

CSCD 330

Network Programming

Spring 2018



Lecture 14

Network Layer

IP Addressing

Reading: Chapter 4

Some slides provided courtesy of
J.F Kurose and K.W. Ross, All Rights Reserved, copyright 2017

Network Layer

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- **4.4 IP: Internet Protocol**
 - **Datagram format**
 - **Fragmentation**
 - **IPv4 addressing**
- 4.5 Routing algorithms
 - Link state
 - Distance Vector
 - Hierarchical routing
- 4.6 Routing in the Internet
 - RIP
 - OSPF
 - BGP
- 4.7 Broadcast and multicast routing

Introduction

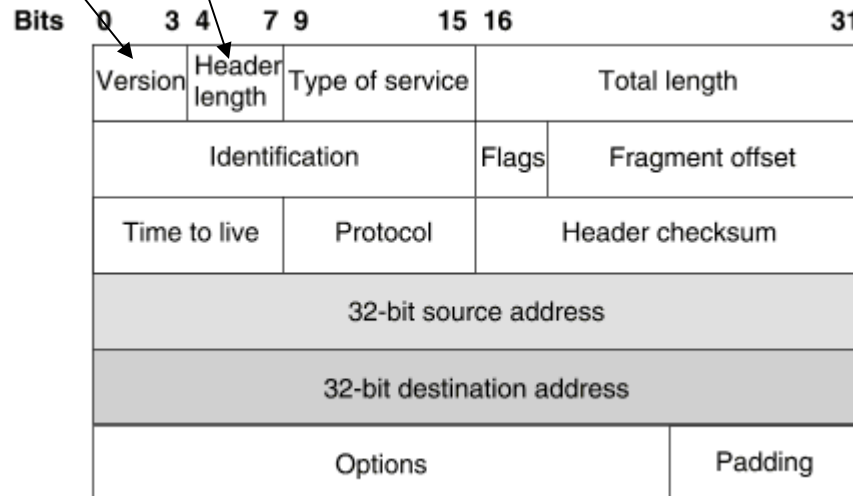
- Last time, began the network layer
- Provides a best effort service most of the time
- Alternate model - virtual circuit like ATM, MPLS tries to create a virtual circuit on top of the best effort datagram environment
- Today, get into IP addressing

IP Version 4 Header

- **IPv4 Frame Header**
 - Designed to handle addressing and routing challenges
 - Think about trying to route through a network where ...
 - **Physical network varies** – different capacity of links
 - **Maximize efficiency** – means minimize redundancy
 - **Account for both uncertain and certain delivery**
 - **Handle errors**
 - Explains the IP packet header ...

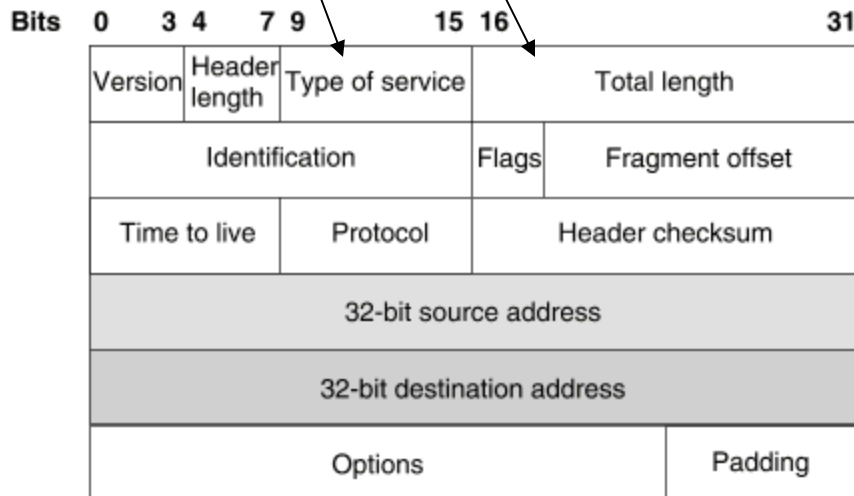
IPv4 Header

- Fields Explained
- First, header length is normally 20 bytes, has it own field
 - **Version** – IPv4 or IPv6 - will contain 4 or 6
 - **Header Length** – Length in 32 bit words (4 bytes)
 - Most of the time its 5 – for 20 bytes of header,



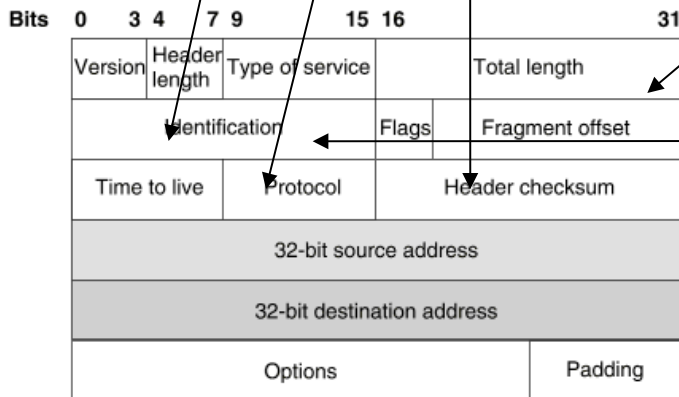
IPv4 Header

- **Type of Service** – Application dependent – applications can say to treat packets differently
 - **Length** – Maximum length packet in bytes
 - 16 bits so, packets can be 65,535 bytes long



IPv4 Header

- **Time to Live (TTL)** – Used to be a time, now its more a hop count – don't want packets circulating through network forever
 - Routers decrease this by 1, default 64
- **Protocol** – Upper layer protocol, TCP or UDP or ICMP
- **Checksum** – Only of the IP Header!
 - Computed by adding header as 16 bit words using ones complement arithmetic and then taking the one's complement of the result.
 - At destination, compare it to value stored in this field and if different discard the packet



Line Two Deals with Fragments

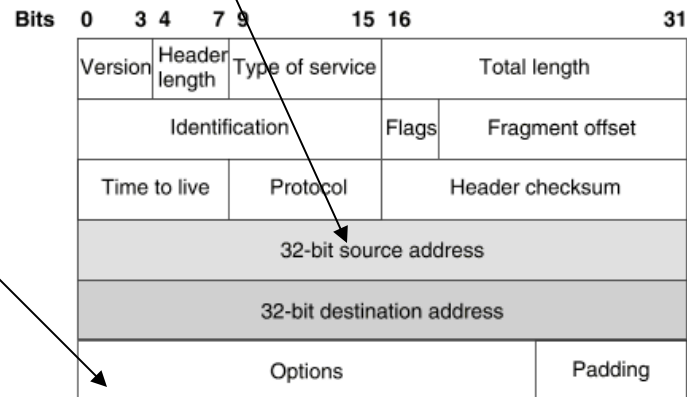
Fragment offset – 13 bits, in bytes/8

Fragment Identification – 16 bits, identifier set by sender

Flags – 3 bits, indicates fragments

IPv4 Header

- **Source Address** 32 bits long
 - **Destination Address** 32 bits long
 - **Options field** – used sometimes – not always
- Gives us 2^{32} addresses or over 4 billion addresses



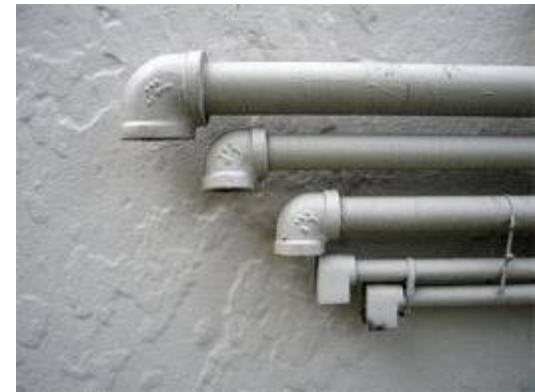
Fragmentation



- One design decision helped Internet remain flexible
 - Able to accommodate multiple network technologies
 - Packet Fragmentation
 - Packets can be divided so can pass through links of different sizes
 - Some typical maximum packet sizes
 - Ethernet – 1500 bytes long
 - FDDI – 4500 bytes long
 - Point to point (PPP) – 532 bytes long

Fragmentation

- Every network sets its MTU
 - **Maximum Transmission Unit**
 - Largest Frame Size of Data link layer
 - Previous slides shows -> MTU varies with link type
 - IP packets need to adjust to that frame size
 - Think of squeezing packets through different sized pipes
 - If MTU along the way, is smaller than IP packet size on your network
 - Datagram must fit within payload of link layer frame
 - **Fragmentation occurs in router**
 - When datagram size > than network MTU it must travel over

