# IP Addressing Design Issues and Protocols

## CS2520/TELCOM2321
## Wide Area Network
## Spring Term, 2019

**Prof. Taieb Znati**
**Department Computer Science**
**Telecommunication Program**

# Outline

- Internet Address Structure
  - Classfull Addresses
  - Classeless Addresses
  - Subnetting and Supernetting
- DHCP and ARP
- Network Address Translation

# IP Addressing

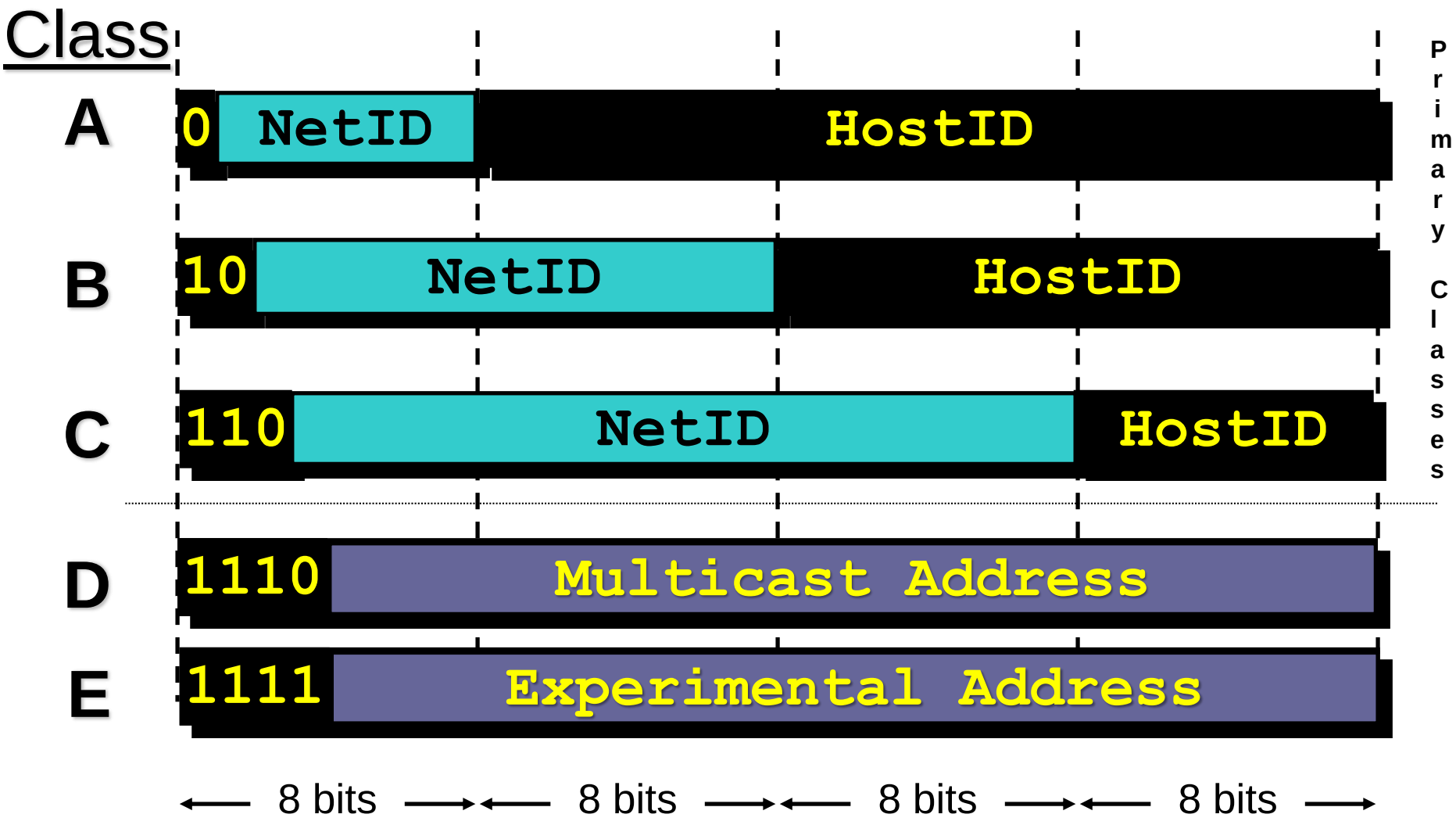
# IP Address

- Every device connected to the public Internet is assigned a unique IP address.
  - Typically addresses are assigned to internet service providers within region-based blocks,
  - IP address can often be used to identify the region or country from which a computer is connecting to the Internet.
    - An IP address can sometimes be used to show the user's general location.
- IP addresses can be assigned by an ISP statically (Static IP Address) or dynamically (Dynamic IP Address)

# IP Address

- IPv4, defined by 4 bytes (32 bits)
  - IP address represents a network interface
  - Routers, for example, are typically assigned multiple IP addresses
- Address spaces
  - 0.0.0.0 ~ 255.255.255.255
  - $2^{32}$ = 4,294,967,296 hosts

# Classful IP Address Format



# Class A Networks

- Class A networks are referred to as "/8" networks, since they have an 8-bit network prefix
  - "/8" address block contains 2,147,483,648 individual addresses ($2^{31}$), or 50% of the total IPv4 address space, 4,294,967,296 ($2^{32}$)
- Max of 126 ($2^7 - 2$) "/8" networks can be defined
  - 0.0.0.0 and 127.0.0.0 are reserved
- Each "/8" supports 16,777,214 ($2^{24} - 2$) hosts per network
  - All-0's and All-1's numbers cannot be assigned to hosts

# Class B Networks

- Class B networks are referred to as "/16", since they have a 16-bit network prefix
  - "/16" address block contains 1,073,741,824 individual addresses ($2^{30}$), or 25% of the total IPv4 address space, 4,294,967,296 ( $2^{32}$ )
- Max of 16,384 ($2^{14}$) "/16" networks can be defined
- Each "/16"network supports 65,534 ($2^{16} - 2$) hosts per network
  - All-0's and All-1's numbers cannot be assigned to hosts

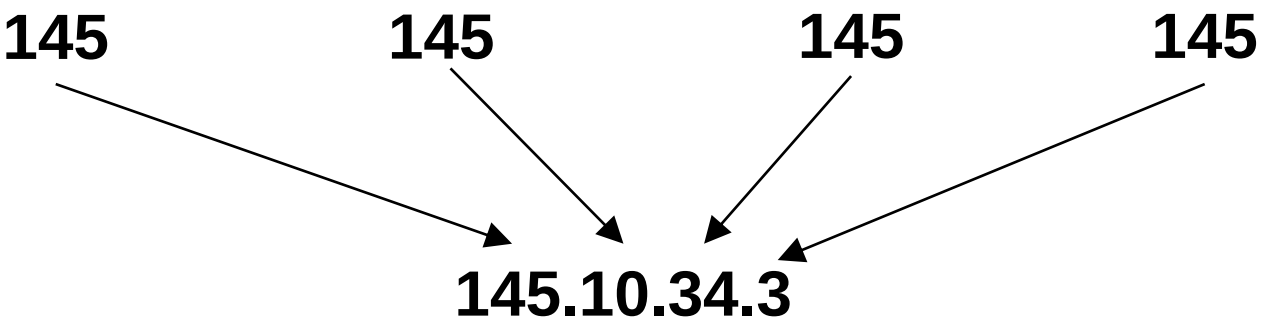
# Class C Networks

- Class C networks are referred to as "/24", since they have an 24-bit network prefix
  - "/24" address block contains 536,872,912 individual addresses ($2^{29}$), or 12.5% of the total IPv4 address space, 4,294,967,296 ($2^{32}$)
- Max of 2,097,152 ($2^{21}$ ) "/24" networks can be defined
  - 0.0.0.0 and 127.0.0.0 are reserved
- Each "/24"supports 254 ($2^{8}$- 2) hosts
  - All-0's and All-1's numbers cannot be assigned to hosts

# IP Addresses

- **IP Address *dotted decimal* notation**
  - **It divides the 32-bit IP address into 4 byte fields and specifies each byte independently as a decimal number with the fields separated by dots**

**10 010001 00001010 00100010 00000011**



# Dotted Decimal Ranges

| Address Class | Dotted-Decimal Notation Ranges |
|---|---|
| A(/8 prefixes) | 1.xxx,xxx.xxx trough 126.xxx.xxx.xxx |
| B(/16 prefixes | 128.0.xxx.xxx through 191.255.xxx.xxx |
| C(/24 prefixes) | 192.0.0.xxx through 223.255.255.xxx |

# Reserved IP Addresses

- 0.0.0.0
  - Default route
- 127.0.0.1
  - Loopback IP address
  - Test  IPC on local machine
- All bits are 0 in host number
  - Denote this network
- All bits are 1 in host number
  - Broadcast address in this network
- Private IP addresses
  - 10.xxx.xxx.xxx, 192.168.xxx.xxx

# Unforeseen Limitation of Classfull Addressing

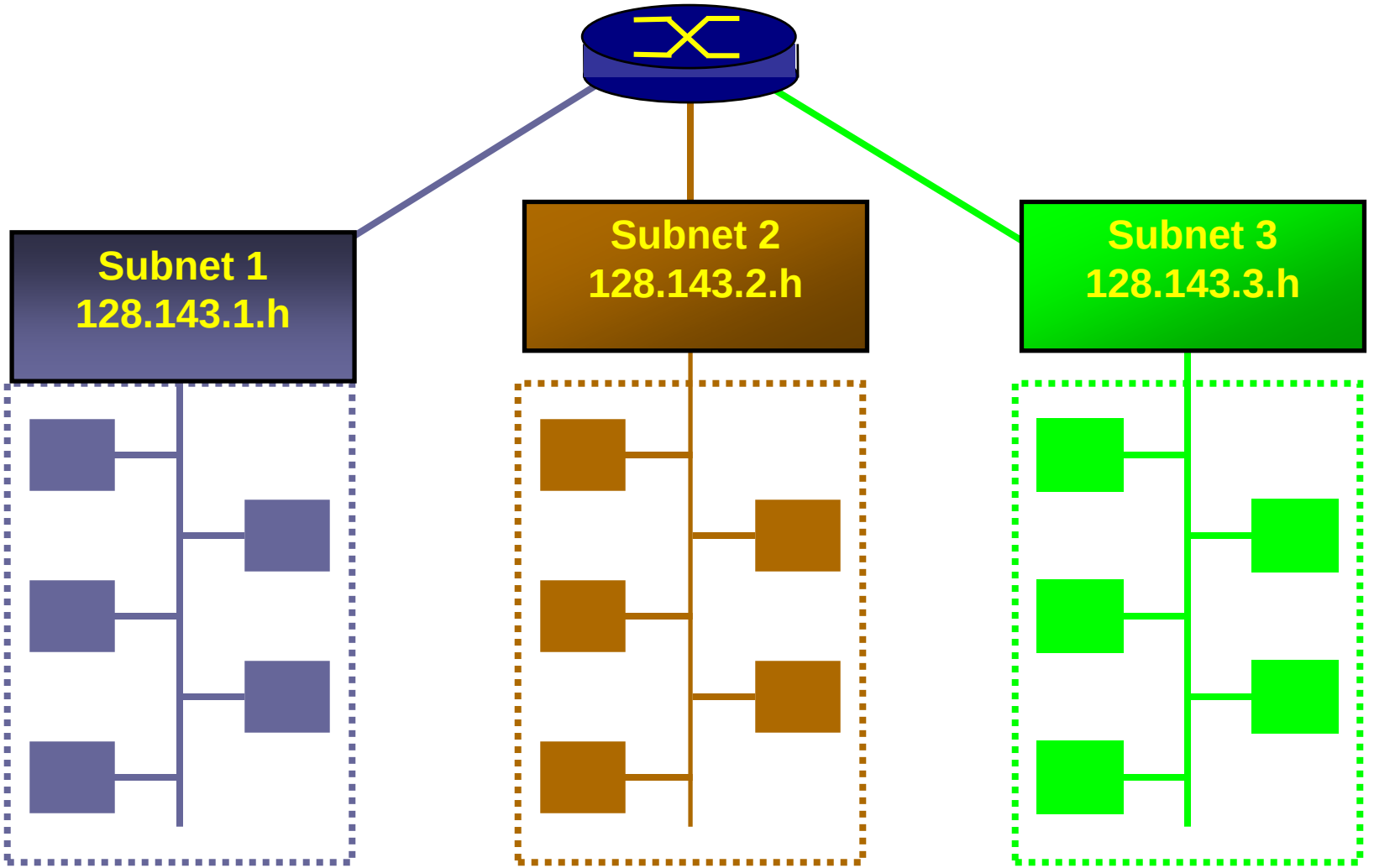
- Addresses were allocated to organizations based their requests rather than actual need
- The decision to standardize on a 32-bit address space did not foresee a network of things
- Classes were easy to understand and implement but did not foster efficient allocation
  - "/24" is too small and "/16" is too large
  - Allocating a "16" to an organization that has several hundreds sites is wasteful and depletes the address space
  - Allocating several "/24"s increases the size of the routing table

