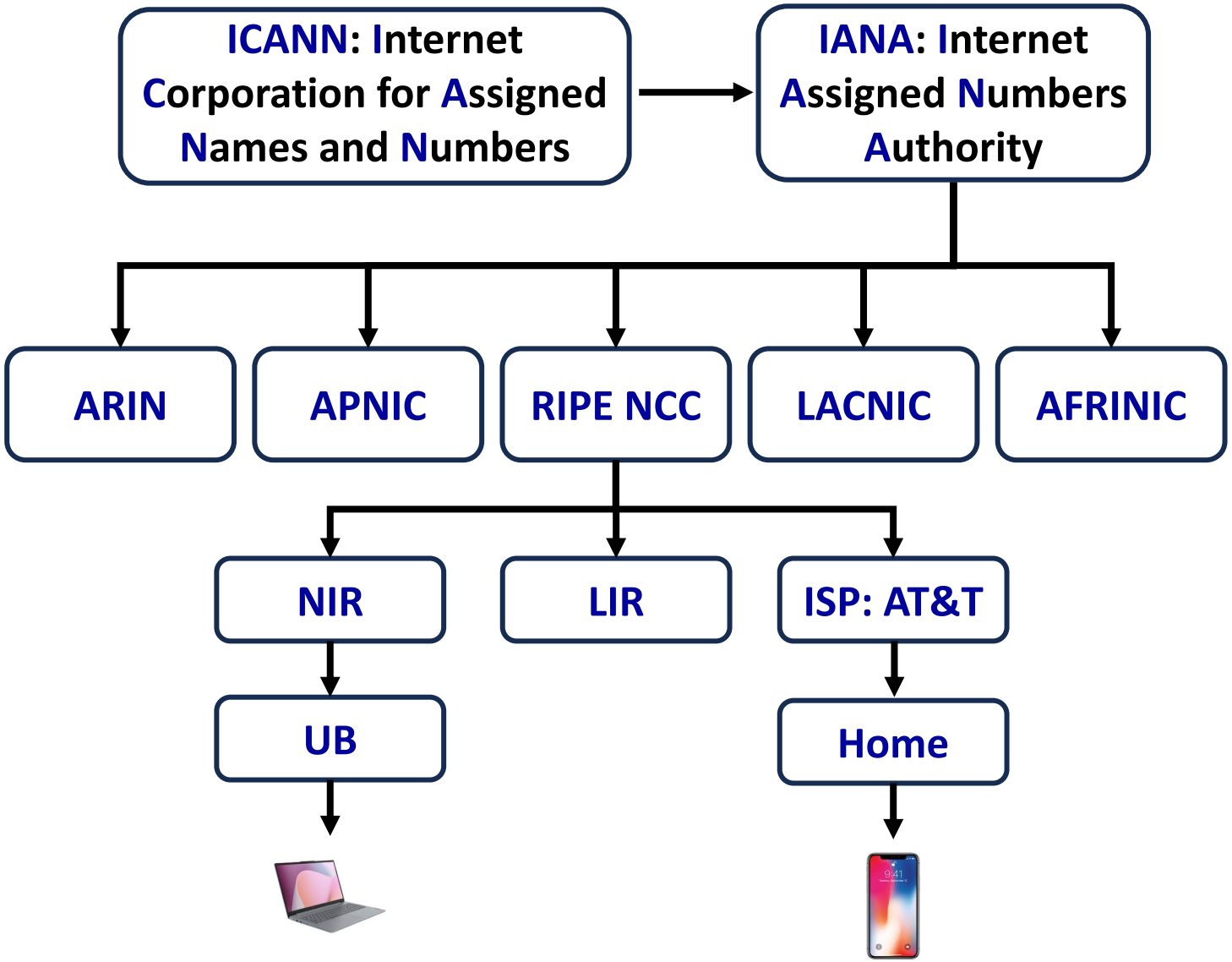


IPv4: 4 bytes

2^{32} = 4,294,967,296

IANA is a top-level coordinator, not the entity that hands out addresses to end devices

IP Address Allocation: A Top Down View



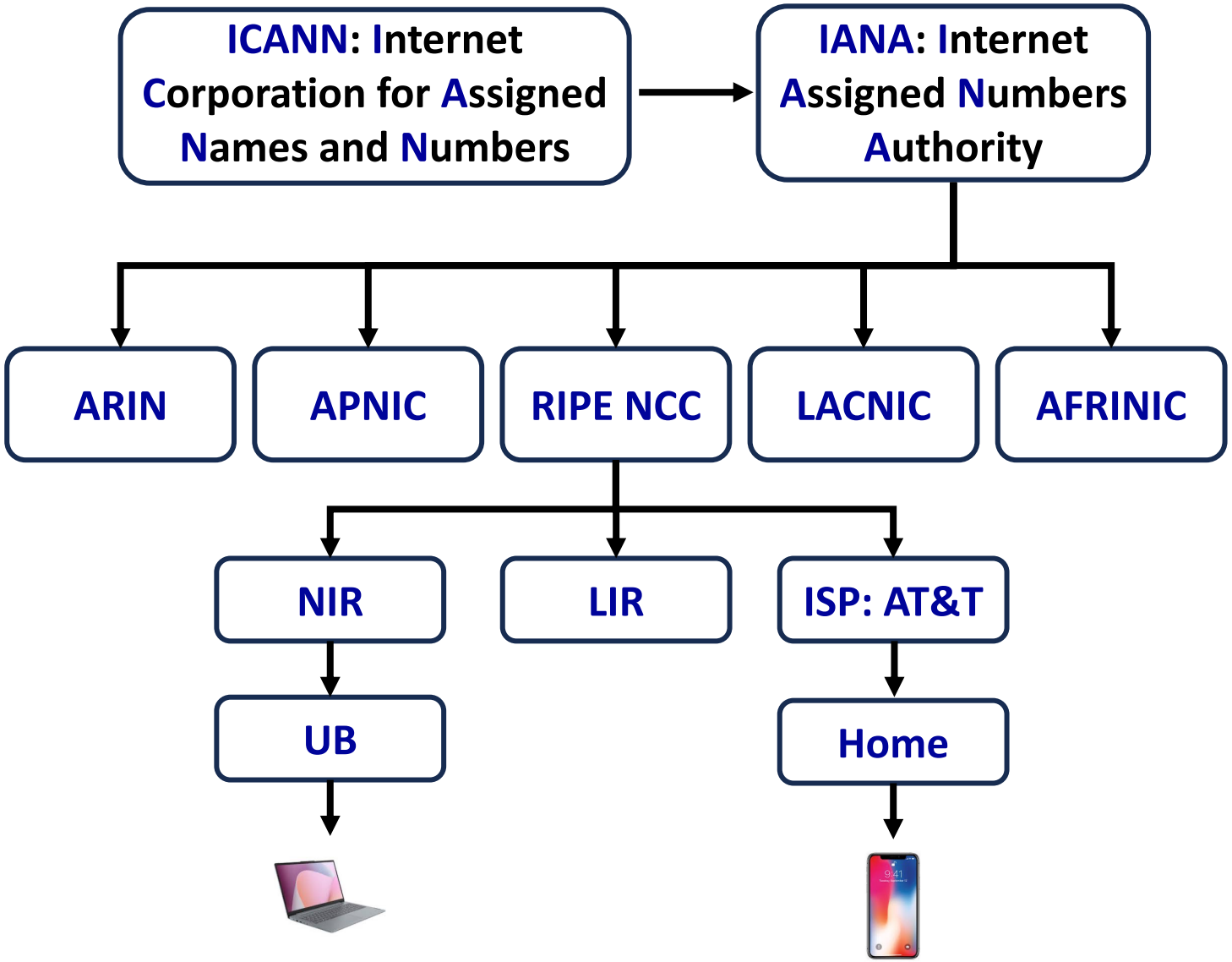
IPv4: 4 bytes

2^{32} = 4,294,967,296

IANA is a top-level coordinator, not the entity that hands out addresses to end devices

IANA helps ensure global uniqueness and consistency.

IP Address Allocation: A Top Down View



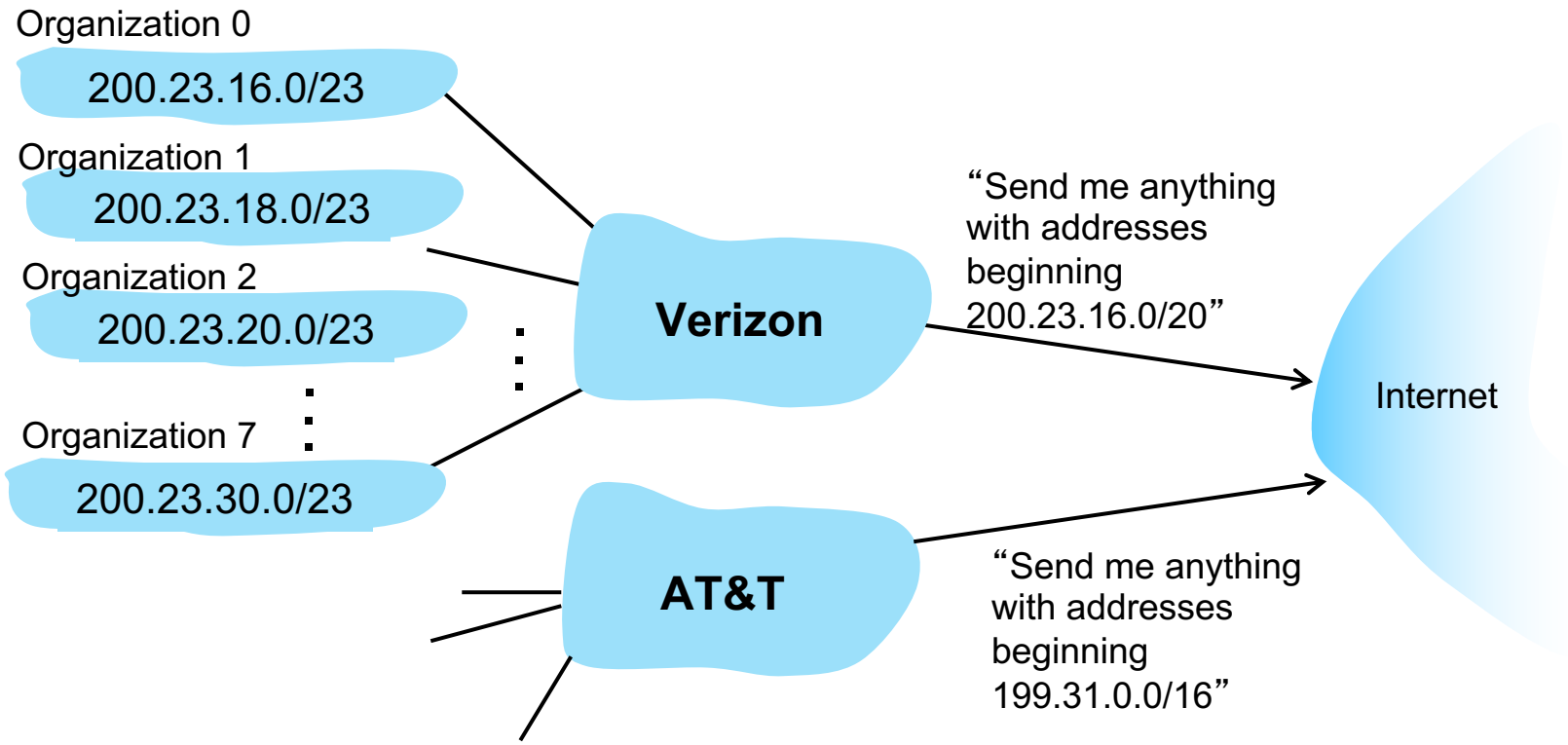
IPv4: 4 bytes

2^{32} = 4,294,967,296

IANA supports a hierarchical allocation structure that makes the system **easier to manage**, **easier to scale**, and better **aligned with route aggregation** in interdomain routing