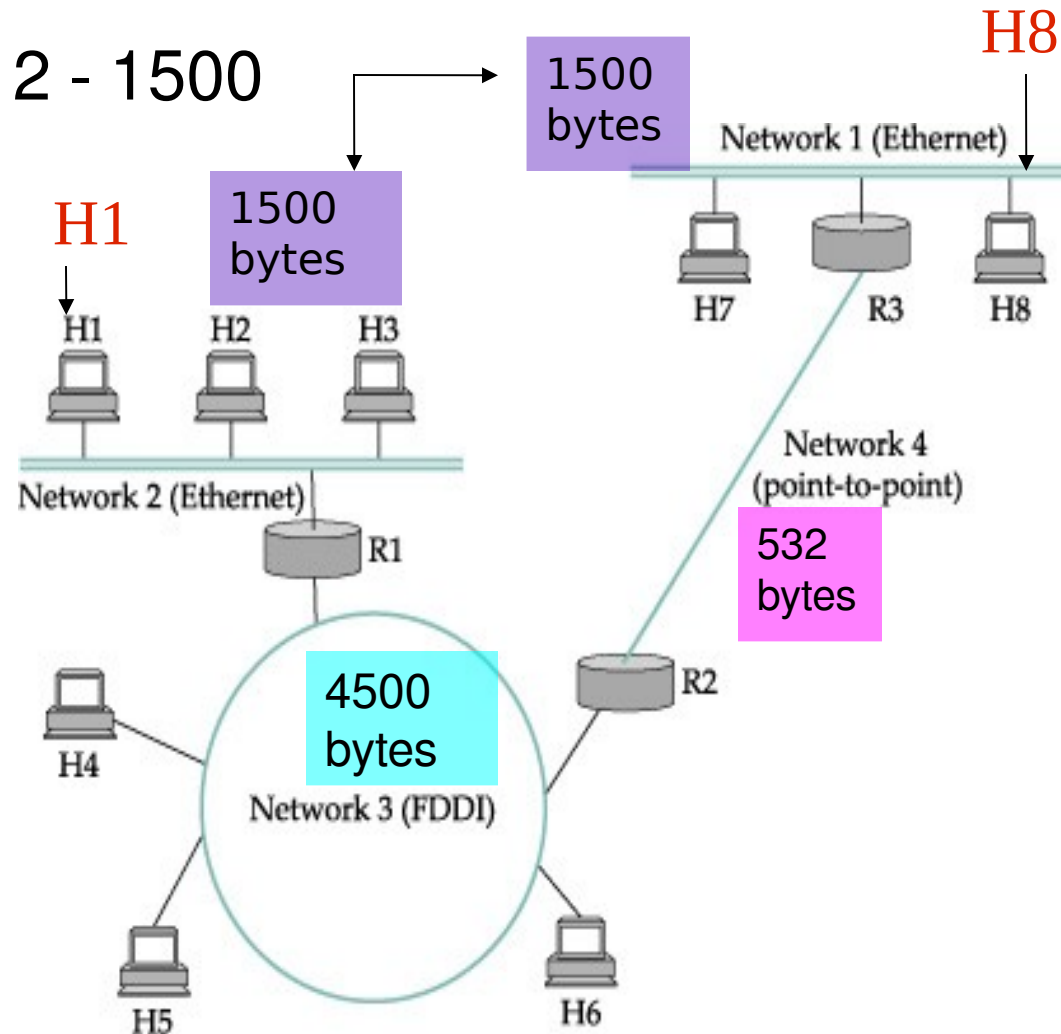


Example Fragmentation Links

- **MTU's**
- Ethernet, Networks 1 and 2 - 1500
- FDDI, Network 3 - 4500
- PPP, Network 4 - 532

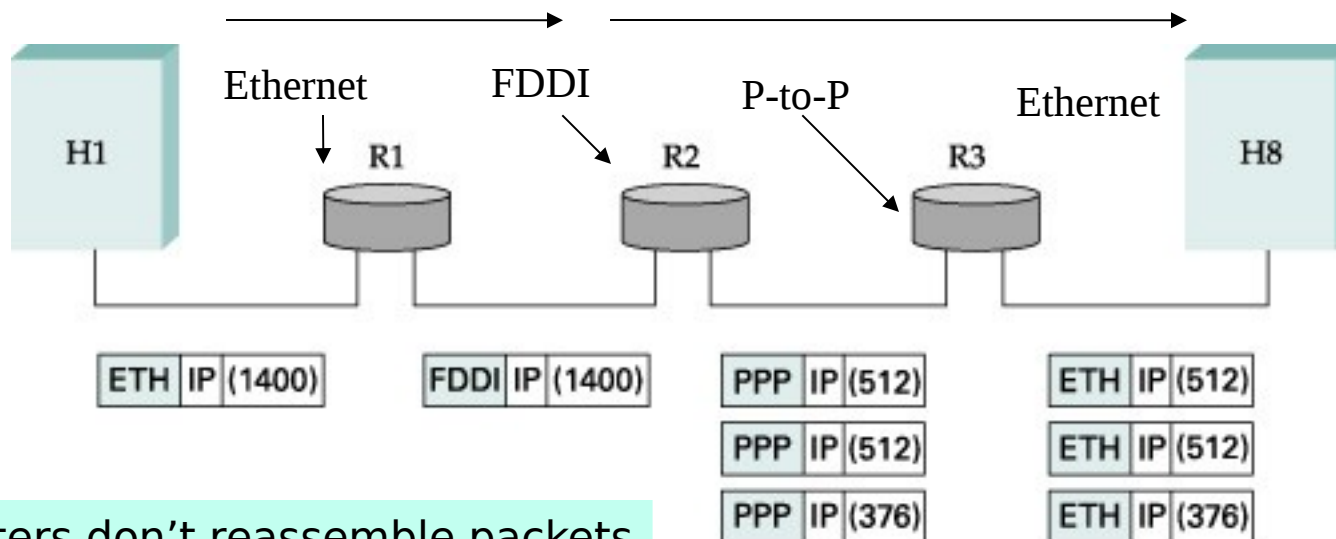
- **H1** sends datagram to **H8**, **1400** bytes data and **20** bytes header **1420** total

Do we Need to fragment?



Fragmentation: Example Network

- Ethernet R1 and FDDI R2 – No fragmentation needed
 - Why is that?
- PPP R3 – Fragmentation is needed

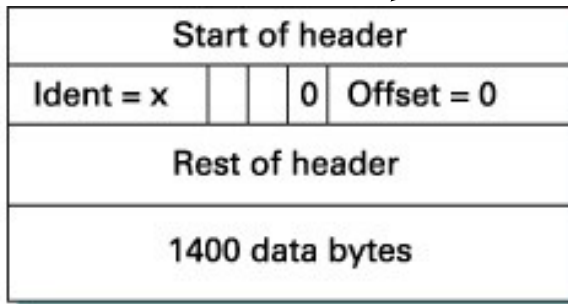


Routers don't reassemble packets
Host does packet reassembly

Original Packet

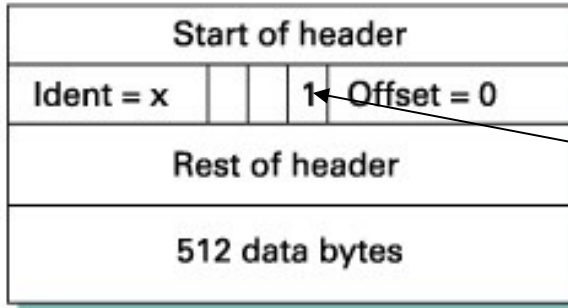
Bytes
1400

(a)

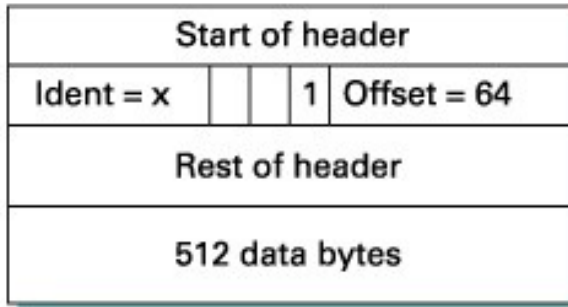


512

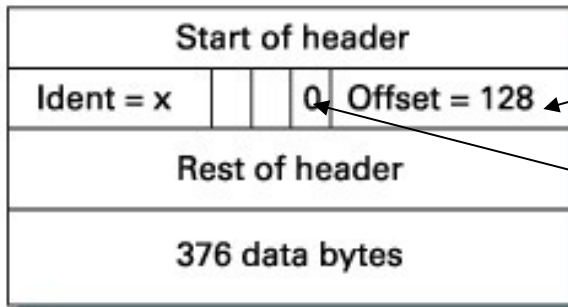
(b)



512



376



Example of fragment

512 bytes of data + 20 bytes of header, why this size?