# Subnet Addresses

- Subnetting is introduced mainly to address depletion and routing table inflation.
- Three-Level Hierarchy
  - The number of subnets must be a power of 2


- The subnet structure of a network is never visible outside the local network
  - This limits considerably the expansion of the routing table

# Sub-netting

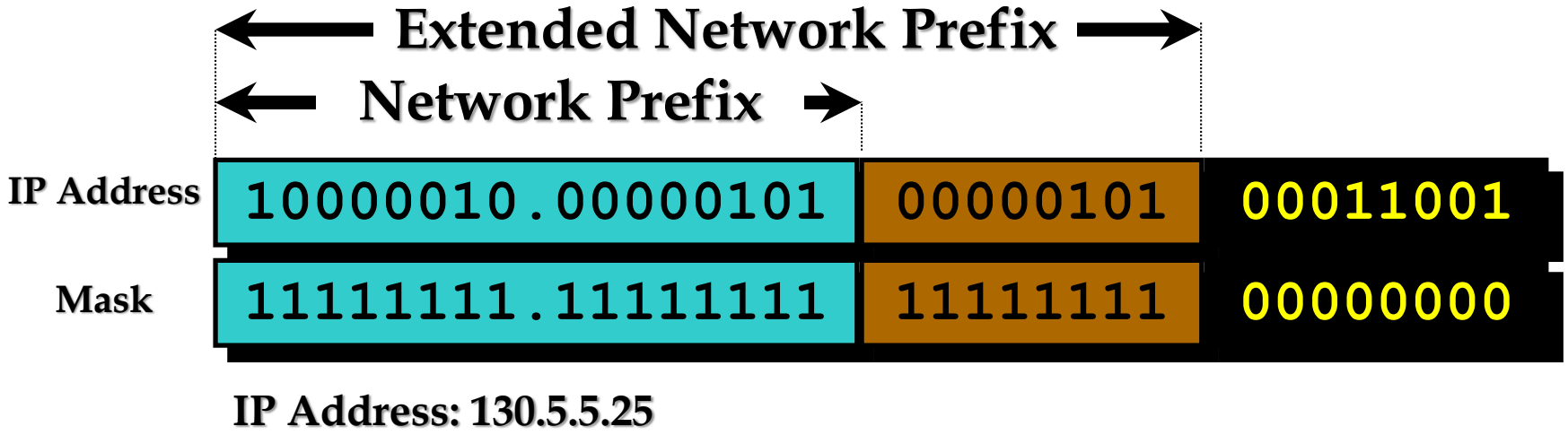


# Extended Network Prefix

- Internet routers use only the network-prefix of the destination address to route traffic to a subnetted environment
  - Routers within the subnetted environment use the extended network prefix to route traffic

# Subnet Masks

- The extended-network prefix is identified by a subnet mask
  - A bit of the subnet mask is set to 1 if the corresponding bit in the IP address must be considered as part of the extended network prefix



# Logical Bitwise AND Operation

**Class B address:** 140.179.220.200

Subnet Mask: 255.255.0.0

**Binary representation:**

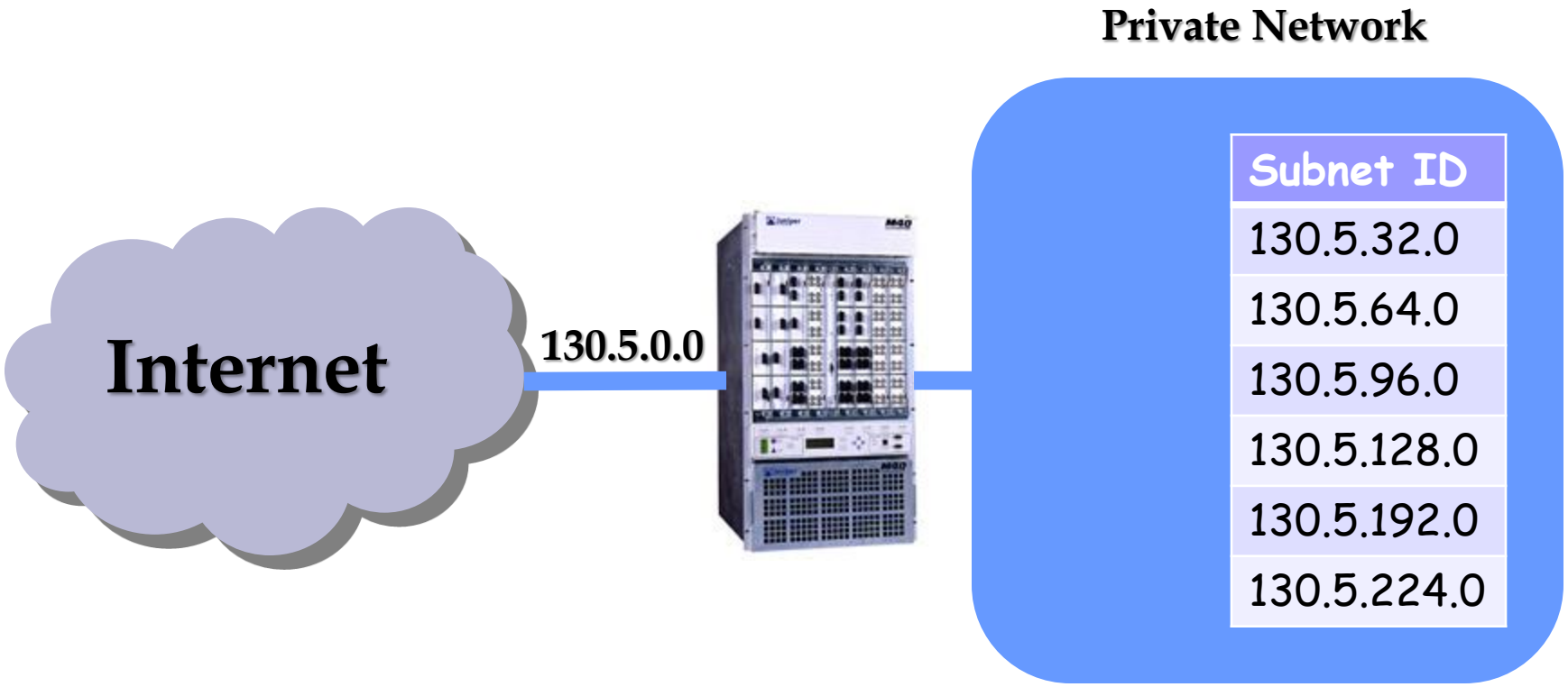
10001100.10110011.11110000.11001000

11111111.11111111.00000000.00000000

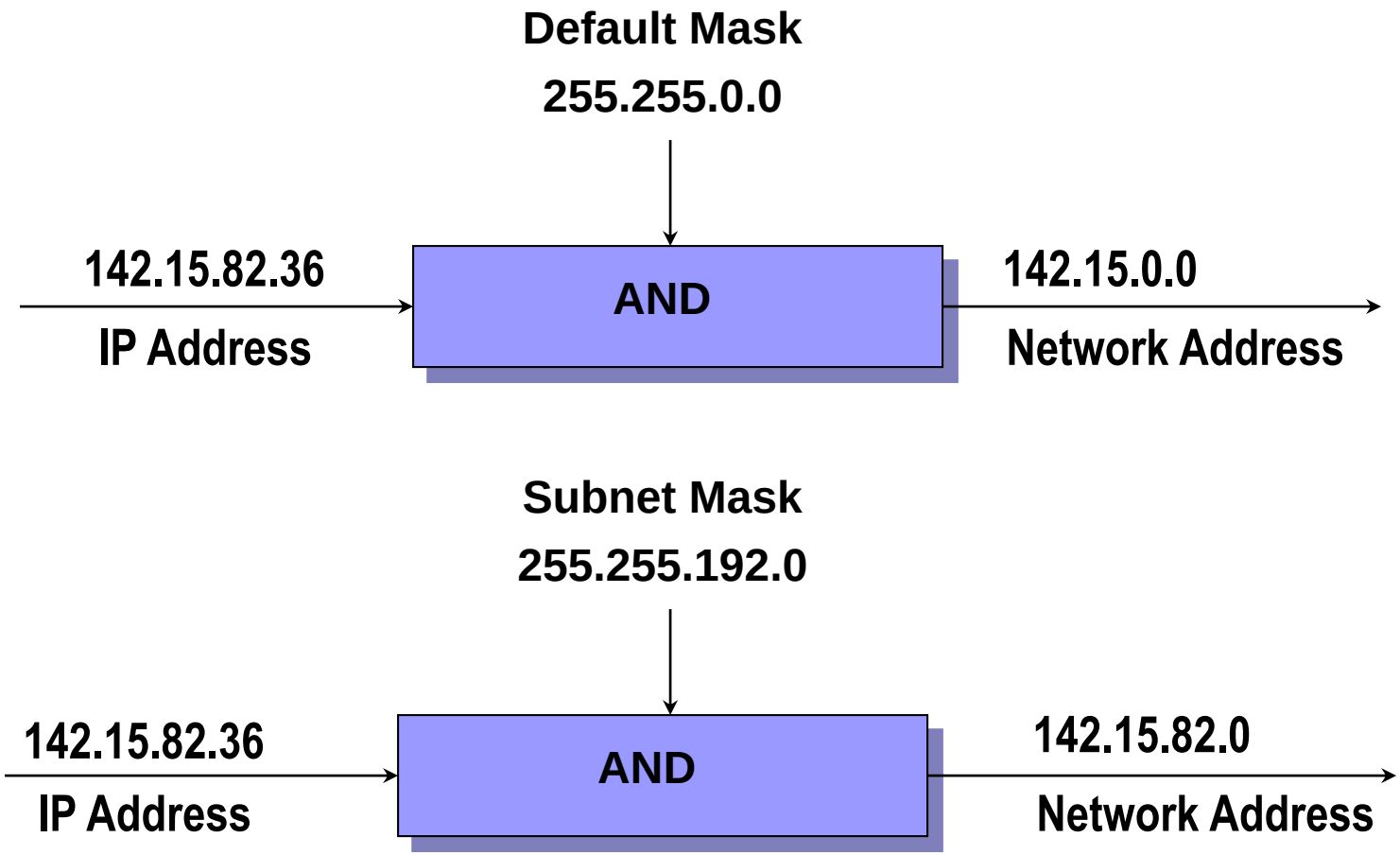
10001100.10110011.00000000.00000000

Network Address **is** *140.179.0.0*

# Subnetting – Prefix Advertisement



# Default Mask and Subnet Mask



# Subnetting - Example

- A company is granted the site address 201.70.64.0: Class C address.
  - The number of 1s in the default mask is 24
- The company requires six subnets
  - How can such a network be designed?