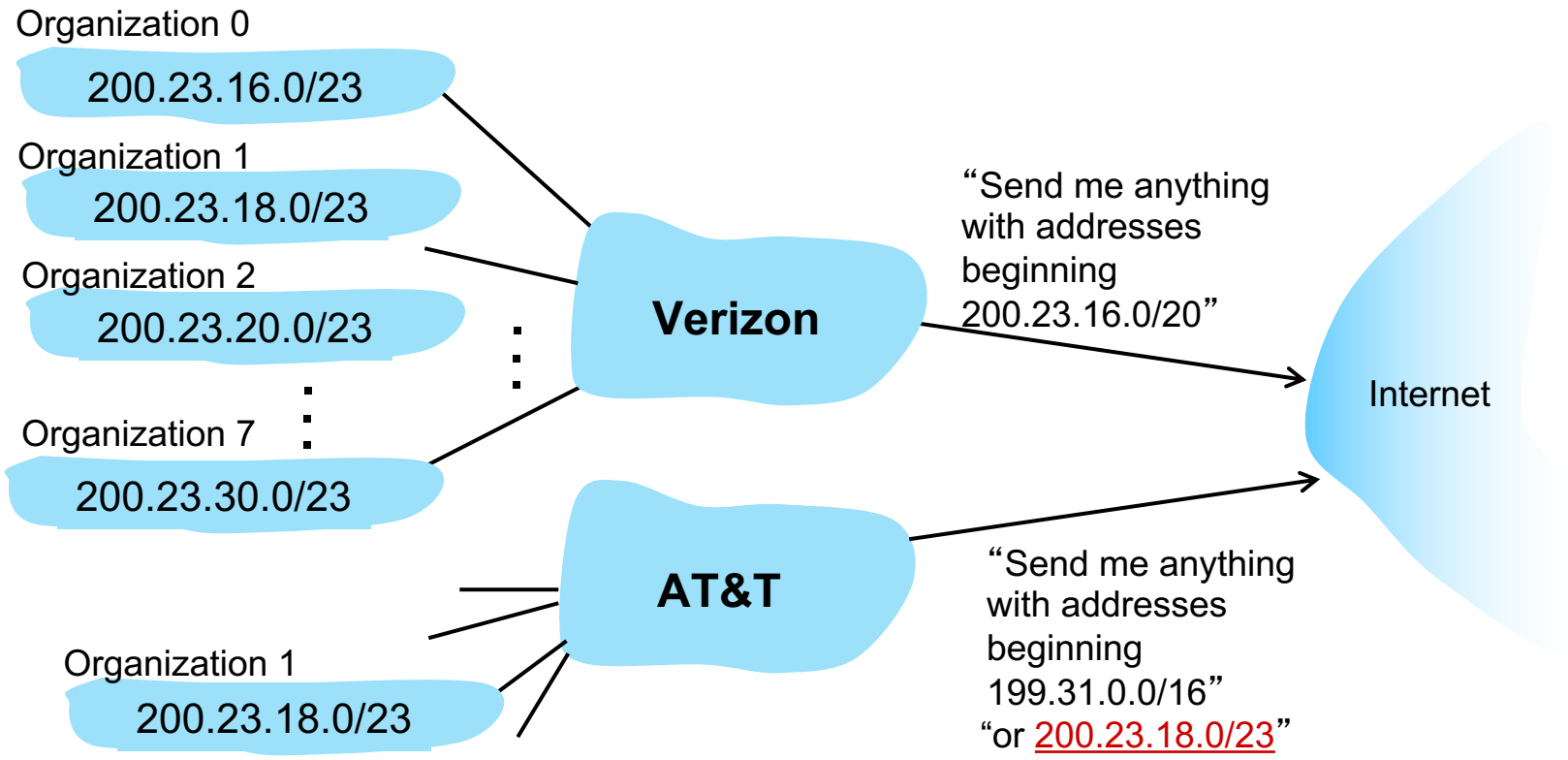
Hierarchical addressing: route aggregation

hierarchical addressing allows efficient advertisement of routing information:



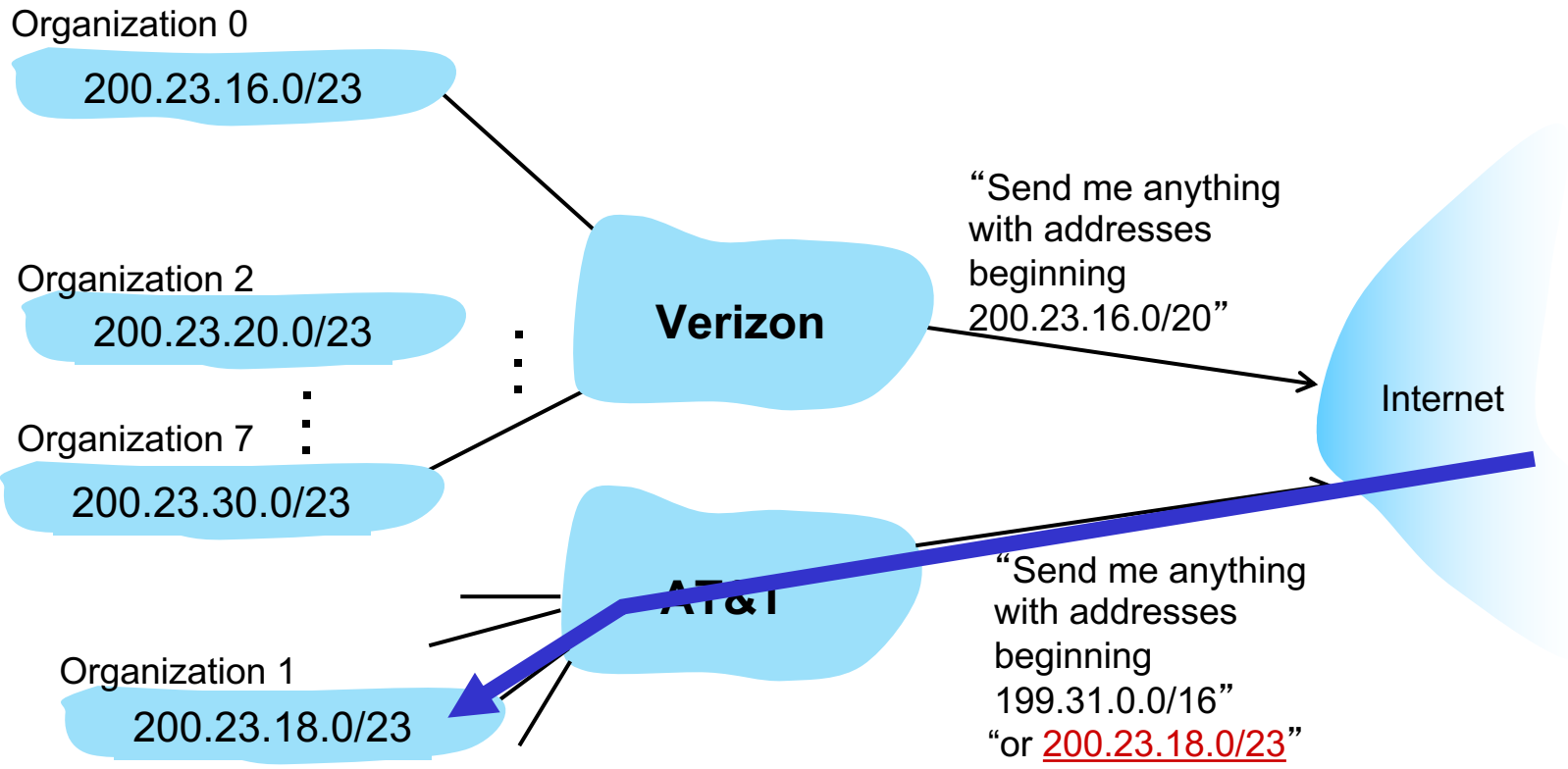
Hierarchical addressing: more specific routes

- Organization 1 moves from ISP-1 (Verizon) to ISP-2 (AT&T)
- ISP-2 (AT&T) now advertises a more specific route to Organization 1