Flag = 1 for more fragments follow

Offset is in groups of 8 bytes

$$512/8 = 64$$

$$\text{Offset} = 2 \times 64 = 128$$

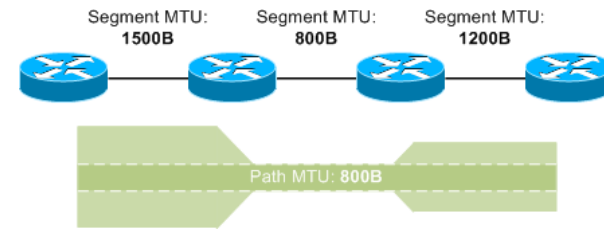
Flag = 0, no more fragments

Fragments Reassembly



- IF all fragments don't arrive,
 - **What do you do?**
 - Discards ones that have arrived
 - Packet will need to be resent
 - Consequently, fragments can waste resources
 - Not recommended !!!
 - Hosts encouraged to perform path MTU discovery
 - Avoid fragments if possible

Fragmentation Discouraged



- To avoid fragmentation, hosts commonly use path MTU discovery to find smallest MTU along path
- Path MTU discovery involves sending various size datagrams until they do not require fragmentation along the path
- Most links use MTU \geq 1500 bytes today
- Also a nice tool to check, TracePath,
<http://www.subnetonline.com/pages/network-tools/online-tracepath.php>
- Nice short summary of MTU Path discovery
<http://www.netheaven.com/pmtu.html>



Network Addressing

Addressing Global Level

- Look at how IP level performs addressing of packets ...
- **Ethernet/Link Layer address scheme is flat**
MAC (Media Access Control) address doesn't have much structure or hierarchy

00:13:02:BA:43:56

- Each MAC address is burnt into an individual Network card including wireless
- **When you connect to a network,**
 - MAC address operates at the link layer and
 - IP address operates at the Network layer

Addressing Global Level

- IP address is divided into Two Parts
 - **Host** and **network**
 - Addresses are also **hierarchical**
 - Allows network to scale
 - Routing tables only need network part
 - Local delivery inside a network doesn't involve router at all and needs the host part

Forwarding in General



- **Forwarding Datagrams**

- Source host sends datagram to Destination host
- Passes through possibly several routers
 - Router asks,
 - “Is Source on Same Network as Destination?”
 - Compares network part of destination with network part of address for each of its interfaces
 - **If match**, destination on same physical network
 - » Packet can be delivered
 - **If no match**, sends datagram to next router
 - » Router chosen is “next hop” router
 - » Found via forwarding table



IP Addresses

IP Addressing An Evolution

- IP Addresses were Grouped together to form networks
 - Didn't happen by accident
 - Evolving process
 - First attempt
 - Divide addresses into rigid categories
 - Where there were fixed number of computers per category
 - Then, more flexible way of addressing
 - Subnetting
 - Still working on it ... IPV6 is next

IP Addresses RFC 1166

- Each host on Internet has unique 32 bit IP address
- Each address has two parts: **netid** and **hostid**
hostid simply the number of the host on that network
netid is network part of the IP address

Network part is Administered by

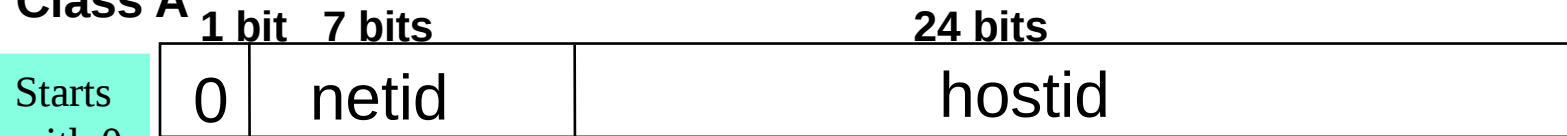
- American Registry for Internet Numbers (**ARIN**) or
- Reseaux IP Europeens (**RIPE**) or
- Asia Pacific Network Information Centre (**APNIC**)
- Each host has Separate IP address
- **Dotted-Decimal Notation**
- IP address of 10000000 10000111 01000100 00000101
 - is 128. 135. 68. 5
 - Called **Dotted-decimal notation**
 - **Each part is an octet, 8 bits**

Classful Addressing

- **Classful Addressing**, formally adopted as part of the Internet Protocol (IP) in RFC 791, 1981
 - Internet's first major addressing scheme
- **There were three address classes to choose from:**
 - A, B, or C, corresponding to 8-bit, 16-bit, or 24-bit network prefixes
 - No other prefix lengths were allowed
 - Not very flexible to needs of organizations

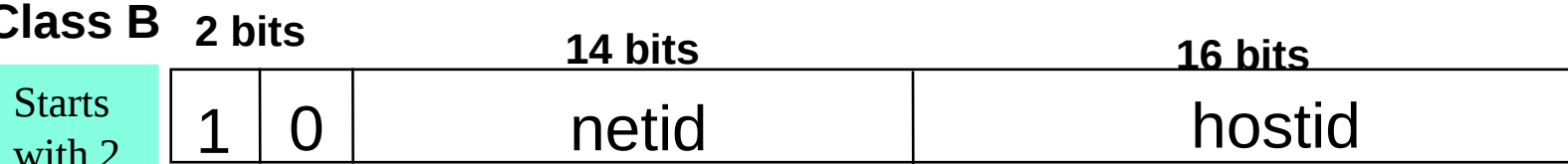
Classful Addresses

Class A



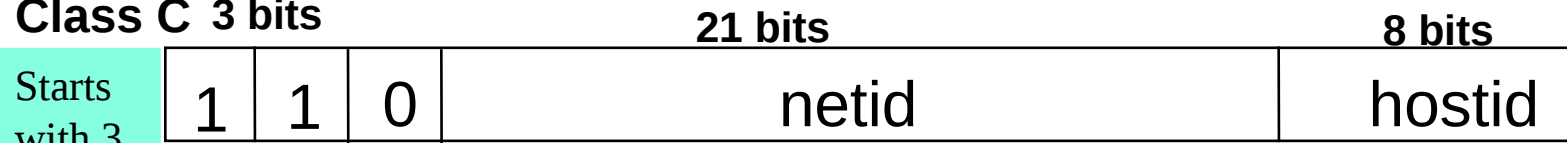
- 128 networks with up to ~16 million hosts 1.0.0.0 to 127.255.255.255

Class B



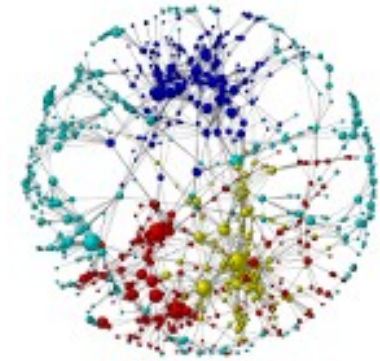
- 16,384 networks with up to ~65,000 hosts 128.0.0.0 to 191.255.255.255

Class C



- ~2 million networks with up to 256 hosts 192.0.0.0 to 223.255.255.255

Classful Addresses



- **Notes**
 - Rigid allocation of networks
 - First 3 bits were fixed, routers easily could tell type of network by first three bits
 - Inverse relationship between network number and number of hosts on network
 - As network number grows, host number shrinks
 - Has to do with how many bits left over for hosts

Classful Address Examples

- Upon installing a new Internet connection, network engineer
 - Requested Class A, B, or C network, depending on expected size of installed network
 - **For example**
 - **U.S. Department of Defense**
 - Large network, got a **Class A** network
 - **University of Maryland,**
 - Mid-sized network, got a **Class B** network
 - **Small consulting firm**
 - Small network, got a **Class C** network

Problems with Classful Addressing

- Soon became apparent that **classful addressing** was inadequate
- For example, a reasonably sized company
 - **Class C, with 254 IP addresses**, would be too small
- Next larger choice would be a
 - **Class B, providing 65,534 IP addresses**, too big
 - Unlikely that thousands of hosts would be located on a single Ethernet LAN
- **How could we fix this?**

Subnetting to the Rescue



- **Many network engineers wanted**
 - Ability to take Class B and break its 65,534 addresses into 100-200 smaller networks of 200-300 addresses each
- Smaller networks became known as **Subnets**, and a standard scheme called subnetting was formalized in **RFC 950**
- Was in 1985 ... around 30 + years ago

<http://www.faqs.org/rfcs/rfc950.html>

Subnets and Subnet Masks

- What does subnetting do?
- **Subnetting** splits **host** field
 - Subnet + host fields, creates a three-part address
 - Network + Subnet + host
 - Network field remains unchanged,
 - You start with the **Classful** classification, then borrow bits from host to make your subnets

Subnets and Subnet Masks

- When subnet created, engineer also creates **subnet mask**,
- 32 bits long, dotted decimal format, like IP address
 - **Purpose:** Lets routers identify portion IP address that's related to **network**
 - Each bit is either 1 to identify bit positions in the network and subnet fields, or 0 to identify bit positions in host field
 - Then, the mask is **AND'd with IP Address**
255 = What is this value in binary?
11111111 by the way

Subnet Mask

- Want only network portion of IP address
- Host part will be zero'd out
- Routers only **care** about network part IP Address
- Mask then, is
 - 1's for the network part including subnet
 - 0's for the host part
- See some examples ...