# Subnetting Solution

- **The number 6 is not a power of 2.**
  - **The next power of 2 is 8 ($2^3$)**
  - **3 more bits are needed for the subnet mask**
    - **The total number of 1s in the subnet mask is 27**
      - **27 = 24 (original) + 3(added)**

# Subnetting Solution

- **Subnet Mask**
  - 11111111 11111111 11111111 11100000
- **In Decimal Dotted Notation**
  - 255.255.255.224
    - The number of subnets is 8.
    - The number of addresses in each subnet is $2^5$ or 32
      - Address 00000 and address 11111 are reserved

# Address Management Challenges And Solution

- Large ISPs:
  - They own class A address blocks
  - Makes it hard to organize IP addresses
- Small enterprises
  - Own a number of class C address blocks
  - Makes it hard to manage so many prefixes
- Two approaches
  - Variable Length Subnet Masks (VLSM)
  - Classless Inter-Domain Routing (CIDR)

# CIDR

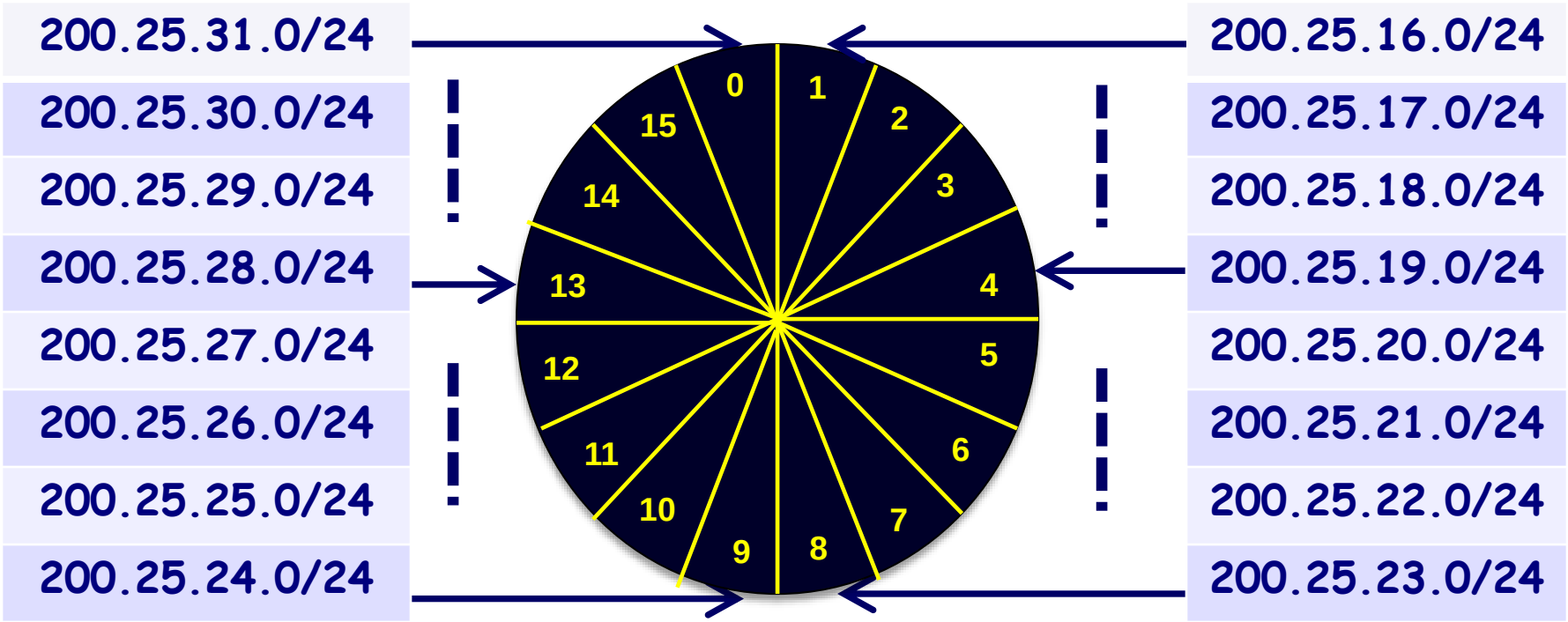
## IP Addressing

# Classless Inter-Domain Routing (CIDR)

- CIDR was designed to address the ROADs problem
- No concept of address classes
- Prefixes are not restricted to /8, /16 and /24
  - Prefixes could be any length from 1 to 32
    - 1<= masklength <=32
- As a result, CIDR supports the deployment of arbitrarily sized networks rather than the standard 8-bit, 16-bit or 24-bit networks numbers
  - Regardless of the class of the IP address, a network with 20 bits of network-number and 12 bits of host number is advertised with a 20-bit prefix length

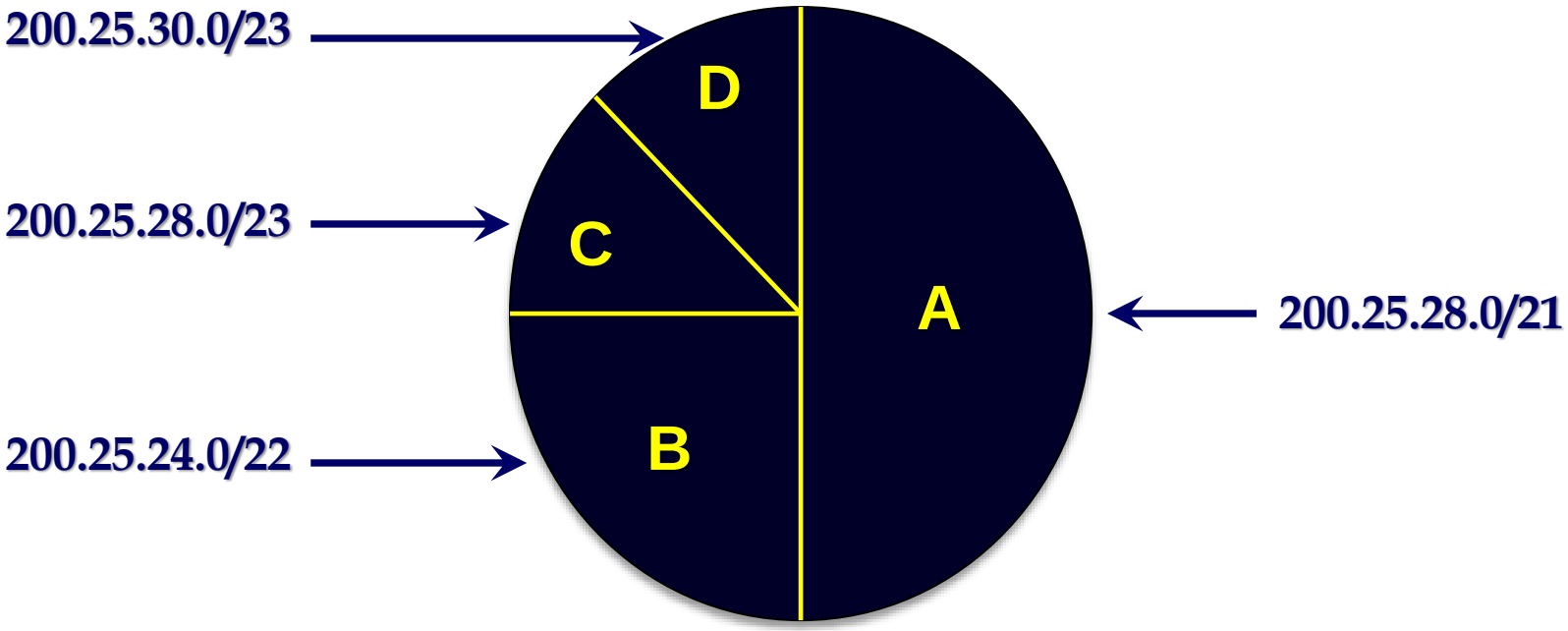
# Classfull Subnetting and Address Allocation

- ISP owns the address block 200.25.0.0/16 and wants to allocate the 20.25.16.0/20 address block
  - In a classful environment, it can only be cut into 16 equal-size segments



# Classless Subnetting and Address Allocation

- Address slices do not have to be of equal size
  - Address block 200.25.16.0/20



# CIDR Address Allocation

- Divide address block 200.25.16.0/20 into 2 equal slices
  - Each block represents one-half of the address space – 2,048 IP addresses
- ISP Block: 11001000.00011001.00010000.00000000 200.25.16.0/20
- Org A: 11001000.00011001.00010000.00000000 200.25.16.0/21
- Reserved: 11001000.00011001.00011000.00000000 200.25.24.0/21

# CIDR Address Allocation

- **Divide reserved address block 200.25.24.0/21 into 2 equal slices**
  - **Each block represents one-half of the address space – 1,024 IP addresses**

| | | |
|---|---|---|
| **Reserved:** | **11001000.00011001.00011000.00000000** | **200.25.24.0/21** |
| **Org B:** | **11001000.00011001.00011000.00000000** | **200.25.24.0/22** |
| **Reserved:** | **11001000.00011001.00011100.00000000** | **200.25.28.0/22** |

# CIDR Address Allocation

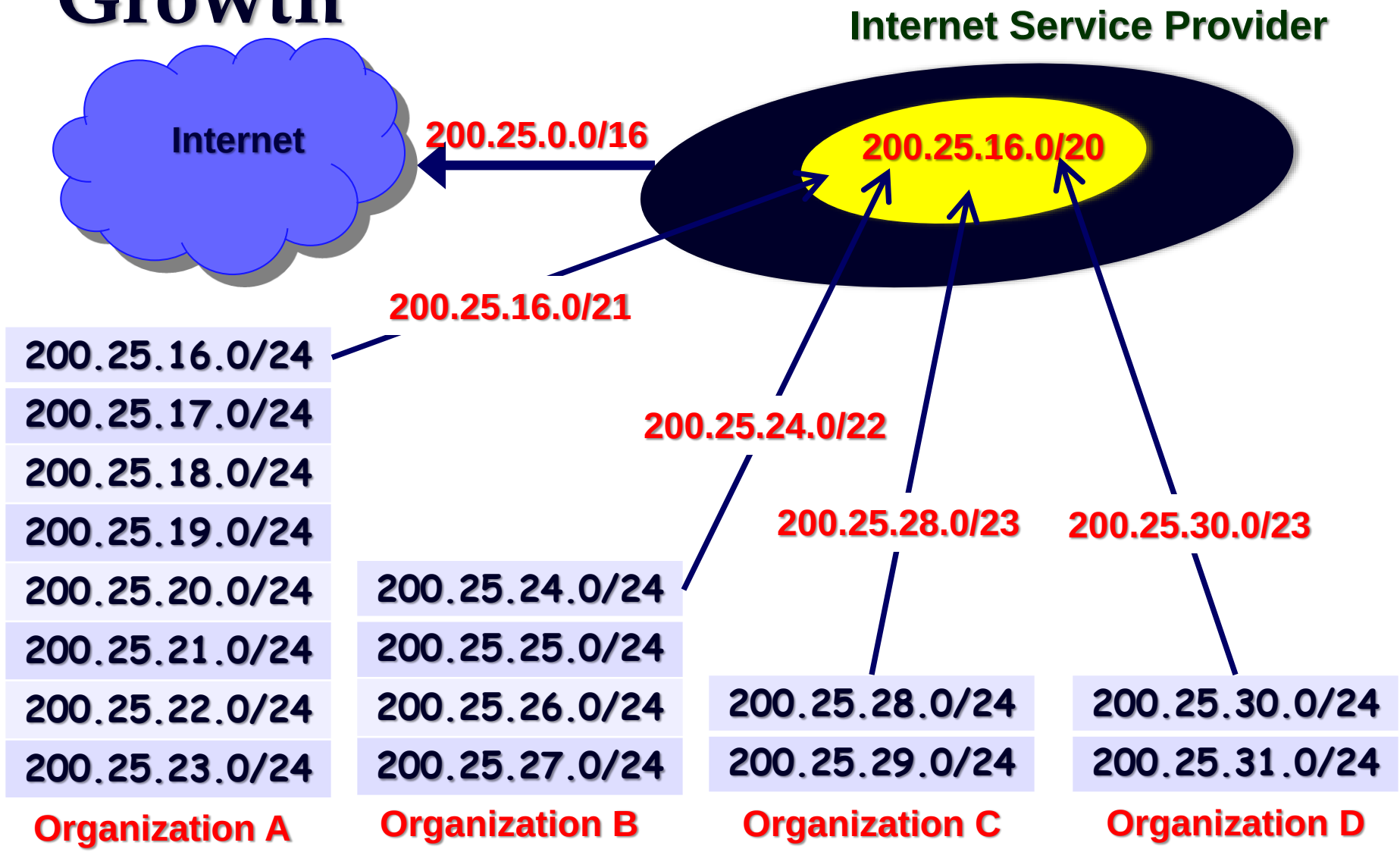
- Divide reserved address block 200.25.28.0/22 into 2 equal slices
  - Each block represents one-half of the address space – 512 IP addresses

| | | |
|---|---|---|
| Reserved: | 11001000.00011001.00011100.00000000 | 200.25.28.0/22 |
| Org C: | 11001000.00011001.00011100.00000000 | 200.25.28.0/23 |
| Org D: | 11001000.00011001.00011110.00000000 | 200.25.30.0/23 |