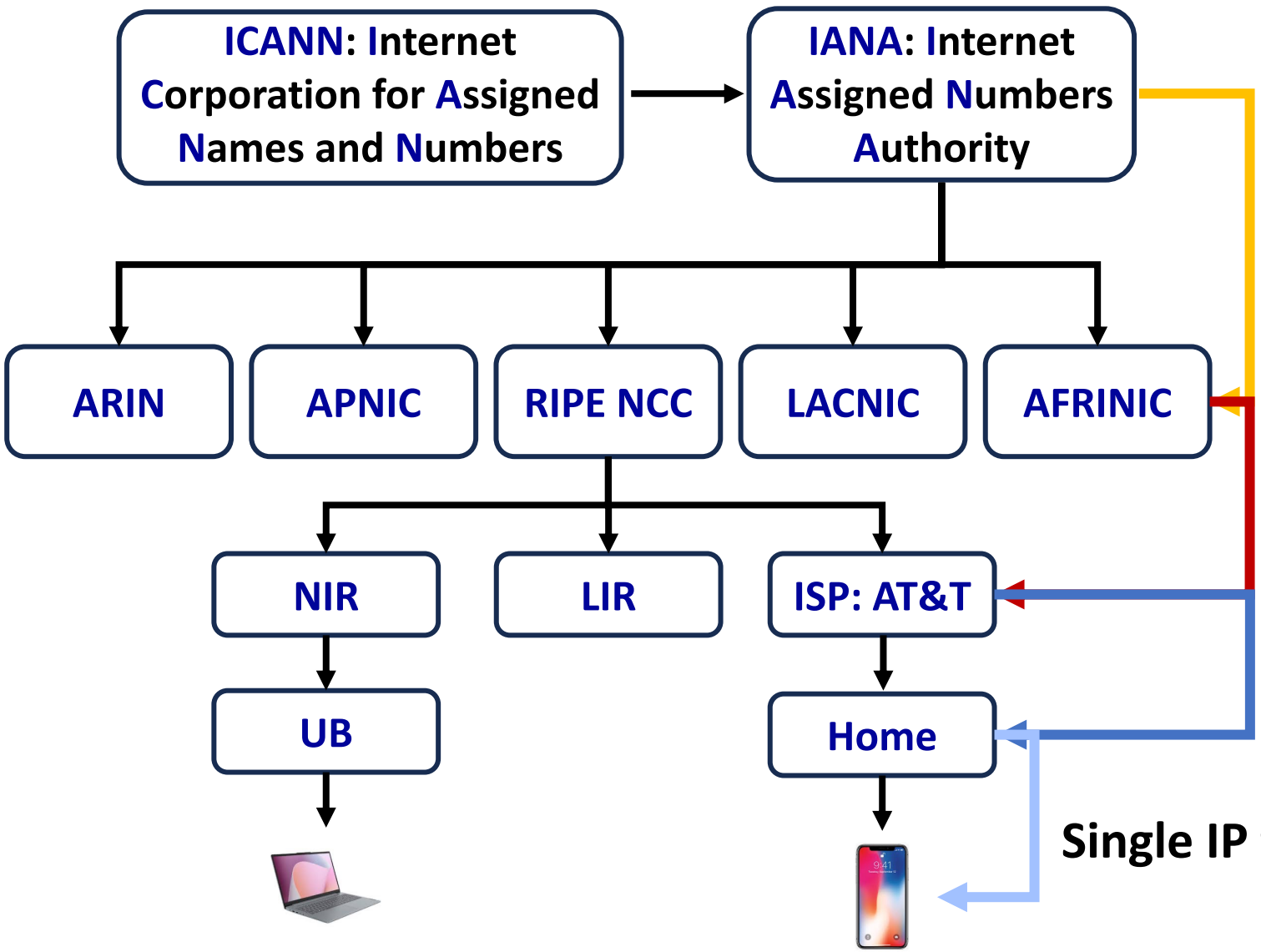


Hierarchical addressing: more specific routes

- Organization 1 moves from ISP-1 (Verizon) to ISP-2 (AT&T)
- ISP-2 (AT&T) now advertises a more specific route to Organization 1



IP Address Allocation: A Top Down View



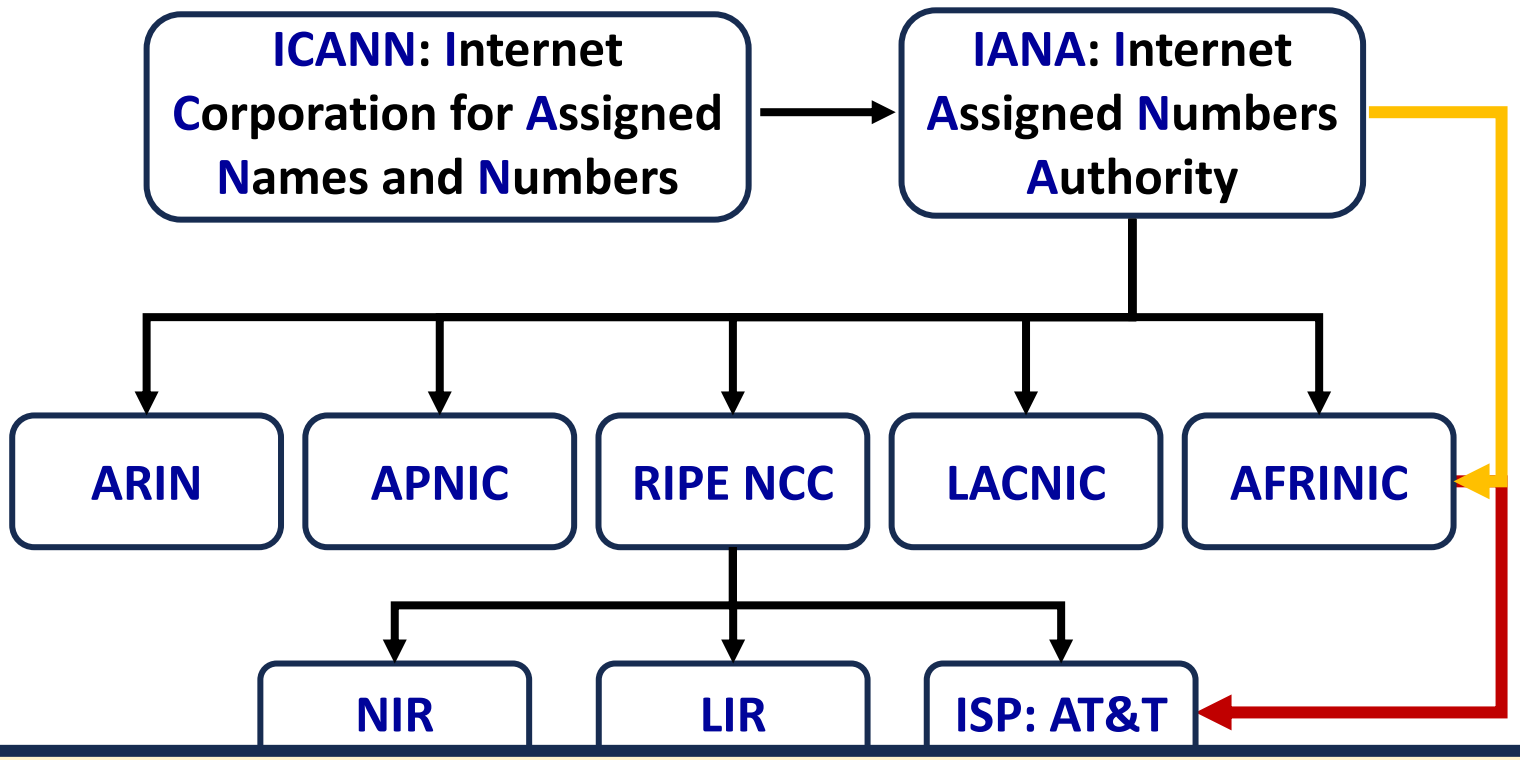
IPv4: 4 bytes

2^{32} = 4,294,967,296

Subnet or a pool of IP address

Single IP to the device

IP Address Allocation: A Top Down View



IPv4: 4 bytes

2^{32} = 4,294,967,296

How does IANA and RIR allocate IP

IP Address Allocation: IANA and RIR

Key Idea: **Divide the IP address into different class**

Network Portion (fixed for a subnet) | **Host Portion (varies with host)**

11000000. 10101000. 00000001. 00000000

11000000. 10101000. 00000001. 01111111

The **number of bits** in the **host portion** decides how many IP address in the subnet or class

11111111. 11111111. 11111111. 10000000 → 7 bits $2^7 = 128$

Network Portion | **Host Portion**

Subnet Mask: 255. 255. 255. 128

IP Address Allocation: IANA and RIR

Key Idea: **Divide the IP address into different class**

Network Portion (fixed for a subnet) | **Host Portion (varies with host)**

11000000. 10101000. 00000001. 00000000

11000000. 10101000. 00000001. 01111111

The **number of bits** in the **host portion** decides how many IP address in the subnet or class

11111111. 11111111. 11111111. 00000000 → 8 bits $2^8 = 256$

Network Portion | **Host Portion**

Subnet Mask: 255. 255. 255. 128

IP Address Allocation: IANA and RIR

Key Idea: **Divide the IP address into different class**

Network Portion (fixed for a subnet) | **Host Portion (varies with host)**

11000000. 10101000. 00000001. 00000000

11000000. 10101000. 00000001. 01111111

The **number of bits** in the **host portion** decides how many IP address in the subnet or class

11111111. 11111111. 00000000. 00000000 → 16 bits $2^{16} = 65536$

Network Portion | **Host Portion**

Subnet Mask: 255. 255. 255. 128

IP Address Allocation: IANA and RIR

Key Idea: **Divide the IP address into different class**

Network Portion (fixed for a subnet) | **Host Portion (varies with host)**

11000000. 10101000. 00000001. 00000000

11000000. 10101000. 00000001. 01111111

The **number of bits** in the **host portion** decides how many IP address in the subnet or class

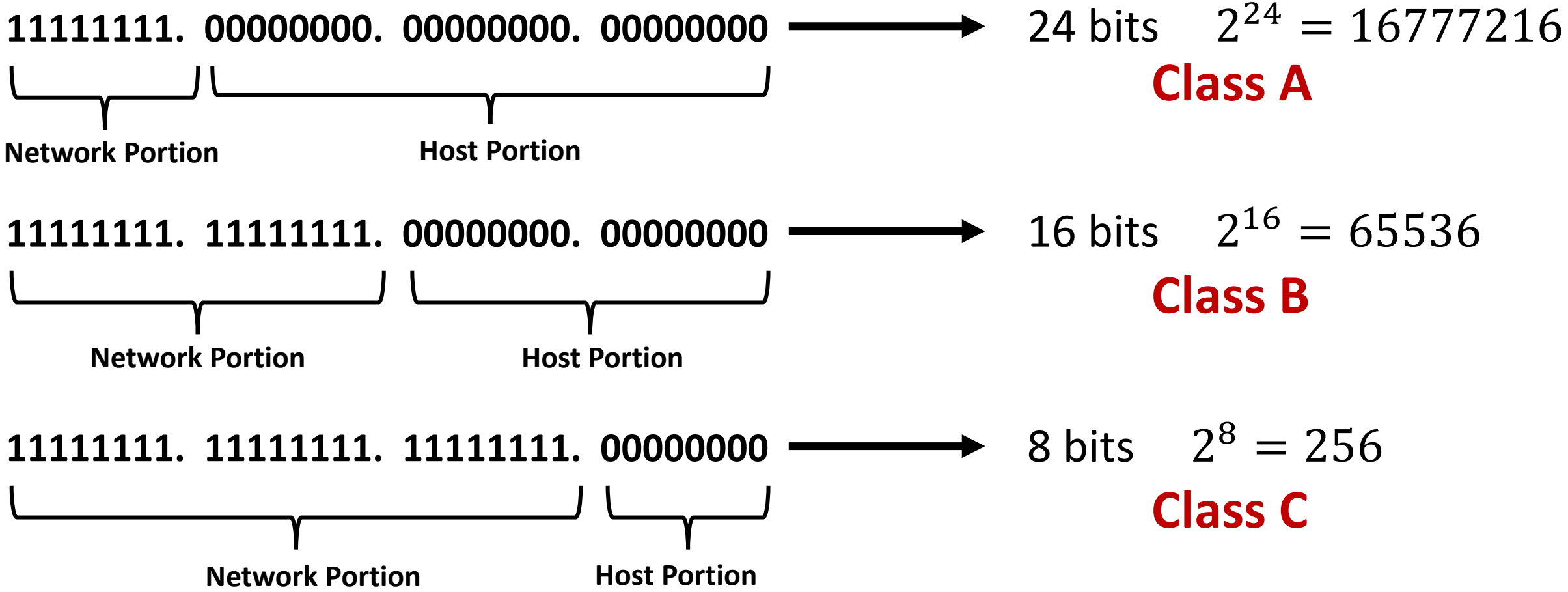
11111111. 00000000. 00000000. 00000000 → 24 bits $2^{24} = 16777216$