Example: Subnet Mask

- **Example 1.: Say 16 bits of network address**
- 191.70.55.130
- 10111111.10000110.00110111.10000010
- What would the mask look like?

- **Example 2: Say 24 bits of network address**
- 192.168.5.10
- 11000000.10101000.00000101.00001010
- What would the mask look like?

Example Apply a Subnet Mask

- IP Address: 187.199.127.5
Subnet mask: **255.255.255.0**
- **Anding** two together gives us
Network part: 187.199.127
- Network Class: B – 16 bits of address
- **Network ID: 187.199**
- **Sub-network ID: 127**
- Router needs above, host is ignored by router

Subnetting Creates Hierarchy

- Idea with subnetting, take one IP address
 - Partition it into several IP addresses
 - Each refers to an actual physical network
- Assume that subnets are geographically close to one another
 - Because ... distant routers only have one IP address for entire set of subnets
 - So sending packets to these subnets through one IP number should route packets in the same general direction

Subnetting Creates Hierarchy

- Subnetting creates another level of hierarchy within IP addresses
 - Now there is a subnet part too, since borrow bits from host

| | | |
|-----------|--------|------|
| Network # | Subnet | Host |
|-----------|--------|------|

| | |
|-----------|------|
| Network # | Host |
|-----------|------|

Addressing Problems not Over

- By 1990, the Internet was facing serious growing pains ...
 - Two most severe problems were
 - **Explosion of routing table size**
 - **Looming exhaustion of Class B networks**
 - Popularity of Internet triggered flood of new networks, and each network included in routing tables
 - Routers were running out of memory, and spending too much time doing address lookup

Classless Interdomain Routing (CIDR)

- Internet Engineering Task Force (IETF), proposed
 - **Classless routing, supernetting, or CIDR**
 - This addressing scheme currently used
- **CIDR** based on subnetting
- **Supernetting** allows subnet boundary to move to the left, into network part
 - Groups of neighboring **classful** networks are combined into single routing table entries
 - Size of routing tables reduced through summarization ... look at that later

CIDR Example

- Created CIDR notation of networks ...

192.0.2.0/18

- /18 says that the first 18 bits are network part of address and 14 bits are available for host addresses
- Network part is called **prefix**
- **Example of number of bits needed for network**
- Assume site requires network for 1000 addresses
- **How many bits of network address gives us 1000 hosts?**
 - Answer: network is assigned block of 1024 addresses with a 22-bit long prefix

See following slide for table

CIDR Prefix Size vs. Network Size

| CIDR Block Prefix | # of Host Addresses |
|--------------------------|----------------------------|
| /27 | 32 hosts |
| /26 | 64 hosts |
| /25 | 128 hosts |
| /24 | 256 hosts |
| /23 | 512 hosts |
| /22 | 1,024 hosts |
| /21 | 2,048 hosts |
| /20 | 4,096 hosts |
| /19 | 8,192 hosts |
| /18 | 16,384 hosts |
| /17 | 32,768 hosts |
| /16 | 65,536 hosts |
| /15 | 131,072 hosts |
| /14 | 262,144 hosts |
| /13 | 524,288 hosts |

CIDR Network Size / Number of Network Bits

- CIDR networks referencing Class C networks

/n = number of network bits

CIDR Block Prefix # Equivalent Class C # of Host Addresses

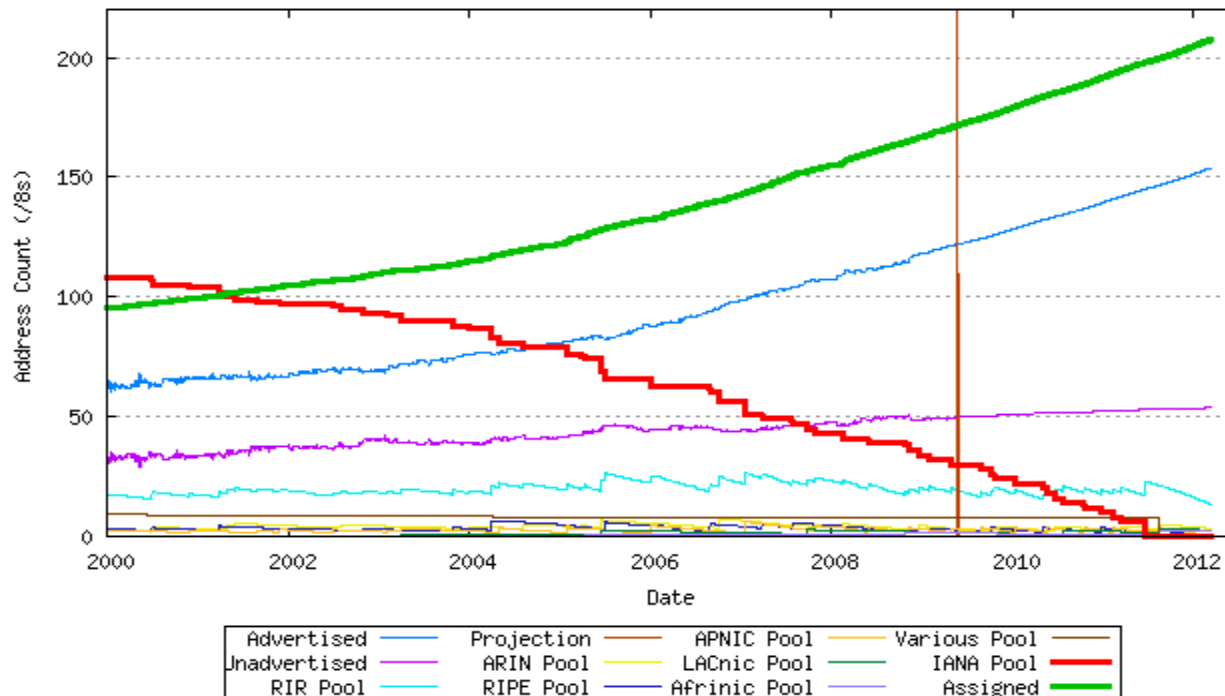
| | | |
|-----|--------------------|--------------|
| /27 | 1/8th of a Class C | 32 hosts |
| /26 | 1/4th of a Class C | 64 hosts |
| /25 | 1/2 of a Class C | 128 hosts |
| /24 | 1 Class C | 256 hosts |
| /23 | 2 Class C | 512 hosts |
| /22 | 4 Class C | 1,024 hosts |
| /21 | 8 Class C | 2,048 hosts |
| /20 | 16 Class C | 4,096 hosts |
| /19 | 32 Class C | 8,192 hosts |
| /18 | 64 Class C | 16,384 hosts |
| /17 | 128 Class C | 32,768 hosts |
| /16 | 256 Class C | 65,536 hosts |

Ipv4 Address Exhaustion

- Depletion of IPv4 allocation pool a concern since late 1980s when Internet started dramatic growth
- Anticipated shortage driving factor in creating and adopting several technologies
 1. Classless Inter-Domain Routing (CIDR) methods in 1993
 2. Network address translation (NAT) and
 3. Internet Protocol, IPv6, in 1998

Ipv4 Address Exhaustion

- Predictions of exhaustion date of unallocated IANA pool seem to converge to between January 2011 and January 2012,
- When did we run out? Did this in Nov, 2011.