# Controlling Routing Table Growth

- CIDR requires that the Internet be divided into addressing domains
  - Within a domain, detailed information is available about all networks that reside in the domain
  - Outside of an addressing domain, only the common network prefix is advertised
- This allows single routing table entry to specify a route to many individual network addresses

# CIDR – Controlling Routing Table Growth



# CIDR Forwarding Algorithm

- All routers must implement a consistent forwarding algorithm based on the "longest match" algorithm.
  - A route with a longer extended-network-prefix describes a smaller set of destinations than the same route with a shorter extended-network-prefix.
  - A route with a longer extended-network-prefix is said to be "more specific" while a route with a shorter extended-network-prefix is said to be "less specific."
- Routers must use the route with the longest matching network-prefix (most specific matching route) when forwarding traffic.

# Classless Routing
# Longest Prefix Match -- Example

- Assume a packet's destination IP address is 11.1.2.5 and there are three network prefixes in the routing table (11.1.2.0/24, 11.1.0.0/16, and 11.0.0.0/8)

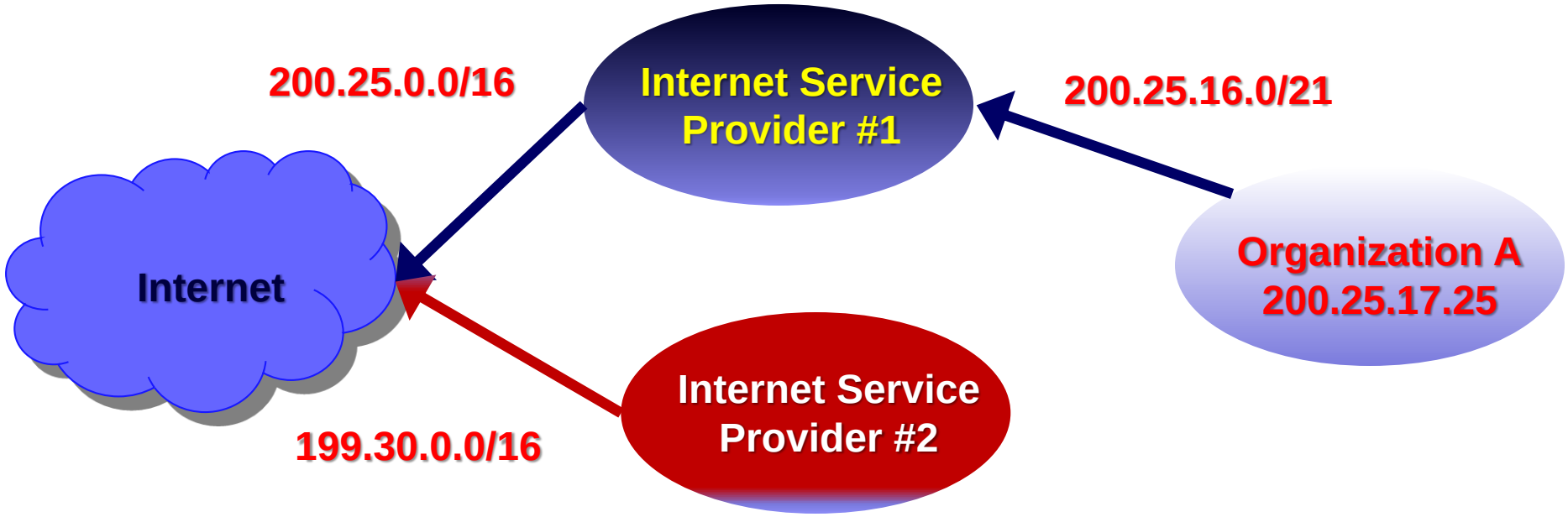
- Destination 11.1.2.5 = 00001011.00000001.00000010.00000101

- Route #1 11.1.2.0/24 = 00001011.00000001.00000010.00000000*
- Route #2 11.1.0.0/16 = 00001011.00000001.00000000.00000000
- Route #3 11.0.0.0/8 = 00001011.00000000.00000000.00000000

- Router would select the route to 11.1.2.0/24.
  - The 11.1.2.0/24 route is selected because its prefix has the greatest number of corresponding bits in the Destination IP address of the packet.

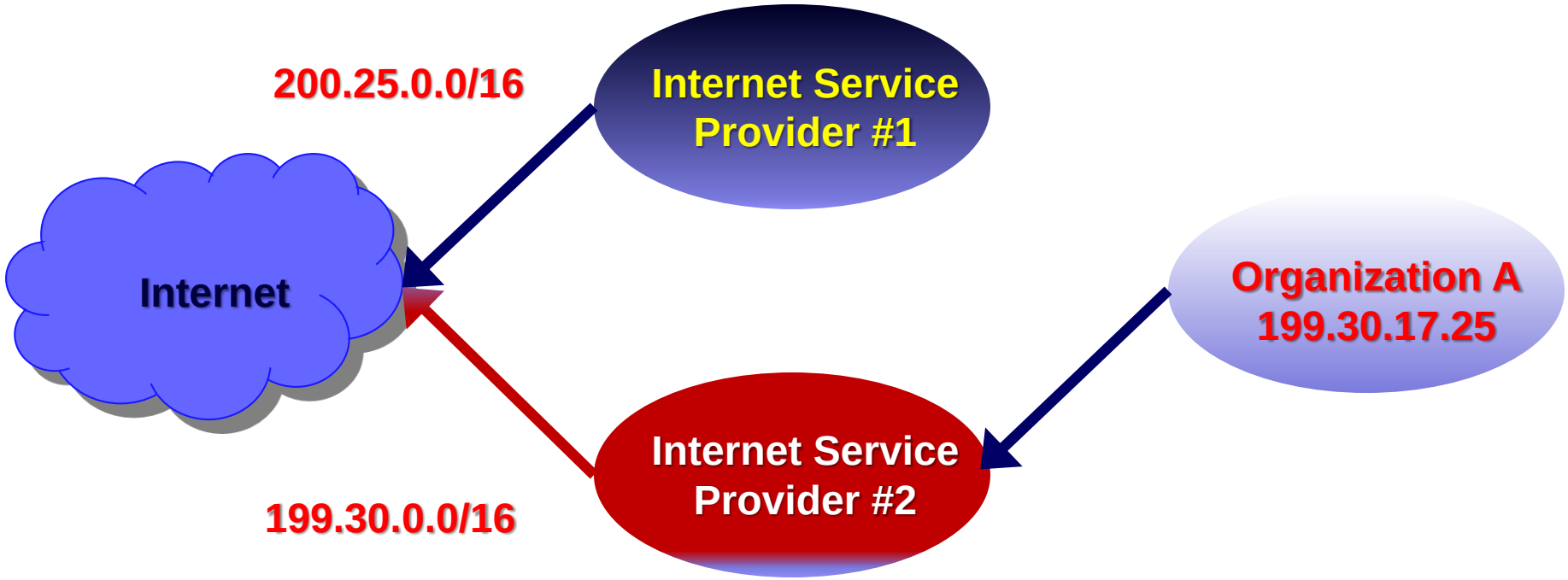
# Routing in Classless Environments

- All A's routes are part of ISP #1's address block, and routes to A are implicitly aggregated via ISP #1's aggregated announcements to the Internet
  - A's 8 networks are hidden behind a single routing advertisement
  - Using the longest match forwarding algorithm, Internet routers will route traffic to host 200.25.17.25 to ISP #1
  - ISP #1 routes the traffic to A



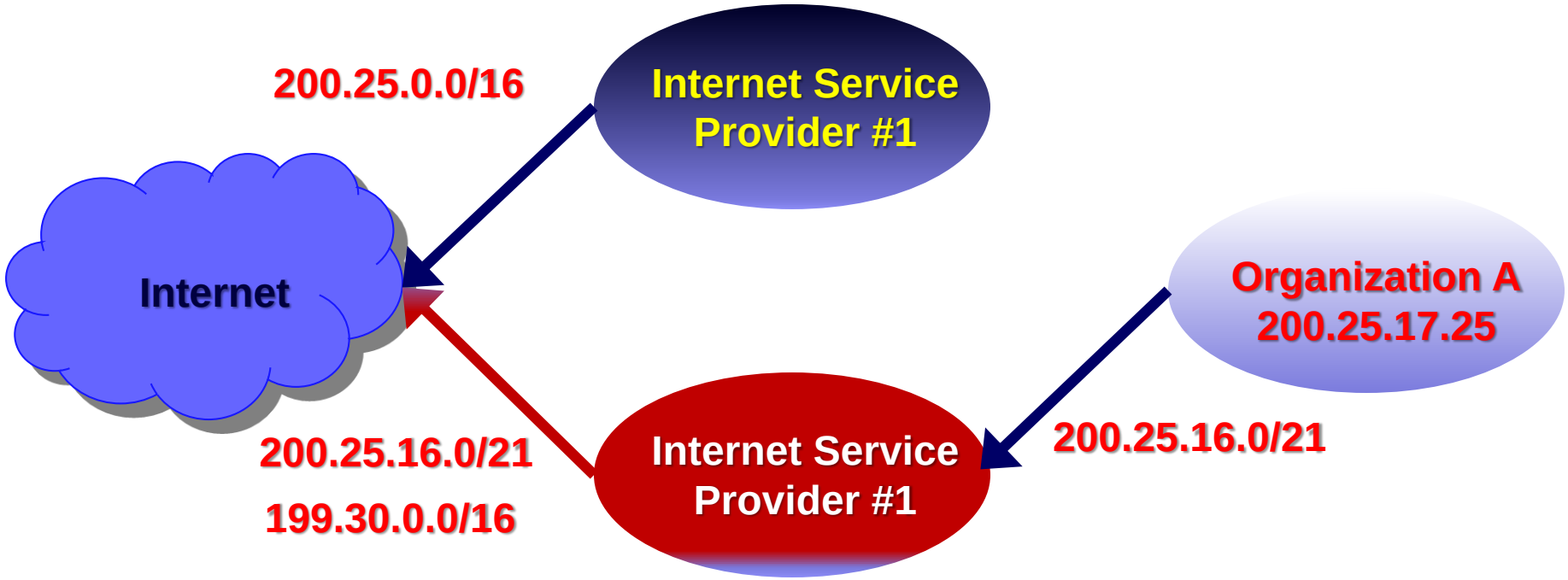
# Routing in Classless Environments

- Organization A changes its network provider to ISP #2
  - Best scenario – obtain a block of ISP #2's address space and renumber
    - A's network will remain hidden
  - In practice, renumbering is hard and costly



