

# Applied Networks & Security

Internet Protocol – with Critical Analysis

<http://condor.depaul.edu/~jkristof/it263/>

John Kristoff  
[jtk@depaul.edu](mailto:jtk@depaul.edu)

# Critical analysis disclaimer

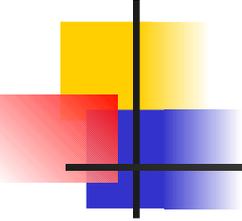
Following this disclaimer are slides used in other versions of the course. We *mark up* some slides using ~~strikethroughs~~ and underlined red in comic sans ms 20pt font. This is not meant to slight other teachers or their material. Much of the material is good and helpful so we use it.

We do this to explore complex issues, refresh dated material, correct inaccuracies and stimulate critical thinking. In some cases we are pedantic where it seems useful, but we are not exhaustive and try to avoid being overly tedious when it is unnecessary.

**IT 263**

**Applied Networks and  
Security**

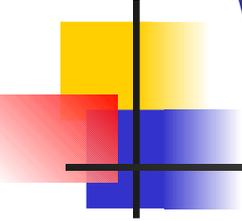
IP Networking



# Topics

---

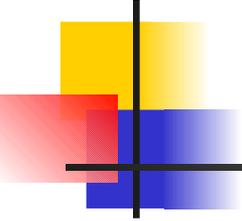
- IP Routing
  - What do routers do?
  - Address Assignment - DHCP
  - Subnets
  - Network Address Translation (NAT)
- Domain Name System (DNS) services
- TCP/IP Utility programs
  
- Read chapters 3 and 5



# What do routers do?

---

- **Store and Forward packets based on destination IP address**
- Interconnect multiple IP subnets
- Provide a gateway between different physical networks.
- Provide a boundary for network security and management
- May also provide other services:
  - Firewall services
  - Traffic prioritization (QOS)