Stop here for now

CIDR and Subnetting

- To understand CIDR, begin with **defaults** for classful addressing
- Each Class has a DEFAULT subnet mask
 - The picture below shows the 3 Network Classes with their respective default subnet mask:

Network Classes with their respective Default Subnet Masks

| <u>CLASS TYPE</u> | <u>NETWORK RANGE</u> | <u>DEFAULT SUBNET MASK</u> |
|-------------------|------------------------------|----------------------------|
| Class A | 1.0.0.0 to 127.255.255.255 | 255.0.0.0 |
| Class B | 128.0.0.0 to 191.255.255.255 | 255.255.0.0 |
| Class C | 192.0.0.0 to 223.255.255.255 | 255.255.255.0 |

Here you can see each Network Class with its range of IP Addresses followed by the default subnet mask for the particular Class. Remember that we can modify the default subnet mask to meet our needs

Default Subnet Masks

- We can see in the picture below,
- IP Address is in Binary and Mask is 24 bits

Class C Classful IP Address

IP Address : 192 . 168 . 0 . 5
Subnet mask : 255 . 255 . 255 . 0

↓ Conversion to Binary

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------|-------------------|----|----|----|---|---|---|---|---|-----|----|----|----|---|---|---|---|---|----------------|----|----|----|---|---|---|---|---|-----|----|----|----|---|---|---|---|
| | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| IP Address : | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | . | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | . | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | . | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Subnet mask : | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | . | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | . | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | . | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Network ID | | | | | | | | | | | | | | | | | | Host ID | | | | | | | | | | | | | | | | |

This Class C network uses 21 Bits for the Network ID (remember, the first 3 bits in the first octet are set) and 8 Bits for the Host ID. The Subnet mask is what splits the Network ID and Host ID. This particular subnet mask is 24 Bits long (consists of 24 one's (1) counting from left side)

Subnet Masks for CIDR Addresses

- To create subnetworks from standard network sizes, must borrow bits to create more networks
- If take bits from host, must change network mask
 - Suppose we want to create eight subnetworks in the 192.168.18.0 address space.
 - How many bits from the host do we need ?

| | | | | |
|----------|----------|----------|--|----------|
| 192 | . 168 | .18 | | .0 |
| 11000000 | 10101000 | 00010010 | | 00000000 |
| Netmask | 11111111 | 11111111 | | 00000000 |

Subnet Masks for CIDR Addresses

- How many bits for 8 subnets?

| | | | |
|----------|----------|----------|----------|
| 192 | . 168 | .18 | .0 |
| 11000000 | 10101000 | 00010010 | 00000000 |
| 11111111 | 11111111 | 11111111 | 11100000 |

How many hosts on each network ?

Netmask

We just add **3 bits** to the length of the 24-bit subnet mask.

Why 3 bits? Binary, because $2^3 = 8$.

So now we have a /27, or 255.255.255.224, subnet mask

How many hosts can we have on each network?

Subnet Masks for CIDR Addresses

- **How many hosts?**
 - Each subnet is limited to $2^5 = 32$ hosts
 - 5 host bits left for creating host addresses
 - But, each subnet is actually limited to 30 hosts
 - **Why is that?**
 - Need 1 address for **base** network address
 - Need 1 address used as **broadcast address** of the subnetwork

Subnet Masks for CIDR Addresses

- Using the /27 mask gives us eight subnets, but we also need to know their numbers if we're going to build a network diagram or a routing table
- The 8 values for each 3-bit subnet are:
 - 000 1
 - 001 0
 - 010 2
 - 011 3
 - 100 4
 - 101 5
 - 110 6
 - 111 7

Then, we add the hosts to each of these networks

Subnets for CIDR Addresses

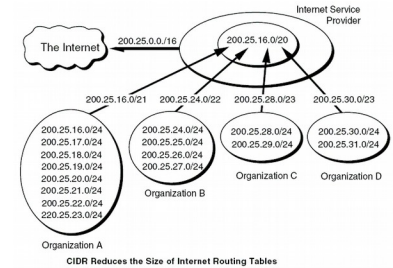
- Starting subnetwork number

- 1 0
- 2 32
- 3 64
- 4 96 (64+32)
- 5 128
- 6 160 (128+32)
- 7 192 (128+64)
- 8 224 (128+64+32)

- Subnetworks are

- 192.168.18.0/27
- 192.168.18.32/27
- 192.168.18.64/27
- 192.168.18.96/27
- 192.168.18.128/27
- 192.168.18.160/27
- 192.168.18.192/27
- 192.168.18.224/27

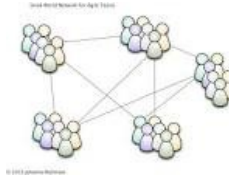
Subnets for CIDR Addresses



- After figuring out the subnetworks base addresses,
 - Figure out the network mask if you haven't already done that
 - Figure out the broadcast address for each subnetwork
 - Figure out the range of hosts assigned to each subnetwork
- Worksheet on this, plus lab on subnetting
- Look at the links for help, end of lecture

CIDR and Subnets Created Flexible Network Sizes

- Helped with running out of address space
 - Blocks of addresses can be assigned to networks as small as **32 hosts**



- **Or, over 500,000 hosts**
- Allows for address assignments that much more closely fit an organization's specific needs
- A single high-level route entry can represent many lower-level routes in the global routing tables



Example of Subnetting

- How to figure out your broadcast address ...

<https://www.countryipblocks.net/identifying-the-network-and-broadcast-address-of-a-subnet>

- URLs for Subnet Calculator

<http://www.subnetonline.com/pages/subnet-calculators/ip-subnet-calculator.php>

<http://www.techzoom.net/tools/network-subnet-calculator.en>



Router Table Aggregation

CIDR Also Helped with Route Aggregation

- Currently, big blocks of addresses assigned to large Internet Service Providers (ISPs)
- Re-allocate portions of their address blocks to their customers
 - Assigns its customers CIDR addresses from that block
 - Customers, smaller ISPs, and in turn re-allocate portions of their address block to their users
 - Yet global routing tables for all these networks can be represented by single route entry

CIDR Also Helped with Route Aggregation or Supernetting

- CIDR provides routing prefix aggregation, also known as **supernetting**
 - Example: **Sixteen Contiguous /24 Networks**
 - Aggregated and advertised as a single /20 route
 - If first 20 bits of their network addresses match!!
 - Two aligned contiguous /20s may then be aggregated to a /19, and so forth
 - Allows a significant reduction in the number of routes that have to be advertised

CIDR Also Helped with Route Aggregation or Supernetting

- **Example:** Want to aggregate 8 network addresses between 131.0.0.0/16 and 131.7.0.0 /16
- Need subnet mask that makes all 8 network addresses appear to be on same single network
 - **How many bits do we need to supernet 8 networks?**
 - 2^3 for 8 networks = 3 bits
 - /16 network has a subnet mask of 255.255.0.0
 - We **steal** 3 bits from network portion of subnet mask, we end up with the mask 255.248.0.0
 - Think of it as **Shortening** the mask

Supernetting

<http://www.2000trainers.com/cisco-ccna-05/ccna-classless-cidr-supernetting/>

- So, range can now be designated as 131.0.0.0/13
This value aggregates all IP addresses between 131.0.0.1 and 131.7.255.254

| | | | | | | |
|-----------------|----------|-------|-----|----------|----------|----------------|
| /13 Subnet Mask | 11111111 | 11111 | 000 | 00000000 | 00000000 | |
| | Network | | | | | |
| 131.0.0.0 | 10000011 | 00000 | 000 | 00000000 | 00000000 | |
| 131.1.0.0 | 10000011 | 00000 | 001 | 00000000 | 00000000 | |
| 131.2.0.0 | 10000011 | 00000 | 010 | 00000000 | 00000000 | |
| 131.3.0.0 | 10000011 | 00000 | 011 | 00000000 | 00000000 | |
| 131.4.0.0 | 10000011 | 00000 | 100 | 00000000 | 00000000 | |
| 131.5.0.0 | 10000011 | 00000 | 101 | 00000000 | 00000000 | |
| 131.6.0.0 | 10000011 | 00000 | 110 | 00000000 | 00000000 | |
| 131.7.0.0 | 10000011 | 00000 | 111 | 00000000 | 00000000 | |
| Network ID | 10000011 | 00000 | 000 | 00000000 | 00000000 | = 131.0.0.0/13 |

Reserved Addresses

Two Reserved Addresses

127.0.0.1 – What is this called?

0.0.0.0 – Also reserved

Can we route to these normally?