# Routing in Classless Environments

- A more practical solution:
  - Retain ownership of its original address space (ISP #1)
  - Have ISP #2 advertise an "exception", more specifically



# CIDR – Summary

- CIDR allows efficient allocation of the IPv4 address space
  - Divide old class A IP address into several reasonably sized IP prefixes
    - 3.0.0.0/8 ➔ 3.1.10.0/24, 3.2.96.0/20, ……
  - Aggregate several class C IP addresses into one reasonably sized prefix
    - 202.64.28.0/24, 202.64.29.0/24
      - 202.64.28.0/23
    - 203.72.174.0/24, 203.72.175.0/24, 203.72.176.0/24, 203.72.177.0/24
      - 203.72.172.0/22
- Routing uses the "Longest Prefix Match"
  - `The prefix 128.119.0.0/16 covers 128.119.96.0/20`
    - `The latter is more precise than the former`

# Internet Addressing – Address Allocation, Resolution and Translation

# DHCP, ARP, NAT

# Why Separating IP and MAC Addresses?

- LANs are not designed for different network protocols
  - IP, IPX, Appletalk, X.25, …)
- Different LANs have different addressing schemes
- Mobile host cannot be assigned static network addresses, as they move to different locations
  - IP addresses depend on host's position in topology
    - New IP address must be assigned to a mobile host, baed on its current location

# IP Address Allocation, Configuration and Mapping

- Who assigns names, IP addresses and MAC addresses?
  - Naming Assignment
  - IP address Assignment
- How are hosts configured with their IP addresses?
  - BootP and HDCP
- How IP addresses are resolved into MAC addresses?
  - Address Resolution Protocol

# Assigning Identifiers for the Internet

- Who assigns institutions their domain names?
- Who assigns network prefix?
- Who assigns “well-know” port numbers?
- The functions were originally assumed by Internet Assigned Number Authority(IANA).
  - IANA is one of the Internet's oldest institutions, with its activities dating back to the 1970s.
  - IANA used to be managed by Jon Postel at ISI

# The IANA Function

- **IANA coordinates some of the key elements that keep the Internet running smoothly.**
  - **IANA allocates and maintains unique codes and numbering systems that are used in the technical standards ("protocols") that drive the Internet.**
- **Today it is operated by the Internet Corporation for Assigned Names and Numbers, an international non-profit organization set up by the Internet community to help coordinate IANA's areas of responsibilities.**

# IANA Various Activities

- **IANA's various activities can be broadly grouped in to three categories:**
  - **Domain Names – IANA manages the DNS root, the .int and .arpa domains, and an IDN practices resource.**
    - **To help foster the deployment of Internationalized Domain Names (IDNs), IANA provides an "informative" repository of "IDN tables" which document the permissible characters for different languages and scripts provided for registration by different top-level domain registries.**
  - **Number Resources – IANA coordinates the global pool of IP and AS numbers, providing them to Regional Internet Registries.**
  - **Protocol Assignments – IANA, in conjunction with standards bodies, manages Internet protocols' numbering systems.**

# Host Bootstrapping

## DYNAMIC HOST CONFIGURATION PROTOCOL

# Dynamic Host Configuration Protocol

- DHCP is the preferred mechanism for dynamic assignment of IP addresses
- Designed in 1993, as an extension of BOOTP
  - DHCP can interoperate with BOOTP clients
    - Uses port numbers as BOOTP
- DHCP Extensions:
  - Support for temporary allocation ("leases") of IP addresses
  - DHCP client can acquire all IP configuration parameters

# DHCP Packet Format

| op (1 byte) | htype (1 byte) | hlen (1 byte) | hops (1 byte) |
|---|---|---|---|
| xid (4 bytes) | | | |
| secs (2 bytes) | | flags (2 bytes) | |
| ciaddr (4 bytes) | | | |
| yiaddr (4 bytes) | | | |
| siaddr (4 bytes) | | | |
| giaddr (4 bytes) | | | |
| chaddr (16 bytes) | | | |
| sname (64 bytes) | | | |
| file (128 bytes) | | | |
| options (variable) | | | |

# DHCP Packet Fields

- op – Message Type
  - 1 = REQUEST: Client to server
  - 2 = REPLY:  Server to client
- htype – *Hardware Address Type*
  - 1 = 100Mbps Ethernet
- hlen – *Hardware Address Length* (in bytes)
  - 6 (bytes) for Ethernet.
- hops – Hops taken so far
  - Client sets to 0 – Optionally used by relay agents when booting via relay agent.
- xid – *Transaction Id*.  Unique number to associate messages.
  - Random number chosen by the client.
- secs – Number of seconds elapsed since client began address acquisition/renewal
  - Filled in by the client.

# DHCP Packet Fields

- **Flags: 16 bits**
  - **B: Broadcast Flag (1 Bit): 1 = Broadcast**
    - **0 = Unicast**
  - **Must Be Zero (15 Bits): For future expansion.**
    - **zero!**
- **ciaddr – *Client IP Address***
  - **Only filled in if client is in BOUND, RENEW or REBINDING states.**
- **yiaddr – *Your IP Address***
  - **The IP Address that the server gives to the client.**

# DHCP Packet Fields

- **siaddr – *Server IP Address***
  - **Address of next server to use. Set by server in DHCPOFFER and DHCPACK.**
- **giaddr – *Gateway/* Relay Agent *IP Address*.**
  - **Used if indirect connection to the DHCP Server.**
- **chaddr – *Client Hardware Address***
  - **The Ethernet/MAC Address of the client.**
- **sname – *Server Name***
  - **Optional server name. Null terminated string.**
- **file – Boot File Name**
  - **Null terminated string.**
- **options – Various optional fields.**

# Options Fields

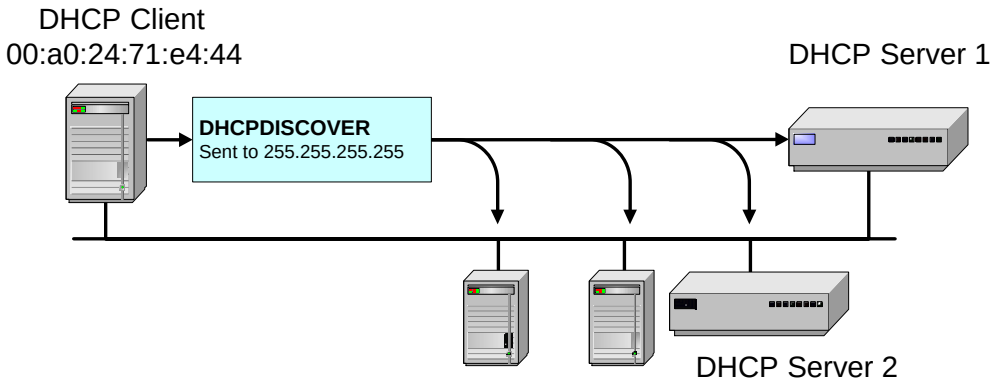
- **Message Type**
  - **Present in most real implementations of DHCP. Makes packet type easier to identify.**
    - **1 = DHCPDISCOVER**
    - **2 = DHCPOFFER**
    - **3 = DHCPREQUEST**
    - **4 = DHCPACK**
    - **5 = DHCPNAK**
- **….**

# DHCP Basic Operations

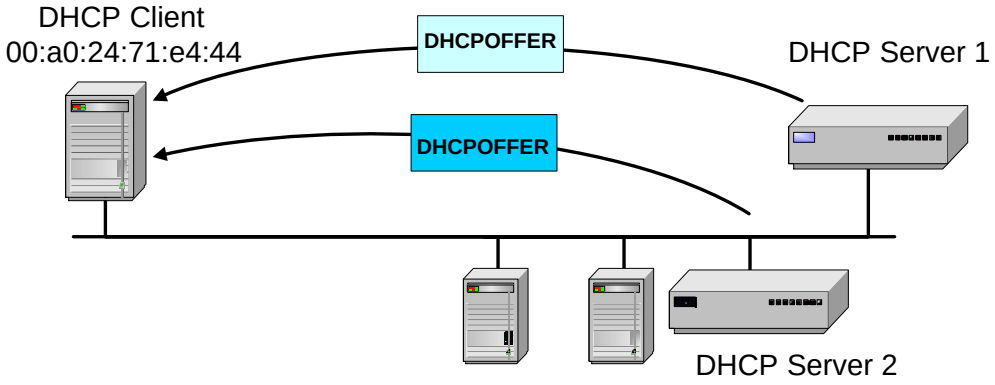
- To obtain an IP Address, a client issue a broadcast packets, with Broadcast (B) Flag set.
  - DHCPDISCOVER – Client may not know which DHCP servers are currently operational.
  - DHCPREQUEST – Broadcast in response to one or more DHCPOFFERs. This implicitly rejects other DHCPOFFERS.
    - When Rebooting the client does not know if the address is now allocated to another node. Therefore it should broadcast.
    - If the client broadcasts the server should respond with a broadcast.
- If the client has an IP Address it can unicast to the server.
  - Broadcast (B) Flag should be unset
    - Extending the lease.
  - Server should respond with unicast

# DHCP Operation

- DCHP DISCOVER

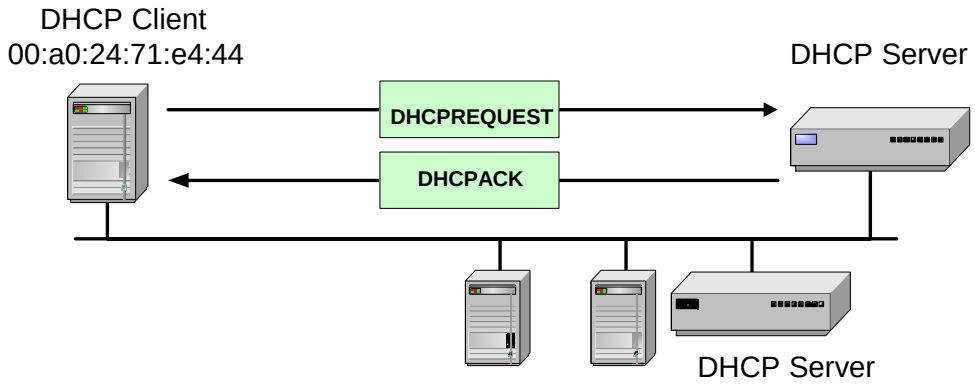



- DCHP OFFER



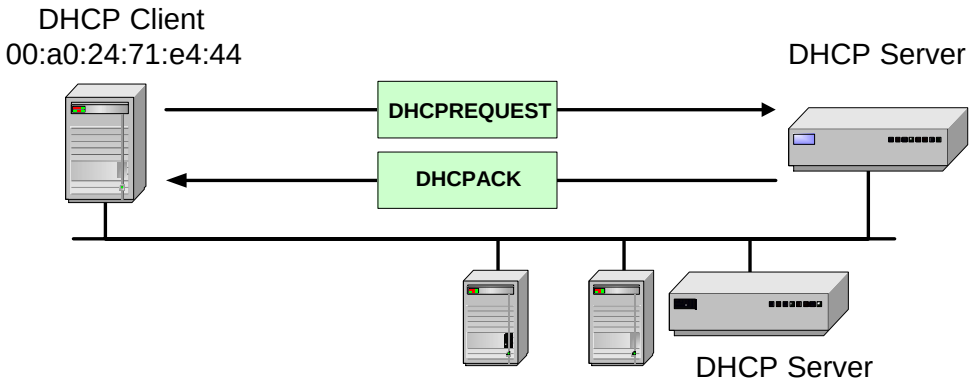
# DHCP Operation

- DCHP DISCOVER



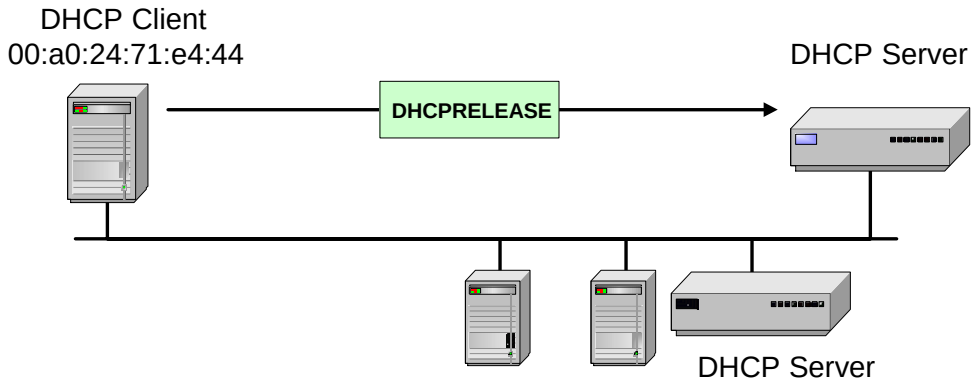

Upon receipt of the DHCPACK, the DHCP client can start to use the IP address

- Renewing a Lease: Sent when 50% of lease has expired. The DHCP server can refuse lease extension by sending a DHCPNACK