Network Portion | **Host Portion**

Subnet Mask: 255. 255. 255. 128

IP Address Allocation: IANA and RIR

Key Idea: **Divide the IP address into different class**



IP Address Allocation: IANA and RIR

Key Idea: **Divide the IP address into different class**

24 bits $2^{24} = 16777216$

Class A

16 bits $2^{16} = 65536$

Class B

8 bits $2^{8} = 256$

Class C

A company needs **2000** IP address.

IP Address Allocation: IANA and RIR

Key Idea: **Divide the IP address into different class**

24 bits $2^{24} = 16777216$

Class A

16 bits $2^{16} = 65536$

Class B

8 bits $2^{8} = 256$

Class C

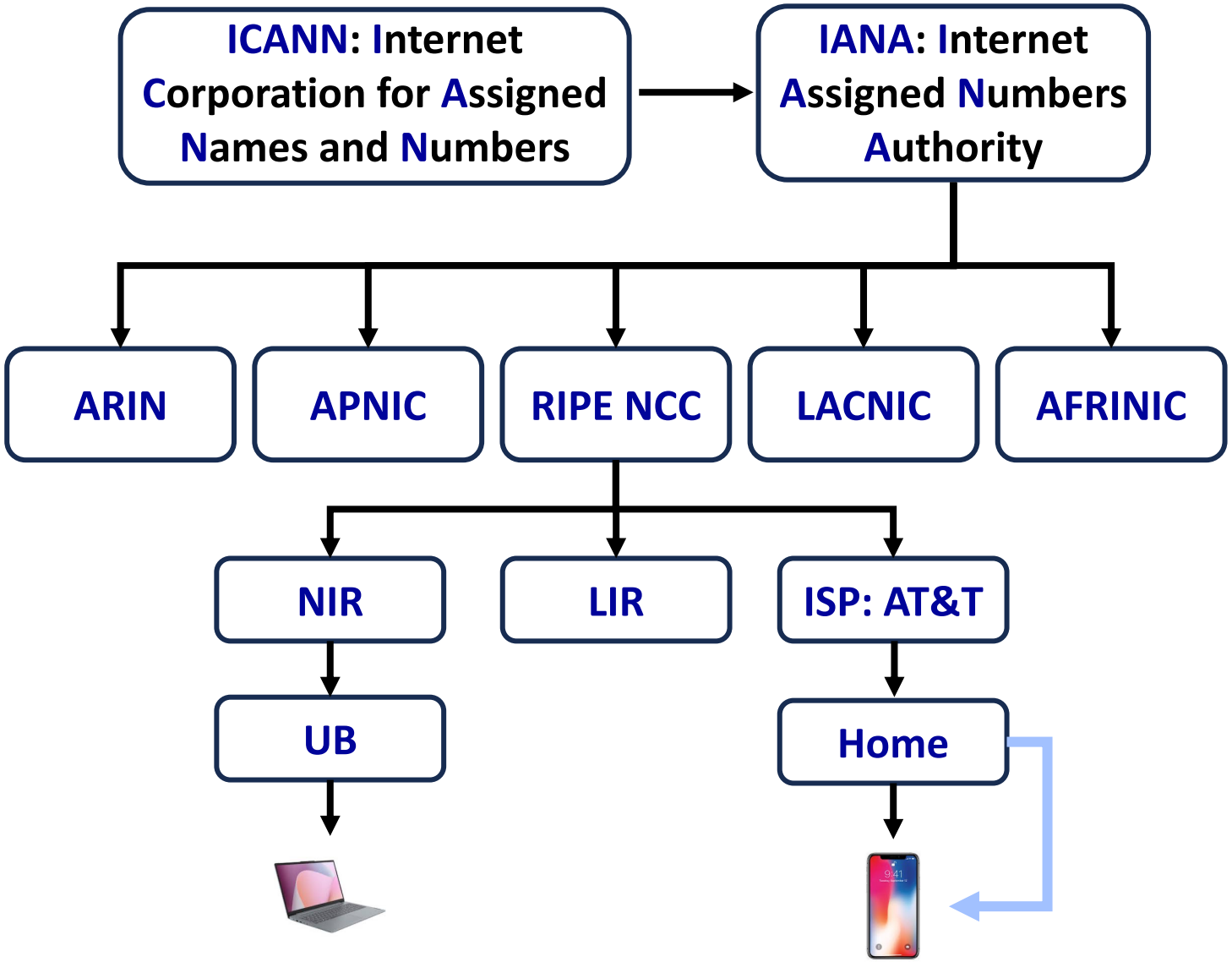
A company needs **2000** IP address.

CIDR: **C**lassless InterDomain Routing

(pronounced “cider”)