Blocks of Private Addresses

10.0.0.0/8 (10.0.0.0 to 10.255.255.255)

172.16.0.0/12 (172.16.0.0 to 172.31.255.255)

192.168.0.0/16 (192.168.0.0 to 192.168.255.255)

169.254.0.0/16 (169.254.0.0 to 169.254.255.255)

Small companies use: 172.16.0.0

Home users use: 192.168.0.0

Subnet References

- **Packet Fragmentation**

http://www.cisco.com/en/US/tech/tk827/tk369/technologies_white_paper09186a00800d6979.shtml

- **Subnetting**

<http://microcomputer-network.net/calculate-subnet-mask>

- **Networking 101 – Understanding Subnets**

<http://www.enterprisenetworkingplanet.com/netsp/article.php/3566521/Networking-101-Understanding-Subnets-and-CIDR.htm>

- **More Subnetting**

<http://learn-networking.com/network-design/how-to-subnet-a-network>

- **Subnetting Made Easy**

<http://www.techexams.net/forums/ccna-ccent/38772-subnetting-made-easy.html>



DHCP

NAT and DHCP

Helping with IP Address Shortage

NAT

Network Address Translation

- NAT developed by Cisco
 - Documented in RFC 1631, 1994
<http://www.internetsociety.org/articles/retrospective-view-nat>
- **Three Types of NAT**
 - **Static** – 1 to 1 mapping of internal to external address (won't cover this)
 - **Dynamic** – Internal mapped to a pool of external addresses
 - **Overloading or Port Addressing** – Multiple internal mapped to 1 external address (home routers)

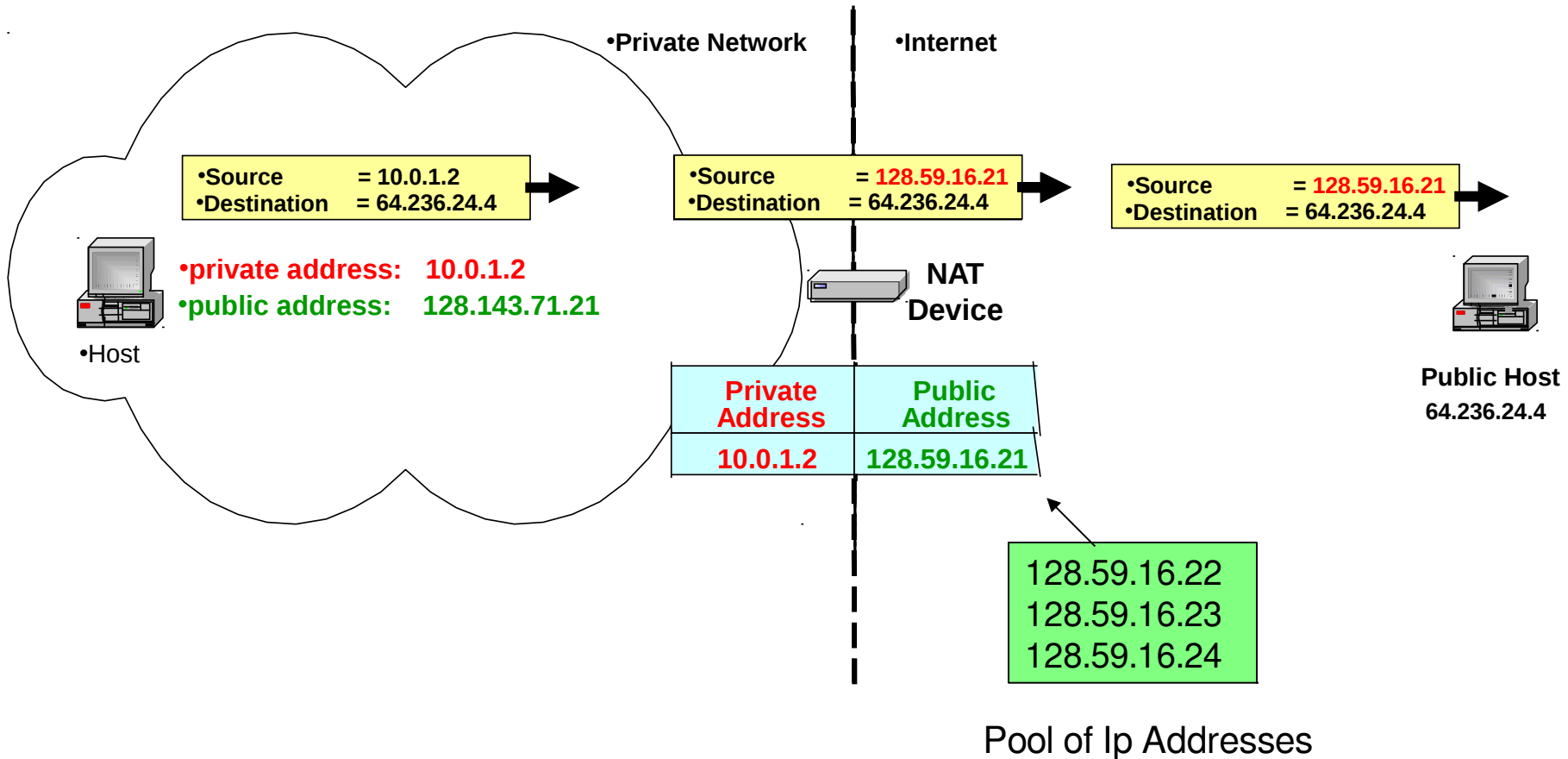
NAT Dynamic

- Internal network has internal IP numbers not from IANA
 - These numbers are non-routable
 - Internal computer wants to talk to web-server
 - Router puts Internal IP in an Address Translation Table (ATT)
 - Router maps internal IP to external IP from pool
 - Incoming traffic is mapped back to Internal IP address according to ATT

Example of Dynamic NAT

- **Scenario:** Corporate network has many hosts but small number of public IP addresses
- **NAT solution**
 - Corporate network has private addresses
 - NAT device, located at boundary between corporate and public Internet, manages a pool of public IP addresses
 - When corporate network host sends IP datagram to host in public Internet, NAT device picks public IP address from address pool, and binds this address to private address of host

Pooling of IP addresses



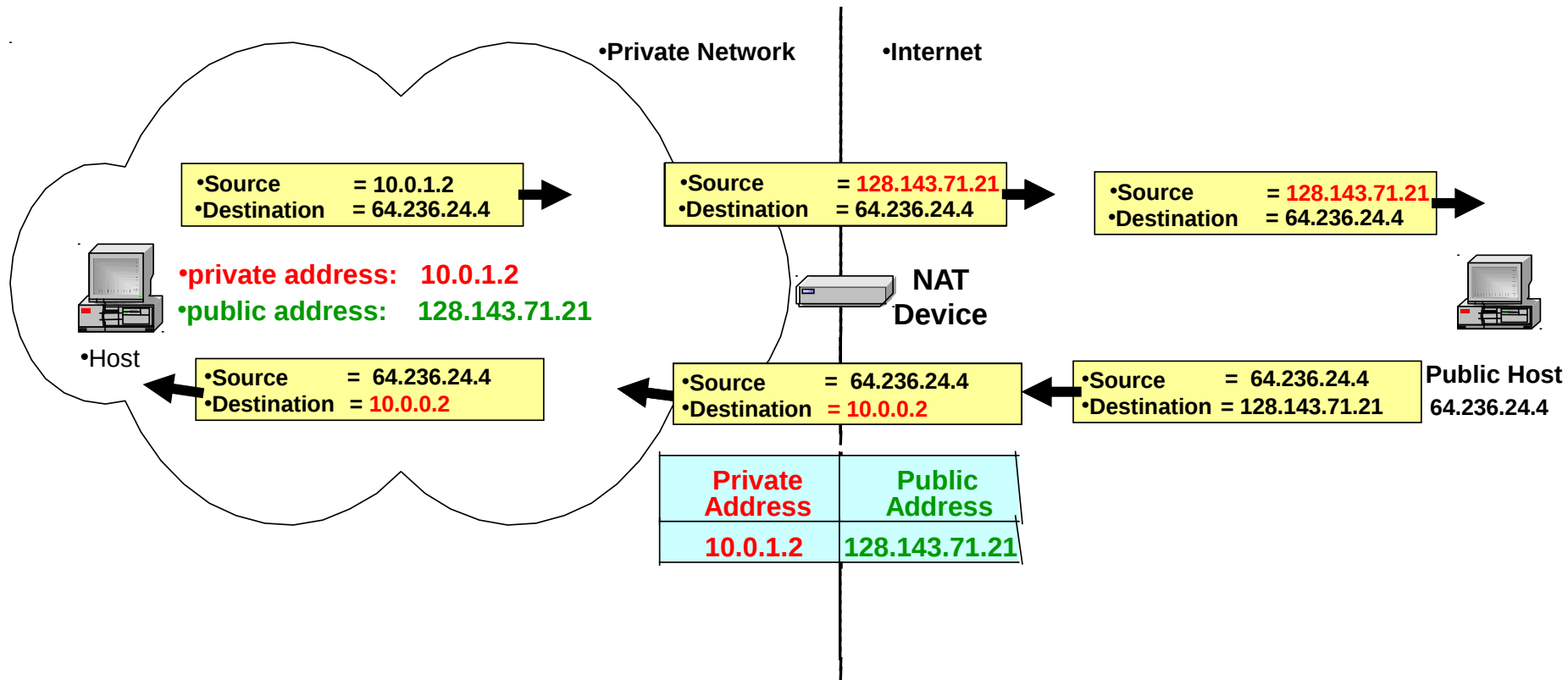
Overloading NAT

- **Translation Table also Saves Port Numbers**
 - Port numbers are saved from internal computers
 - Port number + IP address now distinguishes internal computers
 - All internal computers are mapped to a single external IP address