# DHCP Operation

- DCHP RELEASE: The DHCP client releases its IP address



# ADDRESS RESOLUTION PROTOCOL

# Address Resolution Problem

- Upon configuration, hosts know their IP address. Mask, a default Router, and a DNS server
- Given the IP address of a host, how do we obtain the corresponding hardware address ?
  - This process of is referred to as the Address Resolution
    - Local Process

# ARP Problem

- Host A has ($I_A$, $P_A$) and Host B has ($I_B$, $P_B$), as an Internet and MAC address respectively
  - A knows $I_B$ and needs $P_B$ to physically transmit the packet

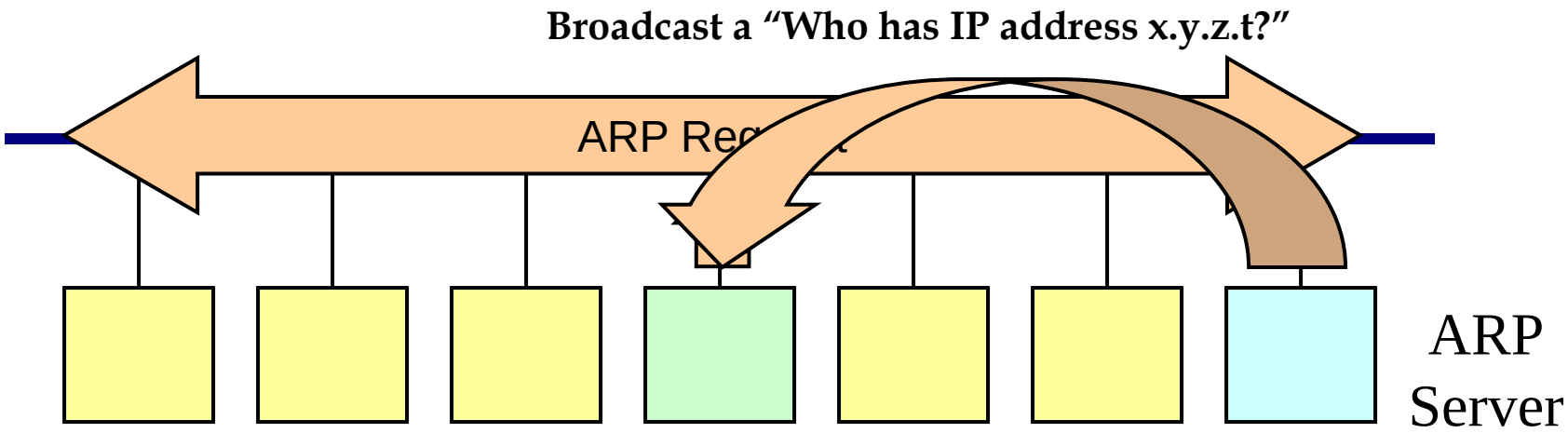
- Conceptually, we need a mapping $F$
  - $P_H = F(I_H)$ where $H$ is a given internet host

# Approaches to the Solution

- Mapping tables in each host containing ($I_H$, $P_H$), for all hosts $H$
  - Expensive in time and space
- Direct mapping
  - Select $I_H$ and $P_H$ such that an extraction of the $P_H$ from $I_H$ is computationally efficient
    - Pronet addressing mechanism
    - May not work for Etherenet (48 bits)
- Dynamic address resolution
  - ARP, a low level protocol to bind addresses dynamically

# Address Resolution Protocol

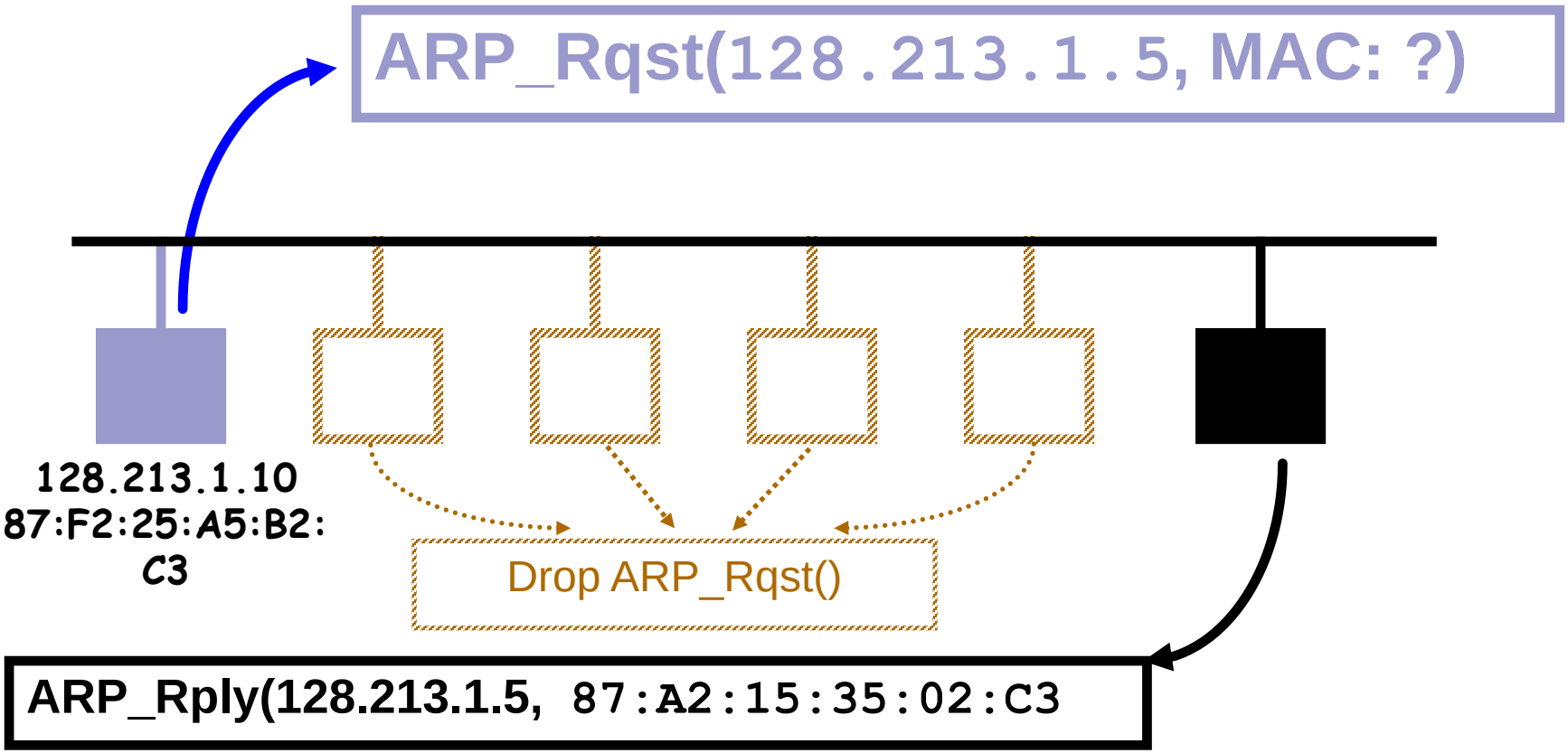
- The *Address Resolution Protocol* is used by a sending host which seeks to resolve the IP address of the destination into the corresponding Ethernet address.
  - Ethernet address will carried into the Ethernet frame which encapsulates the IP datagram
- ARP is a broadcast protocol
  - Every host on the network receives the request.
- Each host checks the request against its IP address, upon receipt of the request
  - Only the sought after station responds.



# ARP Refinement
# Sender Address Mapping

- Sender includes (IS, PS) in every ARP broadcast
  - Anticipates the need of the receiver to resolve PS
    - Receiver no longer need to issue an ARP request
- Other machines can also learn sender's (IS, PS) mapping
  - Direct benefit from the broadcast

# ARP Refinement – Caching




Cache: (128.213.1.5 ←→ 87:A2:15:35:02:C3)

Cache: (128.213.1.10 ←→ 87:F2:25:A5:B2:C3)

# ARP Frame Format

| Hardware Type | | Protocol Type |
|---|---|---|
| HLen | PLen | Operation |
| Sender HA (0-3) | | |
| Sender HA (4-5) | | Sender IP (0-1) |
| Sender IP (2-3) | | Target HA (0-1) |
| Target HA (2-5) | | |
| Target IP (0-3) | | |

# ARP Hardware Types

ARP Hardware Types

| Type | Description |
|---|---|
| 1 | Ethernet (10 Mbps) |
| 2 | Experimental Ethernet (3 Mbps) |
| 3 | Amateur Radio X.25 |
| 4 | Proteon ProNet Token Ring |
| 5 | Chaos |
| 6 | IEEE 802 Networks |
| 7 | ArcNet |

# Reverse ARP

- Usually, a machine's IP address is kept on its secondary storage
  - At startup time, the OS accesses the IP address
- How does a diskless machine determine its IP address?
  - Reverse Address Resolution Protocol

# NETWORK ADDRESS TRANSLATION

# Network Address Translation

- Combined with CIDR, NAT offers a short term solution to the problem of IP address depletion
  - RFC-1631
- NAT is designed to conserve IP addresses
  - Use of private addresses, internally
- Long term solution is provided by IPv6