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

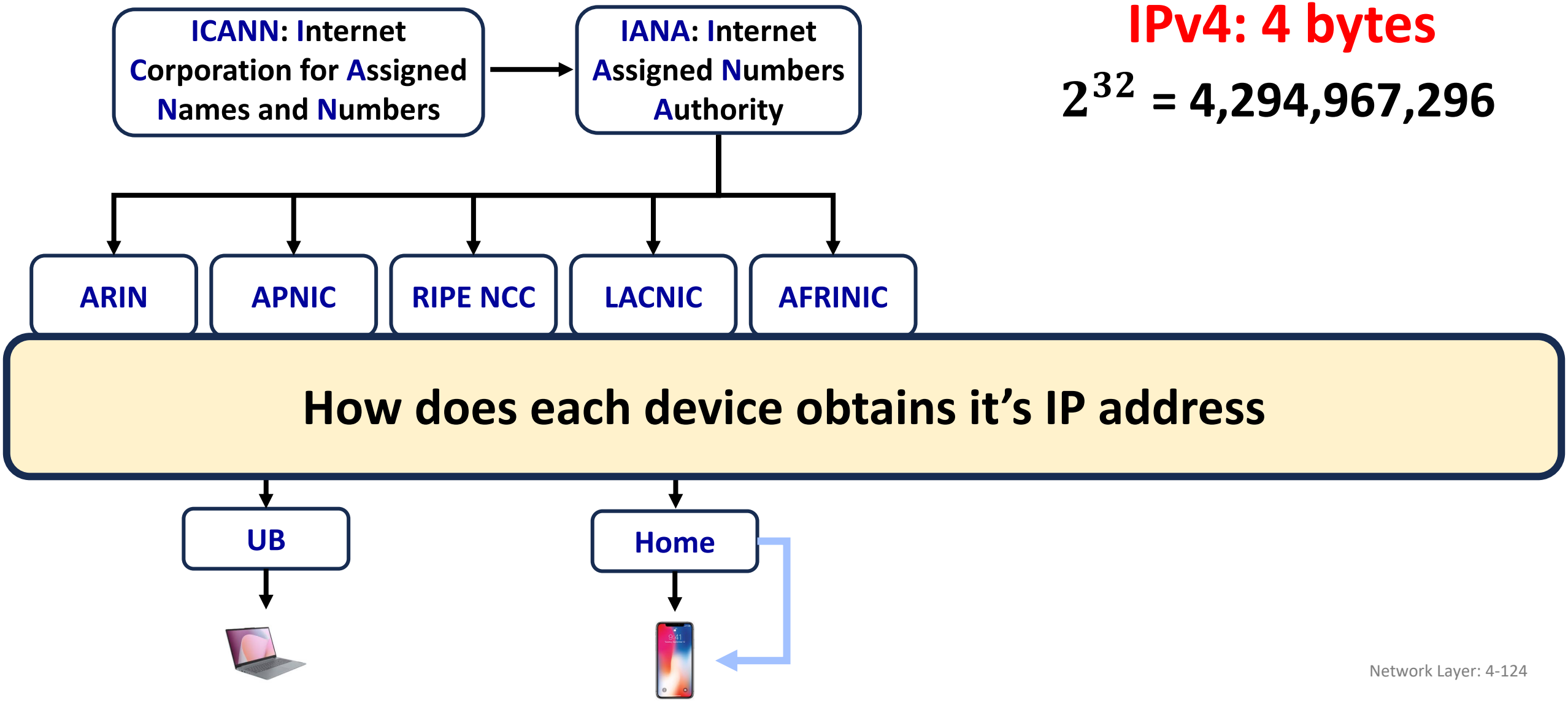
IP Address Allocation: A Top Down View



IPv4: 4 bytes

2^{32} = 4,294,967,296

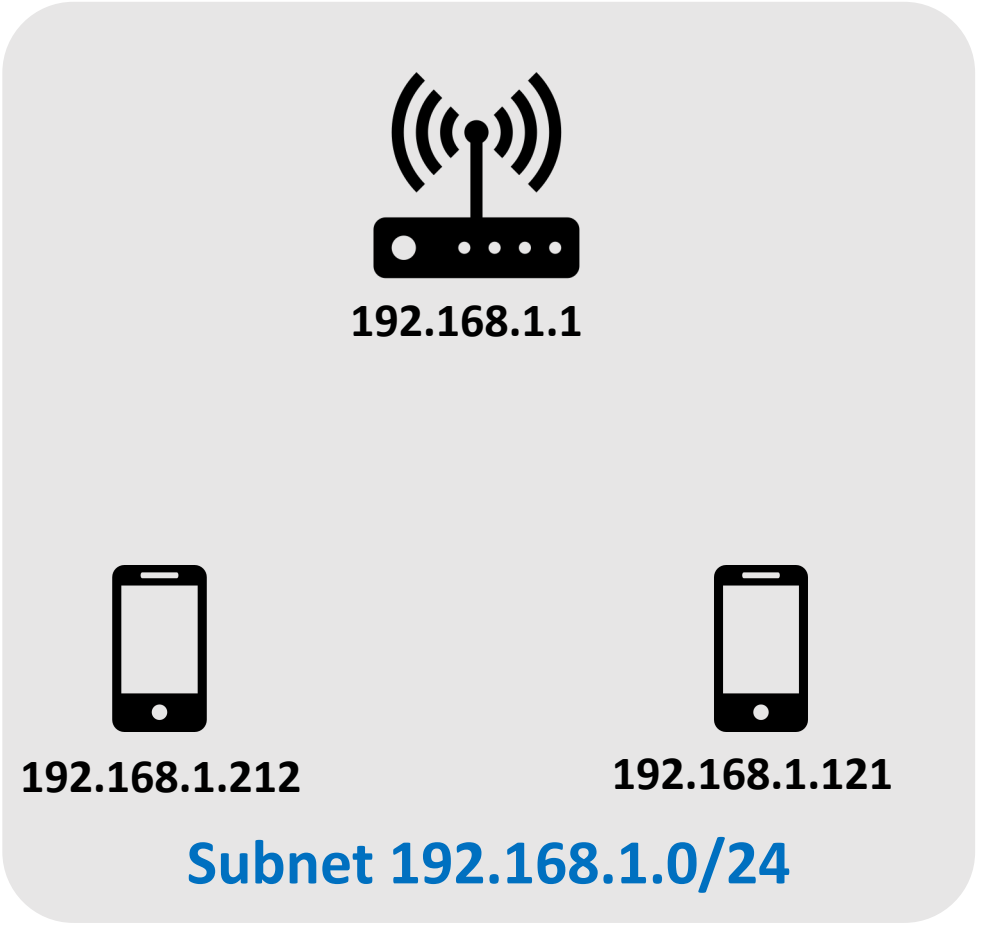
IP Address Allocation: A Top Down View



IPv4: 4 bytes

2^{32} = 4,294,967,296

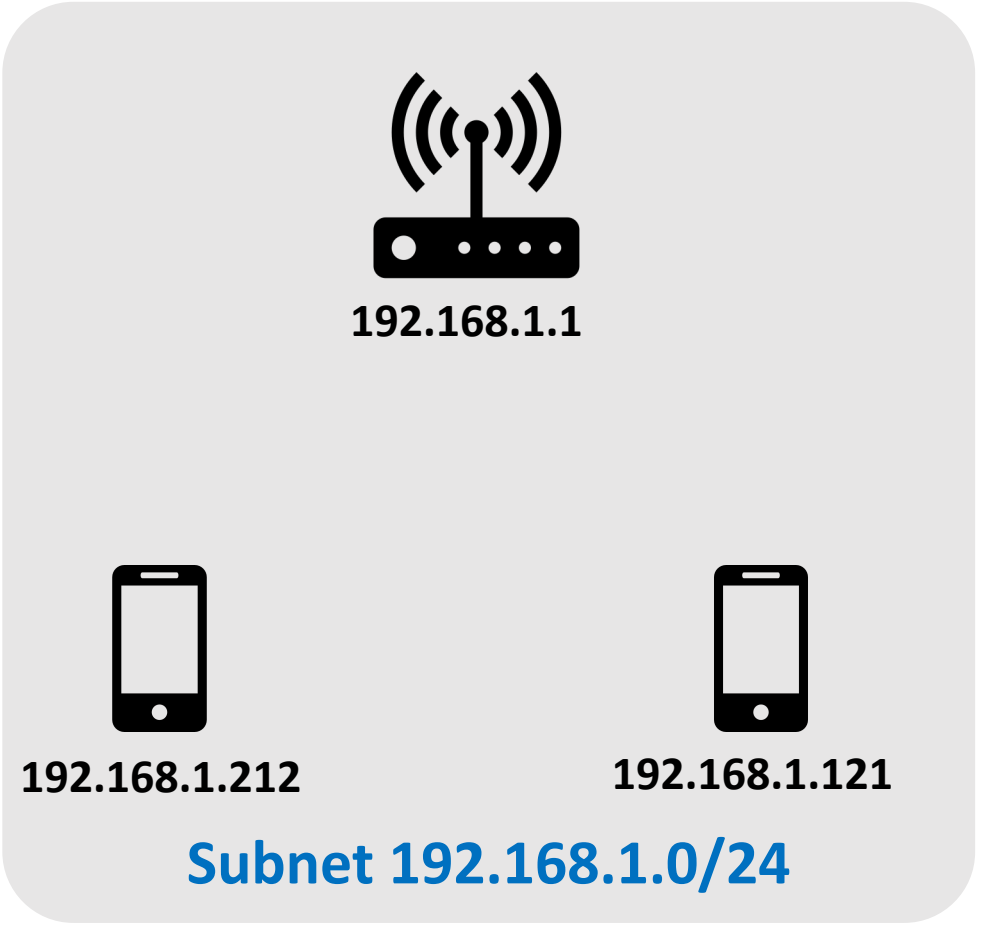
IP address of each device: How to obtain it?



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"

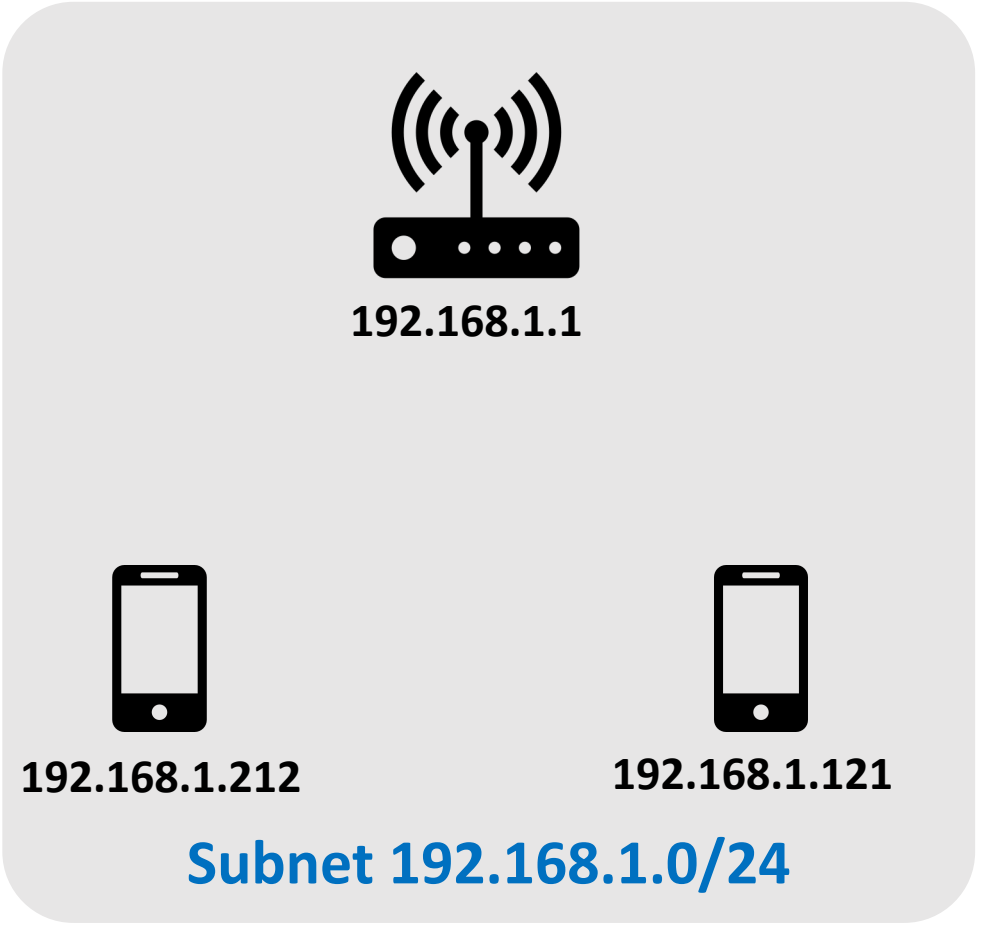
DHCP: Dynamic Host Configuration Protocol



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

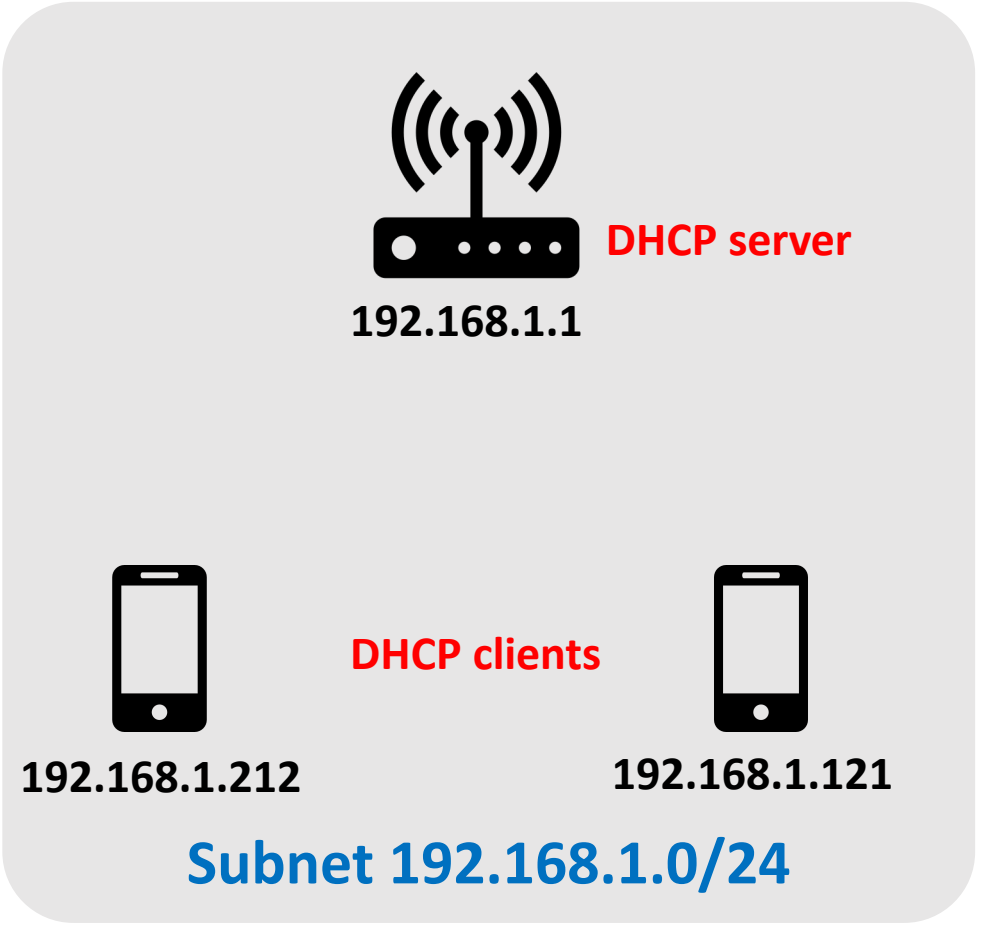
DHCP: Dynamic Host Configuration Protocol