Overloading NAT Example

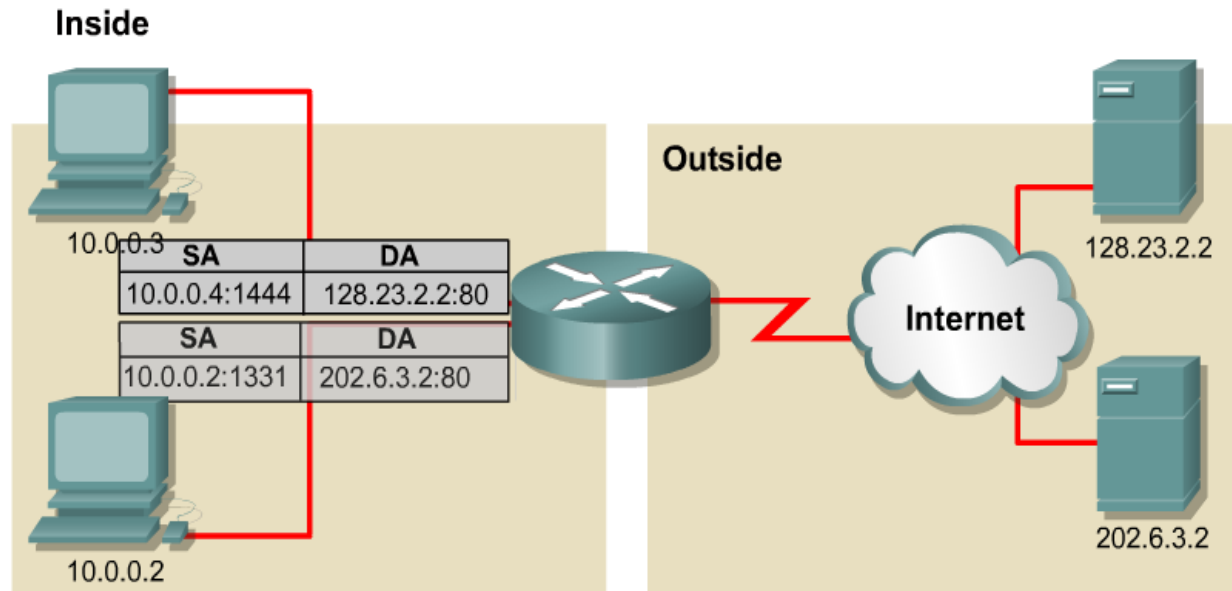
- **Also called**
 - **Network Address and Port Translation (NAPT),
Port Address Translation (PAT)**
- **Scenario**
 - Single public IP address is mapped to multiple hosts in a private network
- **NAT solution**
 - Assign private addresses to the hosts of the corporate network
 - NAT device modifies the port numbers for outgoing traffic

NAT Overloading



- NAT device stores address and port translation tables
- In this example we mapped only addresses.

PAT – Port Address Translation



| NAT Table | | | |
|-------------------------|--------------------------|--------------------------|------------------------|
| Inside Local IP Address | Inside Global IP Address | Outside Local IP Address | Outside Global Address |
| 10.0.0.2:1331 | 179.9.8.20:1331 | 202.6.3.2:80 | 202.6.3.2:80 |
| 10.0.0.3:1555 | 179.9.8.20:1555 | 128.23.2.2:80 | 128.23.2.2:80 |

- PAT modifies the TCP/UDP source port to track inside Host addresses.
- Tracks and translates SrcAddr, DstAddr and SrcPort (which uniquely identifies each connection) for each stream of traffic.

NAT Discussion

- NAT is good for
 - Saving IP addresses
 - Hiding internal IP addresses from outside
 - Functions like a crude firewall
 - Do not rely on NAT for your security !!!
 - Load balancing between servers
- NAT has some problems
 - Has to recalculate headers and checksum
 - Breaks the end-to-end model of the TCP/IP stack – computer to computer
 - Fragments must be carefully dealt with

Dynamic Assignment of IP addresses

- Dynamic assignment of IP addresses is desirable
- Do you know what this protocol is called?
 - IP addresses are assigned on-demand
 - Avoid manual IP configuration
 - Support mobility of laptops
 - Wireless networking and Home NATs

Dynamic Host Configuration Protocol (DHCP)

- Designed in 1993
- Requires a server and free IP address space
- Supports temporary allocation (“leases”) of IP addresses
- DHCP client keeps address for a short time
- DHCP operates in most wireless routers
 - How the internal IP addresses distributed

Reference

<http://www.thegeekstuff.com/2013/03/dhcp-basics/>

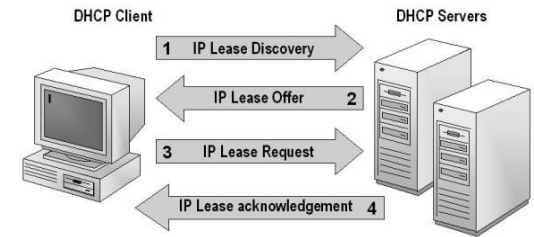
DHCP Overview



1. Devices reach out to local network to discover any available DHCP server
2. Servers manage pools of valid addresses and assign addresses out of those pools
3. DHCP uses idea of leasing - time when IP address will be valid
 - Lease time can vary depending on how long user is likely to require Internet connection

Devices release addresses when their leases expire, request renewal from DHCP server if they are staying online