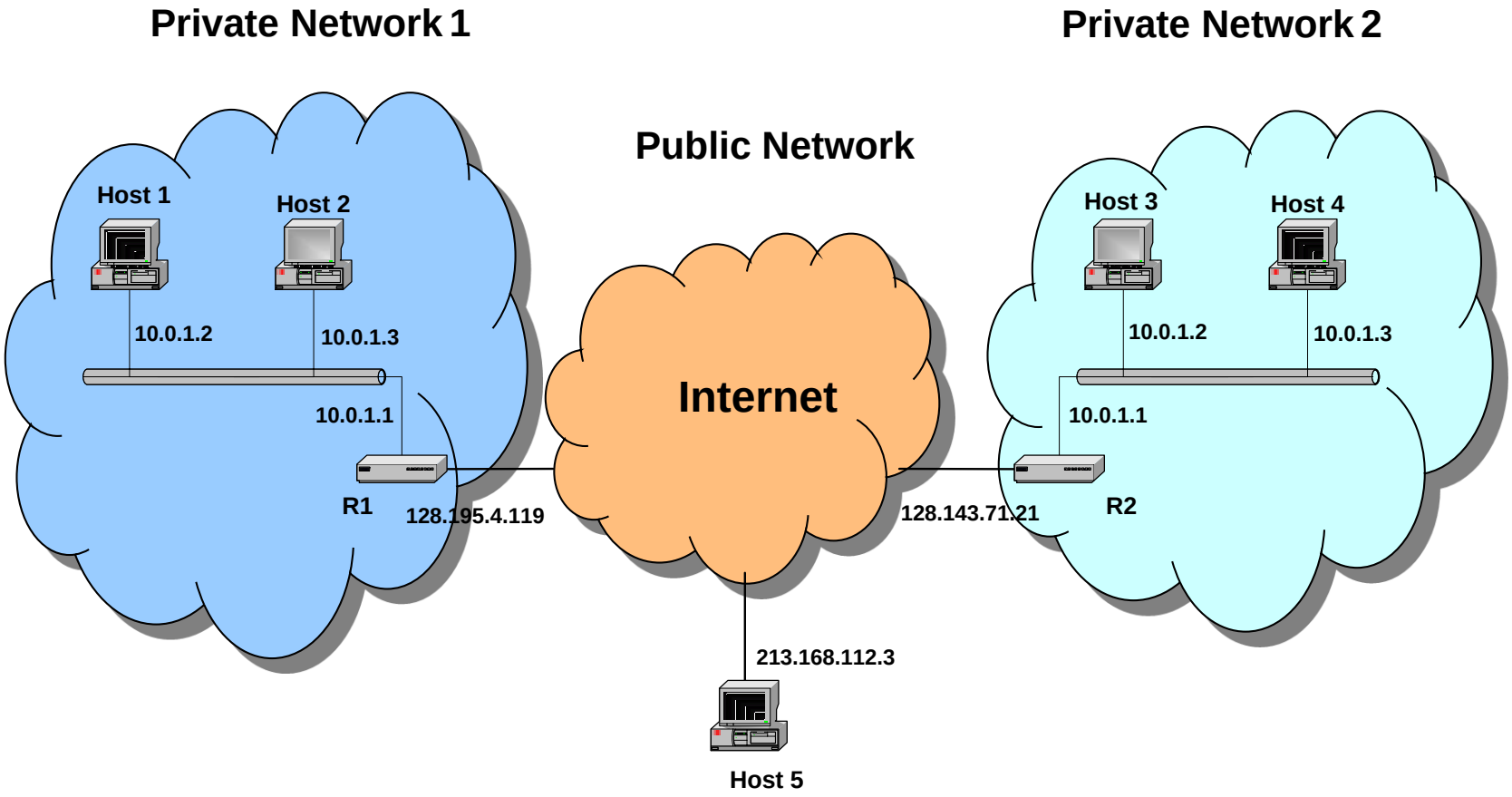
# Network Address Translation Protocol

- NAT is a protocol that enables hosts on private networks to communicate with hosts on the Internet
  - NAT is run on routers that connect private networks to the public Internet,
- NAT replaces IP addresses, and possibly port numbers, of IP datagrams at the boundary of a private network

# Private Networks

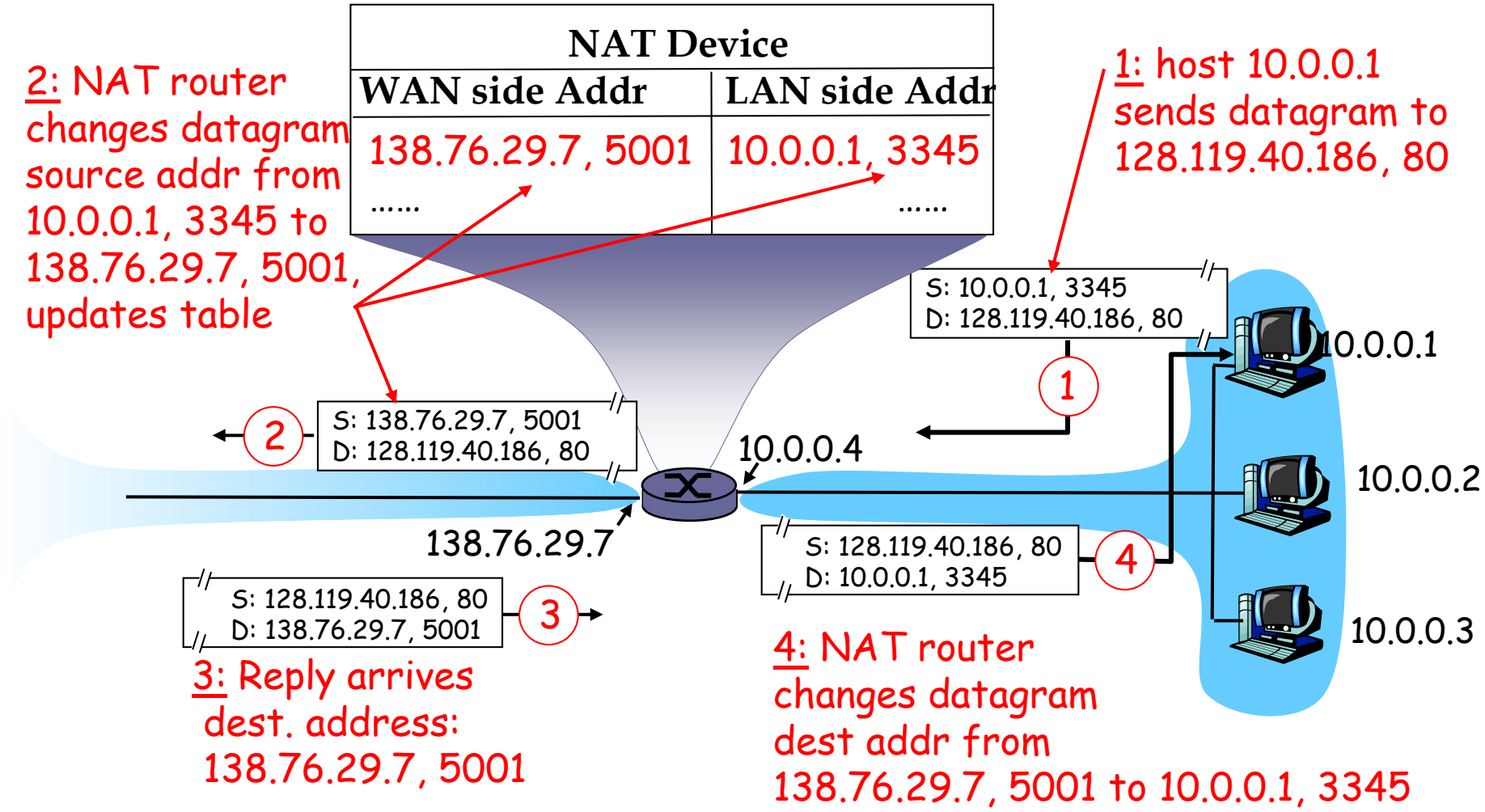
- *Private IP* network is an IP network that is not directly connected to the Internet
- IP addresses in a private network can be assigned arbitrarily.
  - Not registered and not guaranteed to be globally unique
- Generally, private networks use addresses from the following experimental address ranges (*non-routable addresses*):
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.255.255
  - 192.168.0.0 – 192.168.255.255

# Private Networks and Addresses




Private network addresses are not routable

# Network Address Translation Basic Protocol



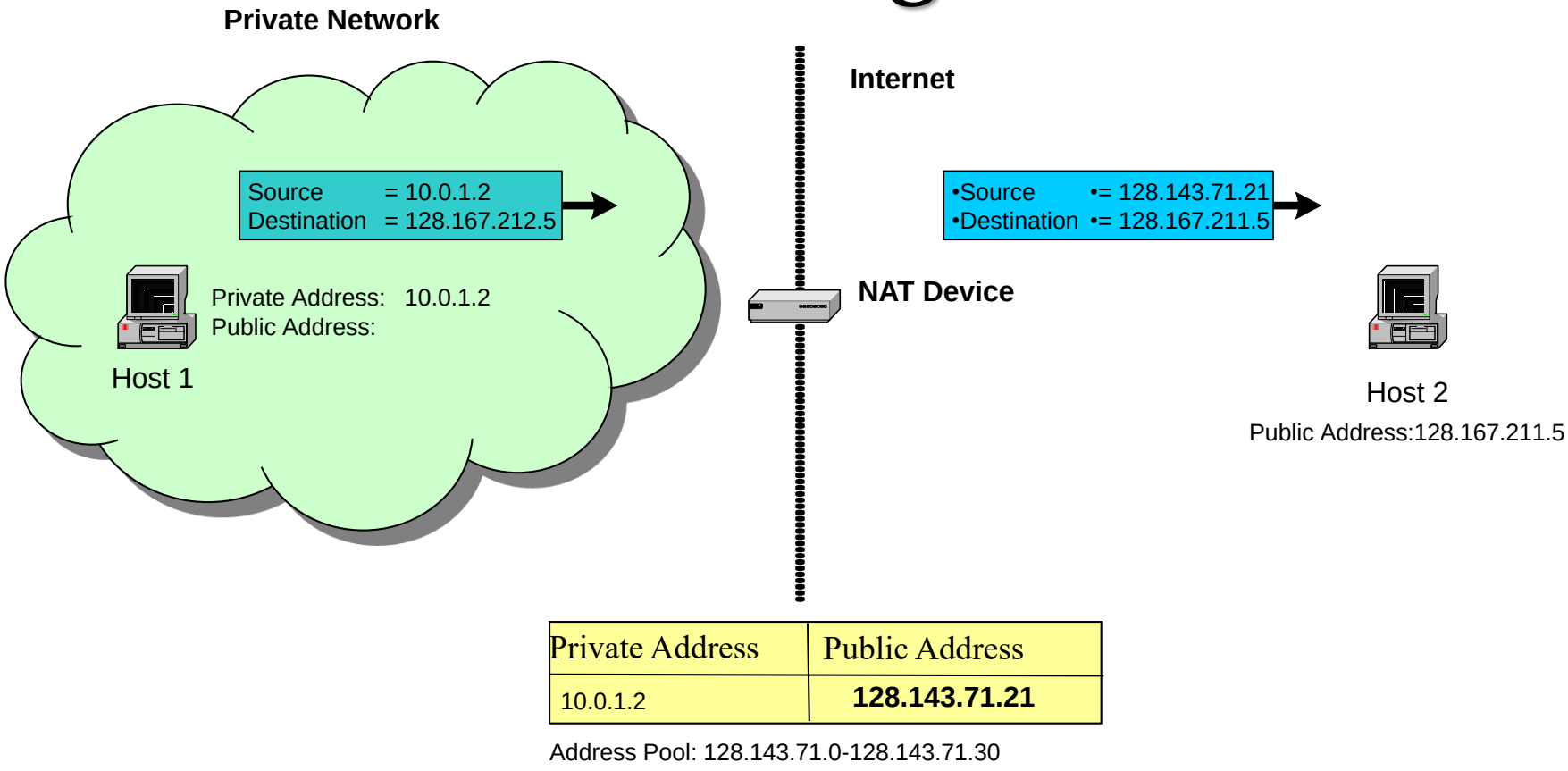
# Main uses of NAT

- Pooling of IP addresses
- Host migration support between network service providers
- IP address and port translation
  - IP masquerading
- Load balancing of servers

# Pooling of IP addresses

- Objective is to alleviate the public address shortage problem of corporate networks with large number of hosts and limited number of public addresses
  - Corporate network is managed internally with a private address space
- NAT device, located at the boundary between the corporate network and the public Internet, manages a pool of public IP addresses
  - NAT device selects a public IP address from the address pool, and binds it to the private address of the host