DHCP overview:

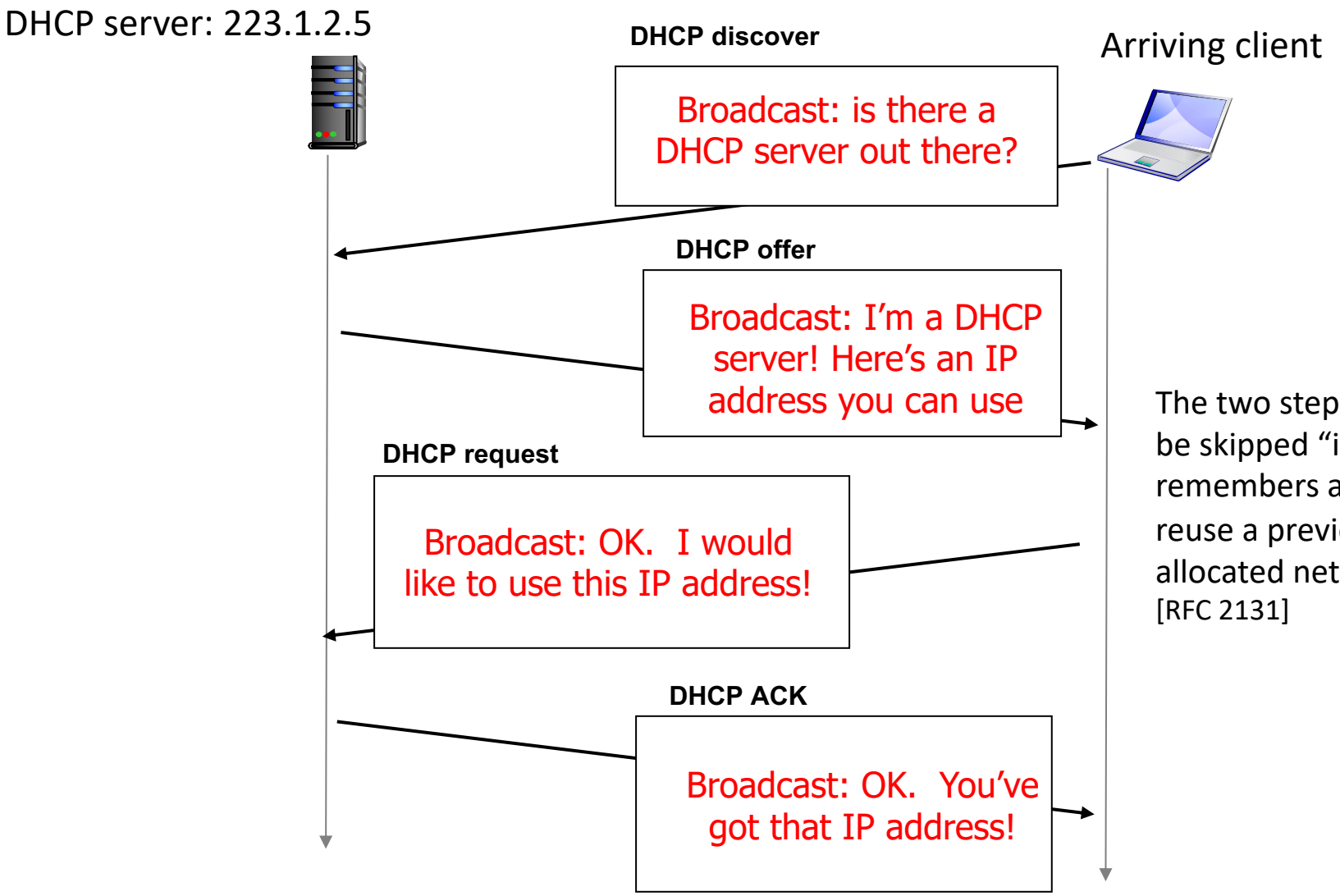
- host broadcasts DHCP discover msg [optional]
- DHCP server responds with DHCP offer msg [optional]
- host requests IP address: DHCP request msg
- DHCP server sends address: DHCP ack msg

DHCP: client-server



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

DHCP client-server scenario



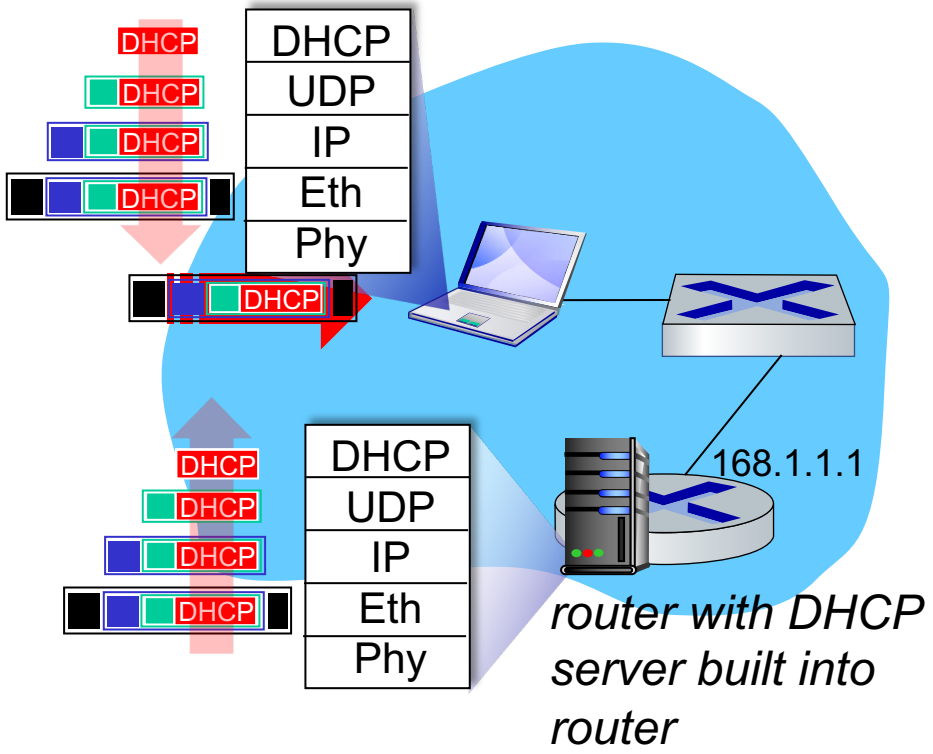
The two steps above can be skipped "if a client remembers and wishes to reuse a previously allocated network address" [RFC 2131]

DHCP: more than IP addresses

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

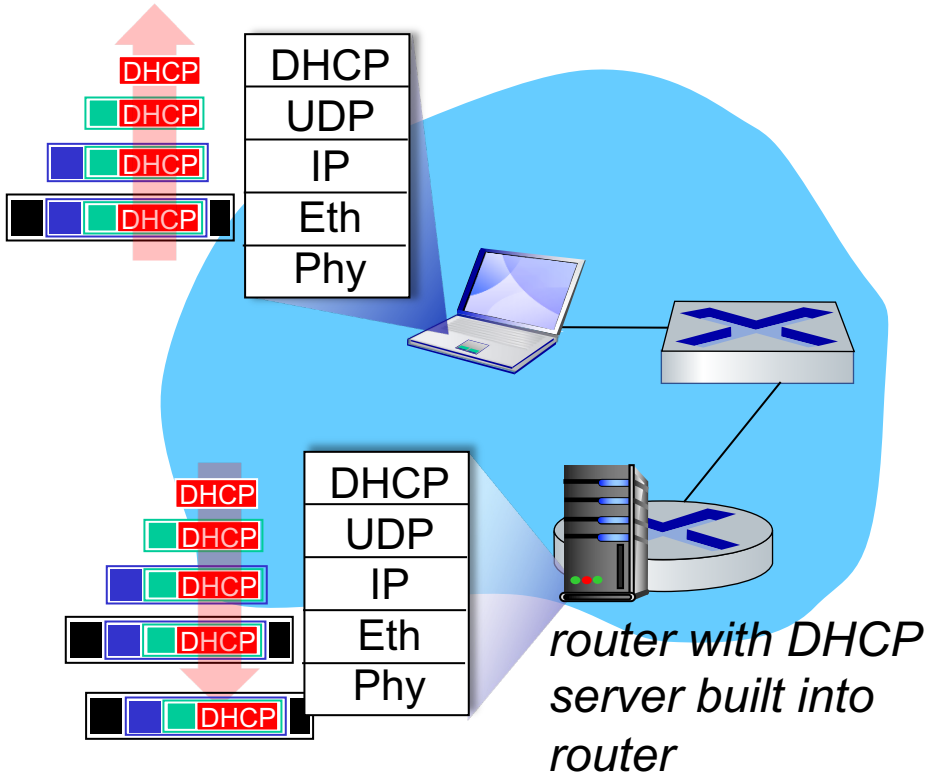
- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)

DHCP: example



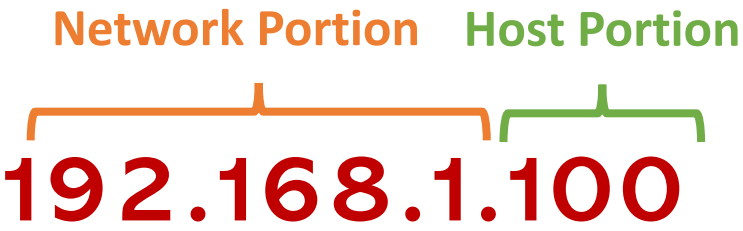
- Connecting laptop will use DHCP to get IP address, address of first-hop router, address of DNS server.
- DHCP REQUEST message encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet de-mux'ed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

DHCP: example



- DHCP 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

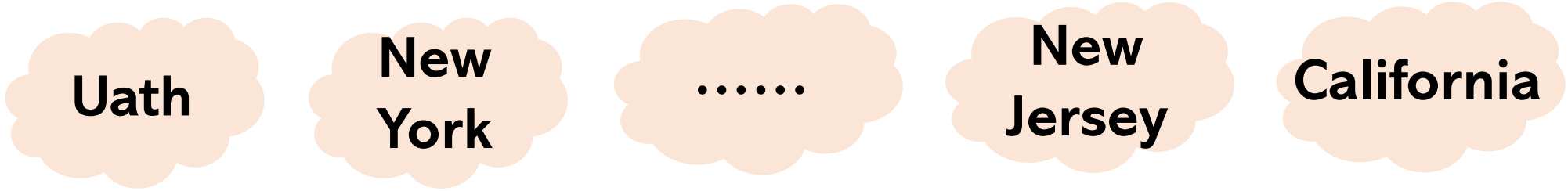
IP addresses: how to get one?