DHCP Message Type

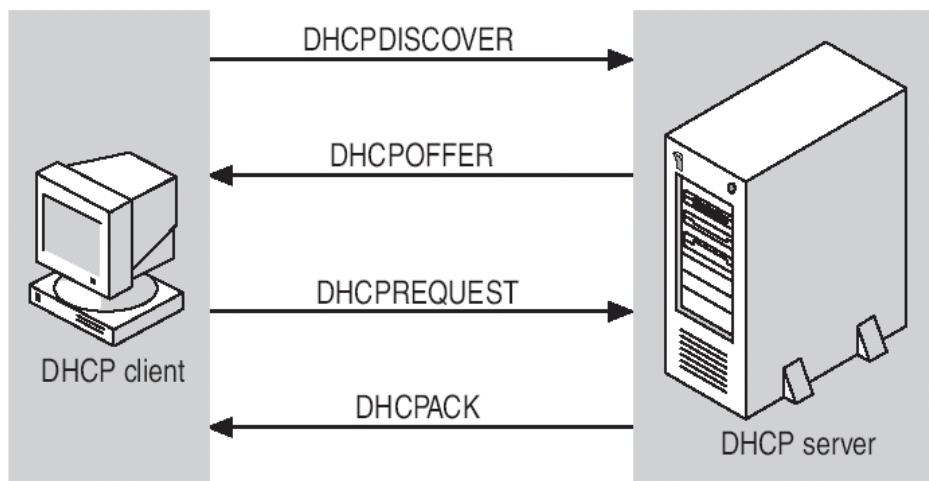


Message Types

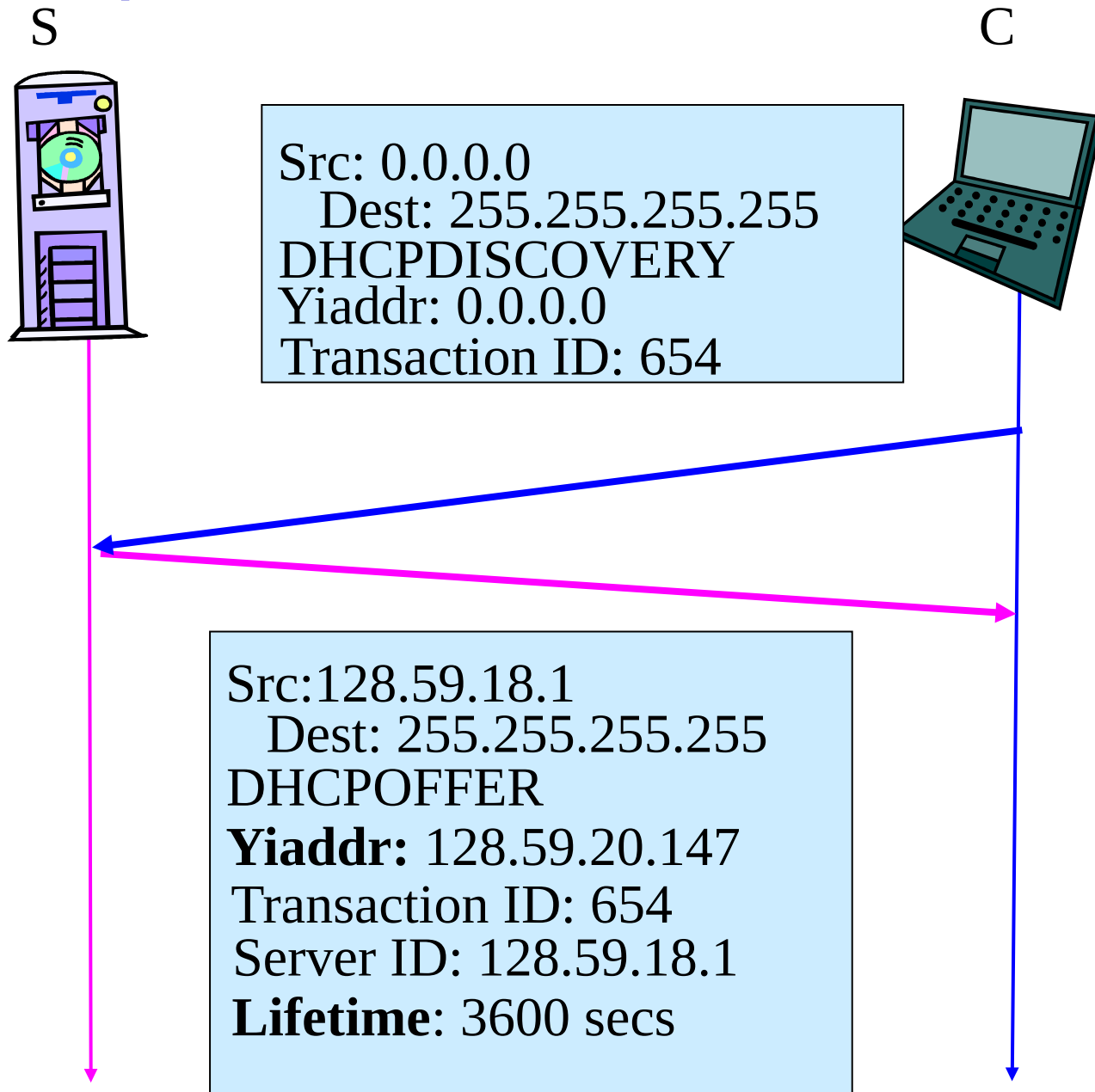
- 1. DHCPDiscover
- 2. DHCPOffer
- 3. DHCPRequest
- 4. DHCPDecline
- 5. DHCPAck
- 6. DHCPNak
- 7. DHCPRelease
- 8. DHCPInform

More on DHCP operations

- Begins with
 - DHCP Discover from client
 - DHCP Offer from a DHCP Server
 - DHCP Request – accepts offer
 - DHCP Ack from server
- Example follows



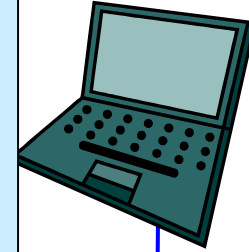
DHCP operations



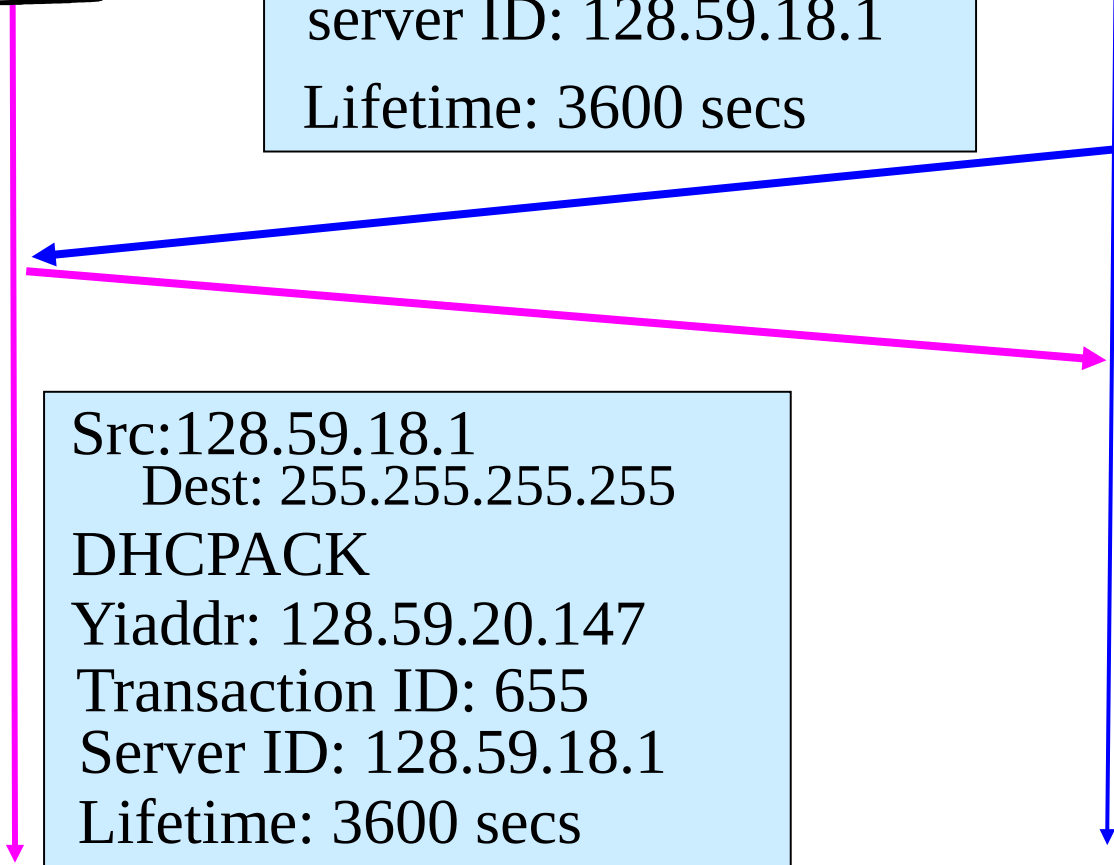
DHCP Operations



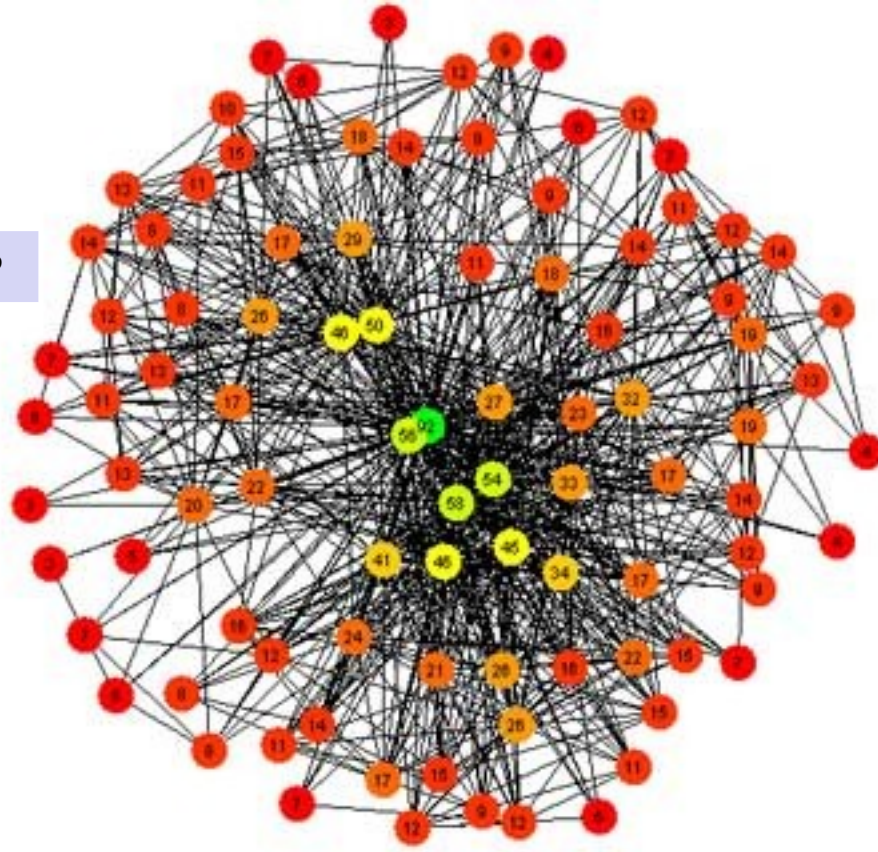
Src: 0.0.0.0
Dest: 255.255.255.255
DHCPREQUEST
Yiaddr: 128.59.20.147
Transaction ID: 655
server ID: 128.59.18.1
Lifetime: 3600 secs



Src:128.59.18.1
Dest: 255.255.255.255
DHCPCACK
Yiaddr: 128.59.20.147
Transaction ID: 655
Server ID: 128.59.18.1
Lifetime: 3600 secs



Routing?



Next

Read: Continue with Chapter 4