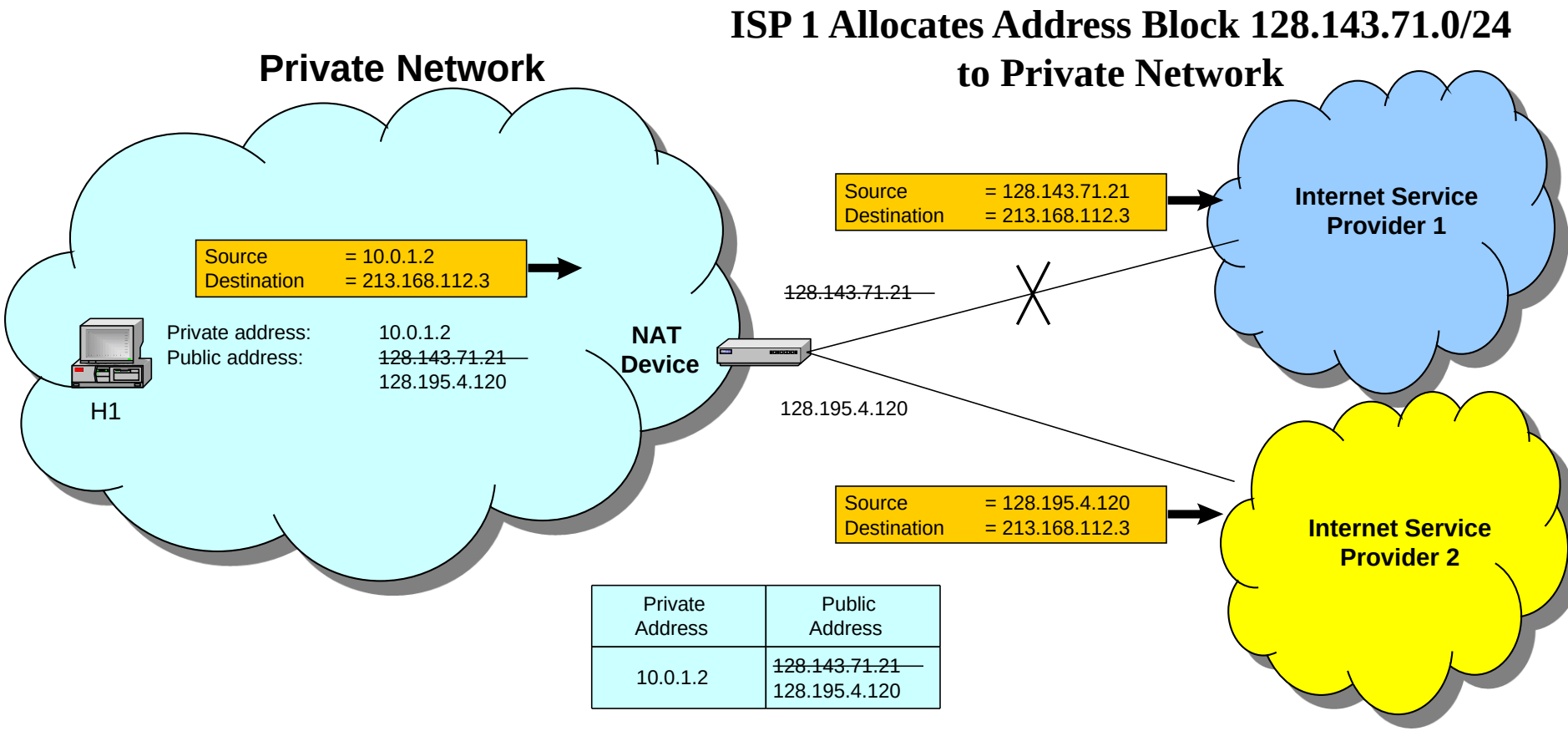
# IP Addresses Pooling




| Private Address | Public Address |
|---|---|
| 10.0.1.2 | **128.143.71.21** |

Address Pool: 128.143.71.0-128.143.71.30

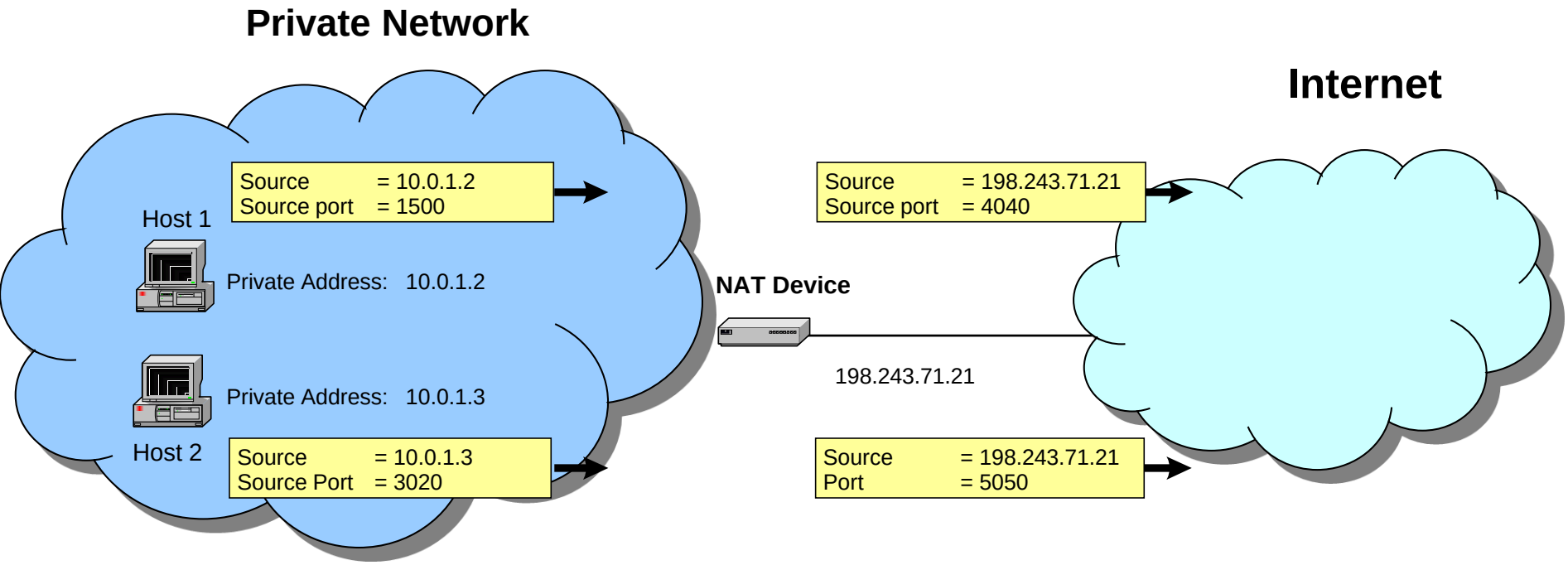
# Host Migration Between Network Service Providers



# IP Address and Port Translation
## Address Port Translation

- Single public IP address is mapped to multiple hosts in a private network.
  - Assign private addresses to the hosts of the corporate network
  - NAT device modifies the port numbers for outgoing traffic

# IP Address and Port Translation




| Private Address | Public Address |
|---|---|
| 10.0.1.2/1500 | 198.243.71.21/4040 |
| 10.0.1.3/3020 | 198.243.71.21/5050 |

# Server Load Balancing

- A set of identical servers, accessible from a single IP address, are configured to provide similar service
  - The objective is to balance the load
- Servers are assigned private addresses
  - NAT device acts as a proxy for requests to the server originating from the public network
  - The NAT device substitutes the destination IP address of arriving packets to one of the private addresses for a server
    - Typically, server address substitution is done in a round-robin fashion.

# Running Servers Behind NATs

- Running servers is still possible, yet difficult
- By explicit configuration of the NAT box
  - E.g., internal service at <dst 138.76.29.7, dst-port 80>
  - … mapped to <dst 10.0.0.1, dst-port 80>
- More challenging for P2P applications
  - Especially if *both* peers are behind NAT boxes
- Workarounds are still possible
  - Skype, for example
  - Ongoing work on "NAT traversal" techniques

# NAT Limitations

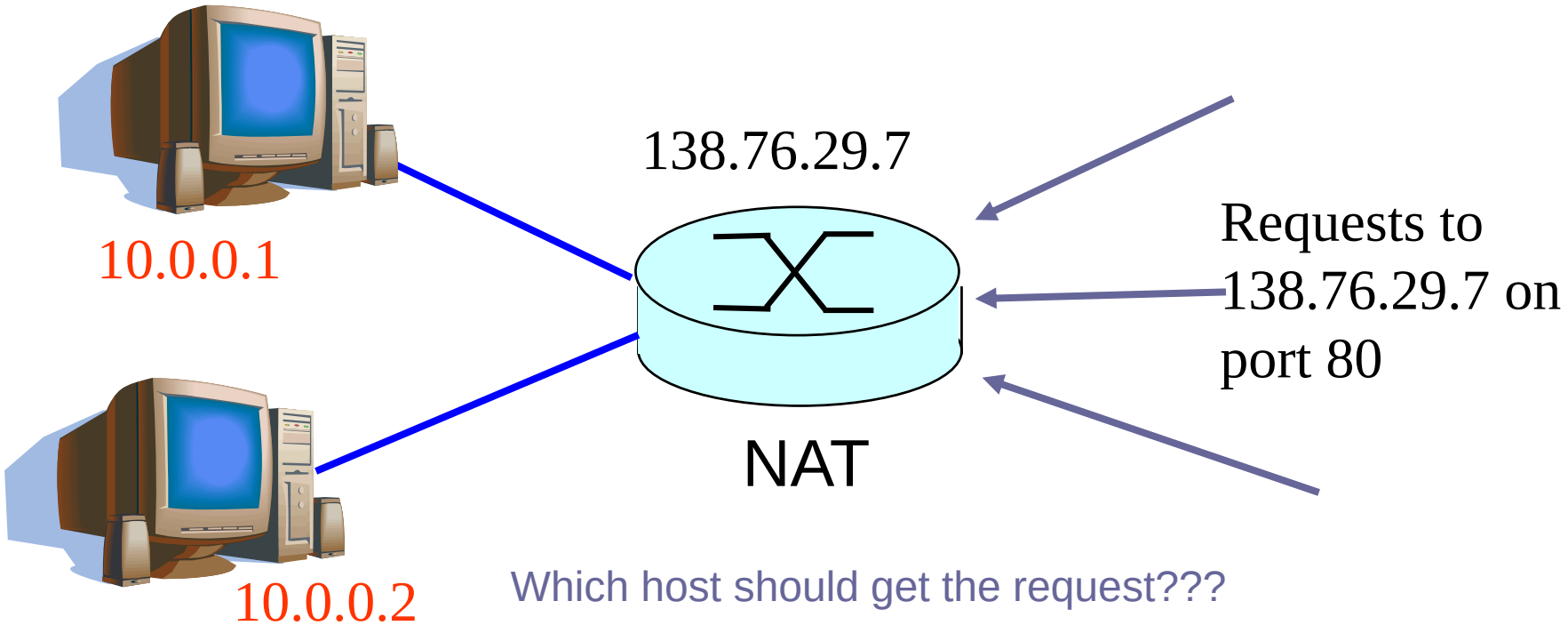
- NAT use is problematic with:
  - Protocols that require a separate back-channel
  - Protocols that encrypt TCP headers
  - Embed TCP address information
  - Specifically use original IP for some security reason

# Services That Cause NAT to Fail!

- H.323, CUSeeMe, VDO Live – video teleconferencing applications
- Xing – Requires a back channel
- Rshell – used to execute command on remote Unix machine – back channel
- IRC – Internet Relay Chat – requires a back channel
- PPTP – Peer-to-Peer Tunneling Protocol
- SQLNet2 – Oracle Database Networking Services
- FTP – Must be RFC-1631 compliant to work
- ICMP – sometimes embeds the packed address info in the ICMP message
- IPSec – used for many VPNs
- IKE – Internet Key Exchange Protocol
- ESP – IP Encapsulating Security Payload

# Practical Objections Against NAT

- Port #s are meant to identify *sockets*
  - Yet, NAT uses them to identify *end hosts*
  - Makes it hard to run a server behind a NAT



# Principled Objections Against NAT

- NAT violates the *end-to-end* argument
  - Network nodes should not modify the packets
    - Network layer should care *only* about IP header, … and *not* be looking at the port numbers at all
- IPv6 provides a cleaner solution to many of these issues
  - Increasing deployment of IPv6 to address ROADS, mobility and security