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

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

ISP's block 11001000 00010111 00010000 00000000 200.23.16.0/20



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

Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
…	…..	….	….
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

IP addressing: last words ...

Q: how does an ISP get block of addresses?

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

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

Q: are there enough 32-bit IP addresses?

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

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

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
  - IPv6

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

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
  - IPv6

Problem: Insufficient IPV4 address

Public IP VS. Private IP

A public IP is a globally visible address on the Internet

Analogy: the **street address of a building**

- It is **globally unique**
- It is **visible to the Internet**
- It is usually assigned by an **ISP**

A private IP is an address used only inside a local network

Analogy: the **room number inside the building**

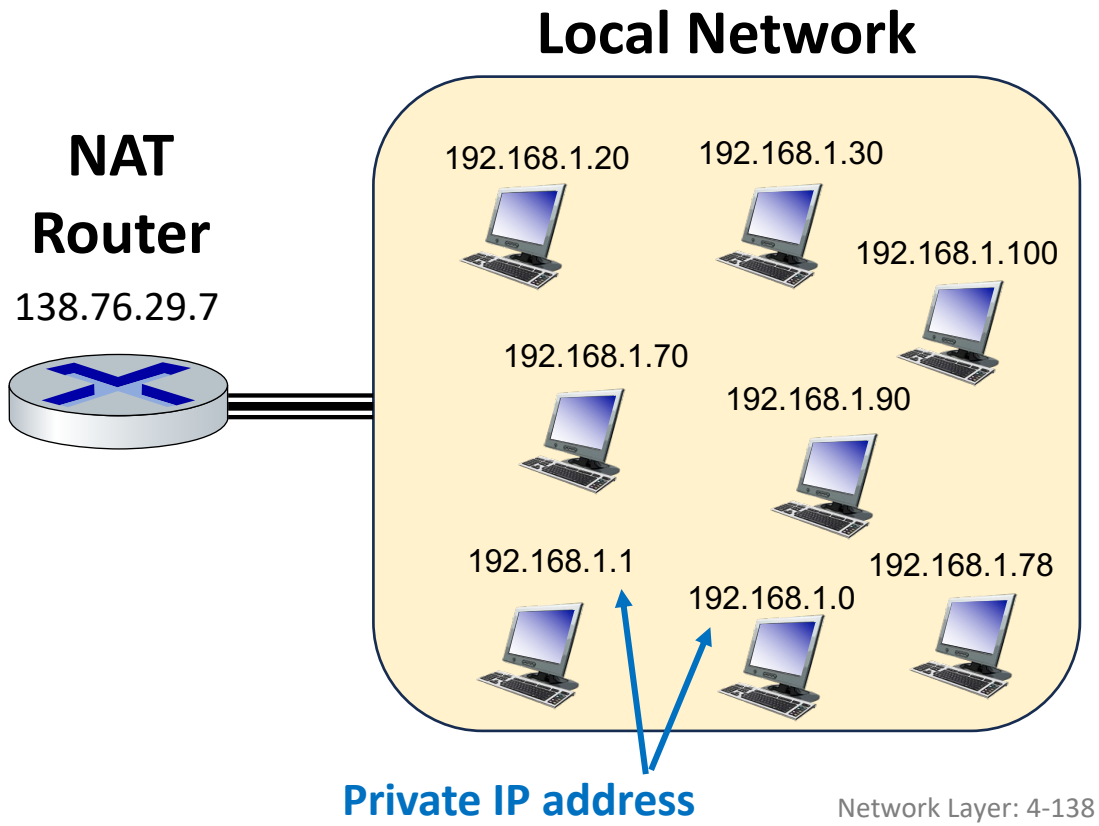
- It is used **inside local networks**
- It can be **reused in many different networks**
- It is usually assigned by the **local router via DHCP**

Three standard private IPv4 ranges:

`10.0.0.0/8` `172.16.0.0/12` `192.168.0.0/16`

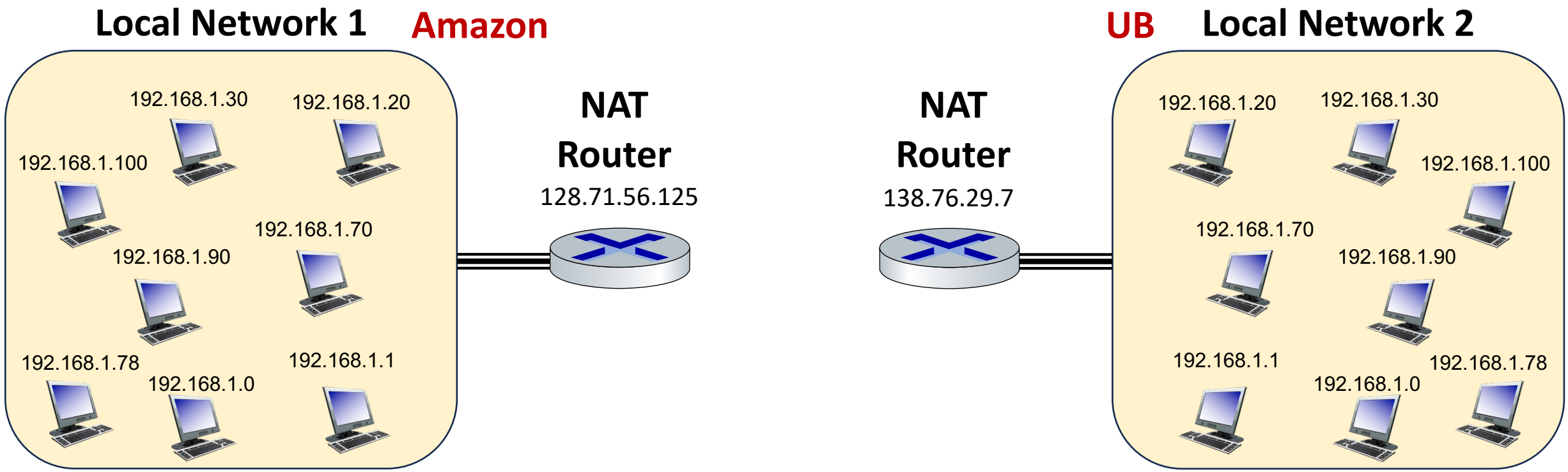
NAT: network address translation

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



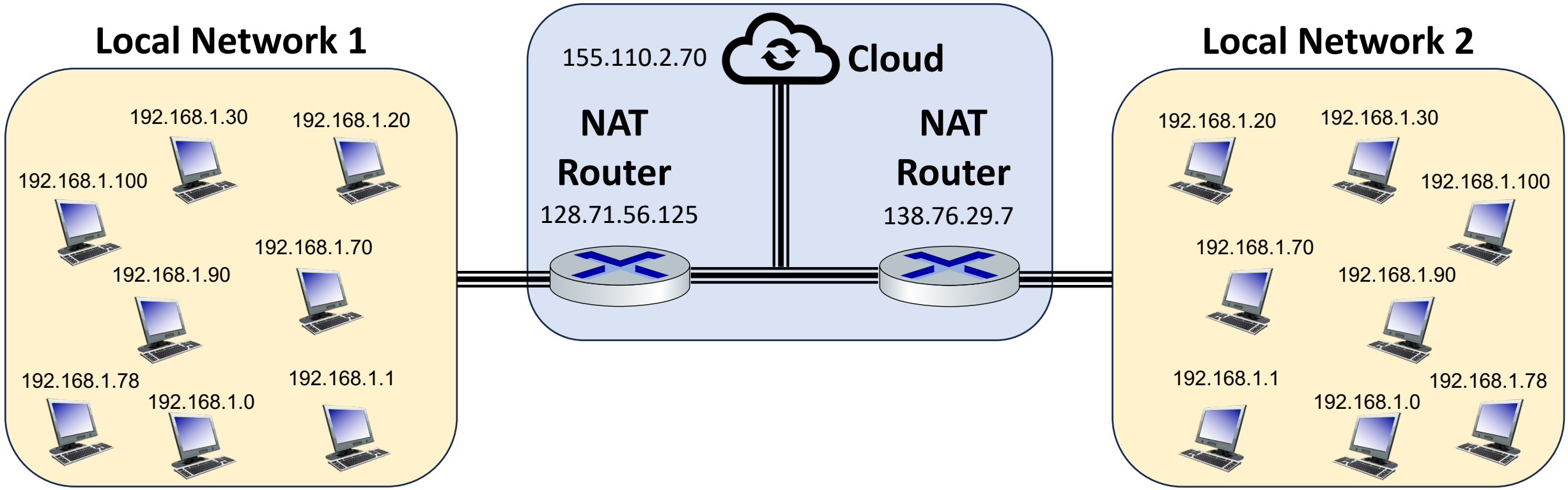
NAT: network address translation

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



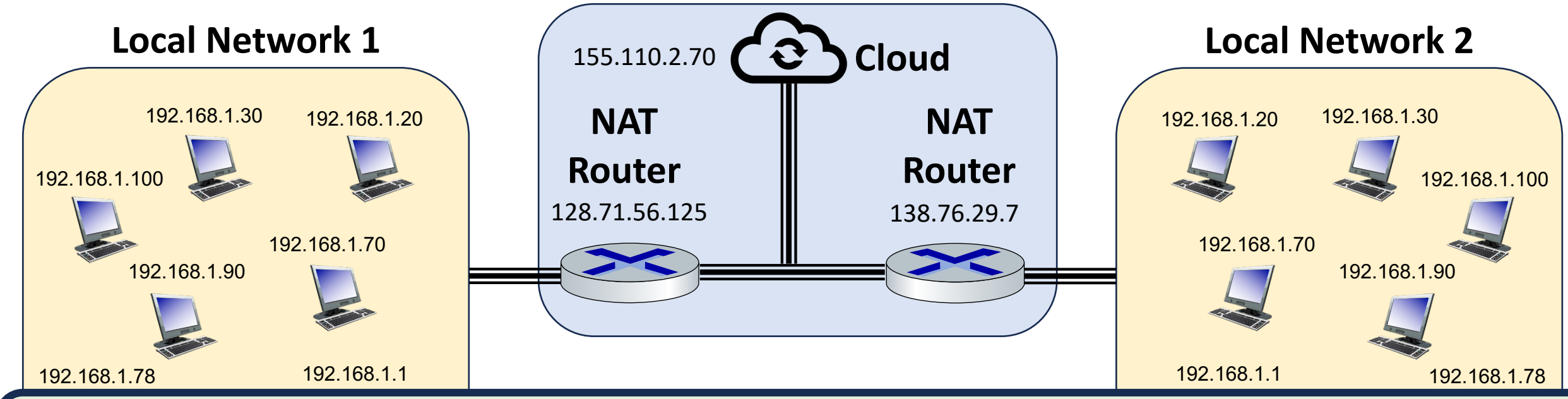
NAT: network address translation

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



NAT: network address translation

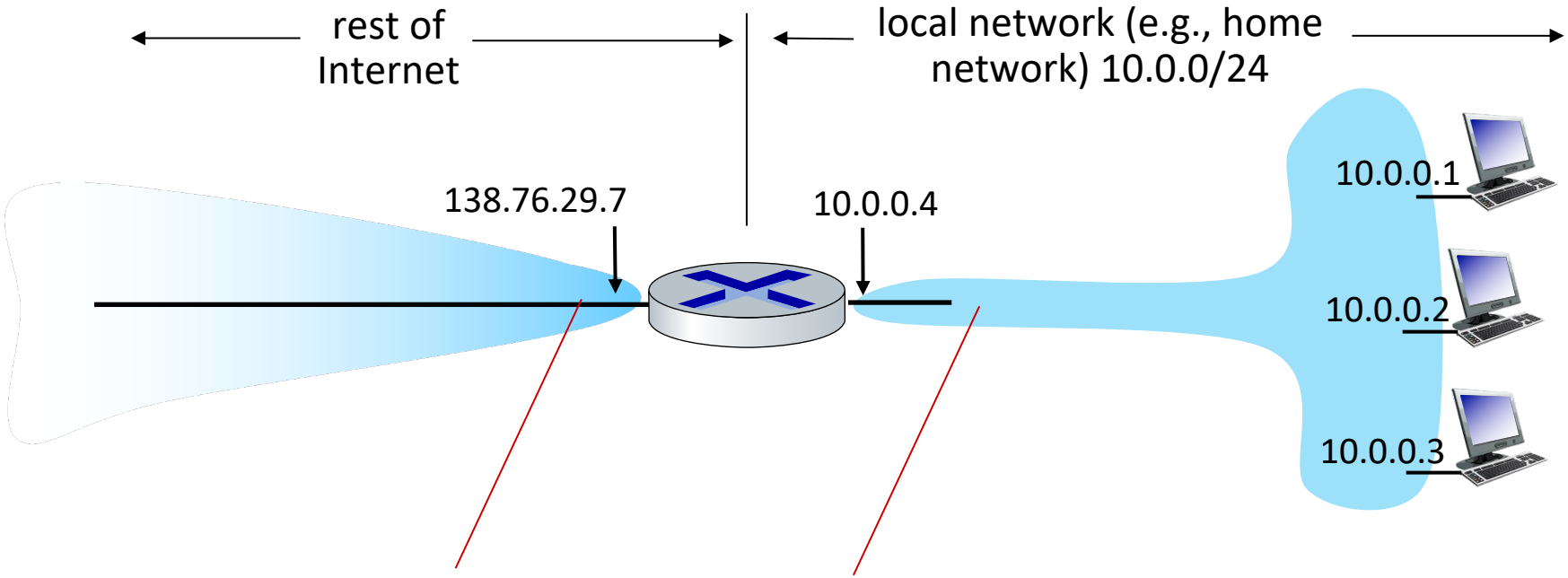
NAT: all devices in local network share just one IPv4 address as far as outside world is concerned



How to we translate the IP address from Private to Public?

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

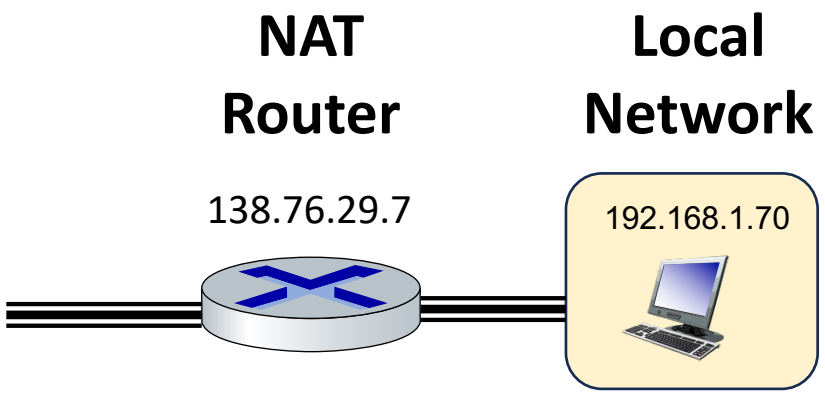
datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

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
- 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

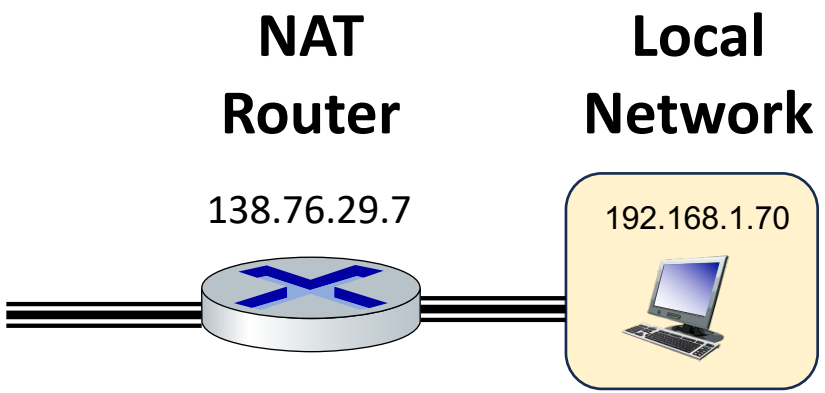
- outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  - remote clients/servers will respond using (NAT IP address, new port #) as destination address
- remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair



NAT Table: NAT IP, New Port, Src IP, Src Port

NAT: network address translation

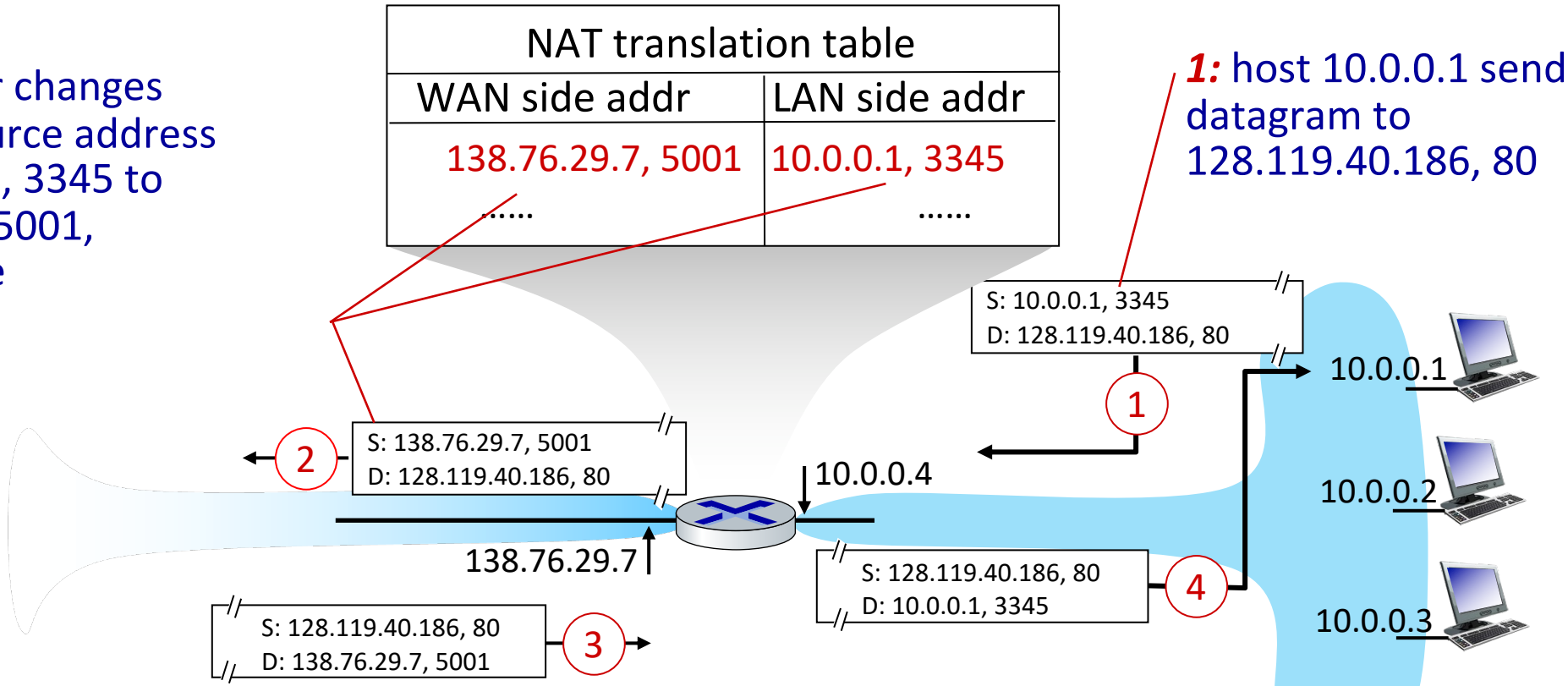
- incoming datagrams: replace (NAT IP address, new port #) in destination fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table



NAT Table: NAT IP, New Port, Src IP, Src Port

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



3: reply arrives, destination address: 138.76.29.7, 5001

NAT: network address translation

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