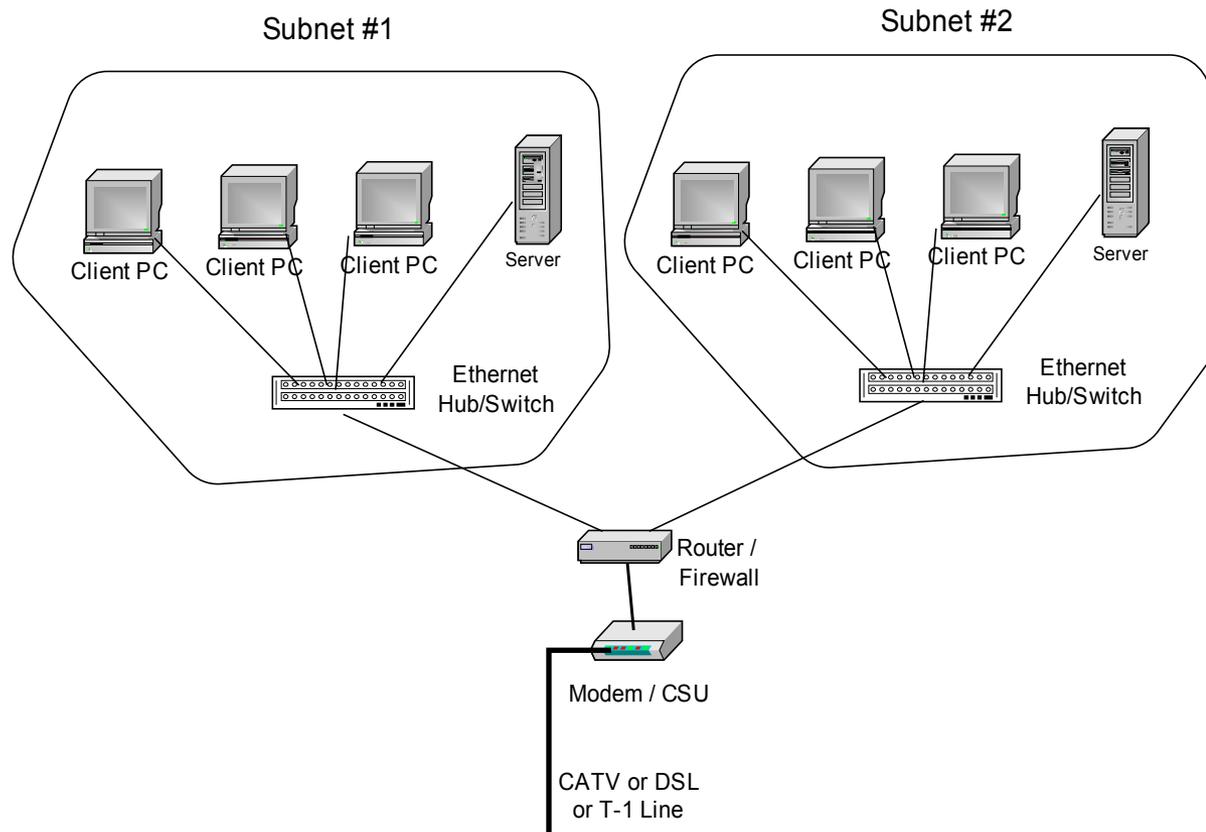


# Where do routers go?

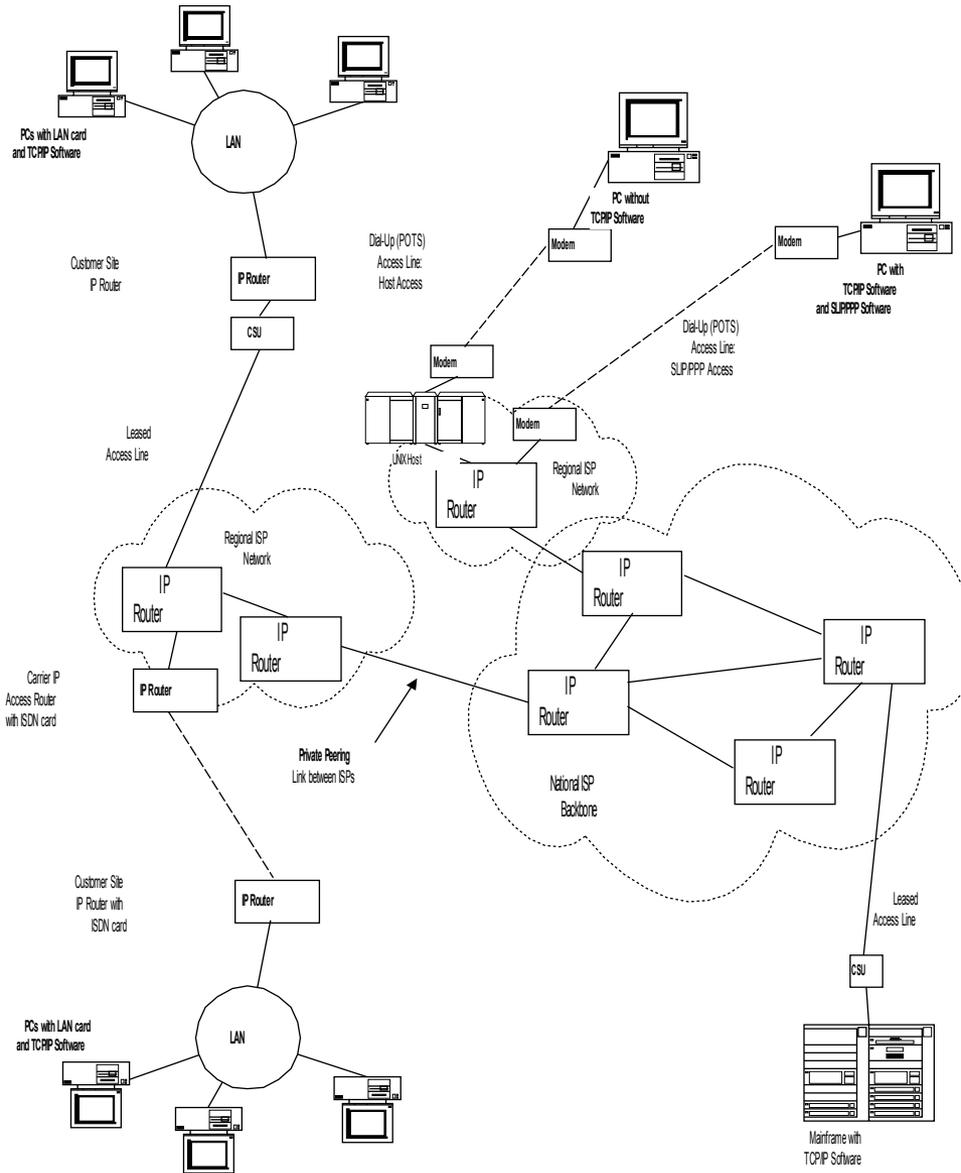
---

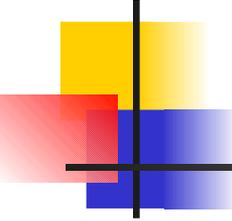
- Each router port connects to a hub, switch or another router.
- Internal routers interconnect LANs within an organization (Intranet).
- An access or WAN router is used to connect LANs into a WAN network (Usually private WAN)
- Border routers provide access to the Internet – Connects to the ISP router.

# SOHO router



# Internet Service

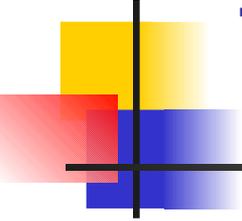




# IP Addresses

---

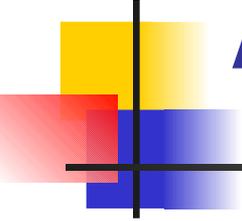
- Each **IP address** is **4 bytes** long
- Dotted decimal notation
  - Each byte (8 bits) is written in decimal separated by dots, like
  - Each of the 4 values is in range 0 - 255.
  - Example: 150.21.39.52



# IP Addresses

---

- **IP address** contains 3 parts (*not that simple*)
  - An **IP Network** part that is used by Internet backbone routers to deliver packets to a particular Network.
  - An optional **IP Subnet** part that is used by access and internal routers within an IP Network to deliver packets to a particular Subnet.
  - An **IP Host** part that identifies a particular individual device on the subnet.



# Address Example

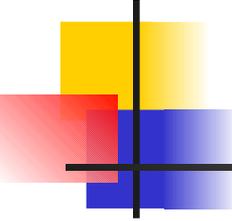
---

Network	Subnet	Host	
1 3 0	8 8	5 5	1 2

Network = 130.88

Subnet = 55

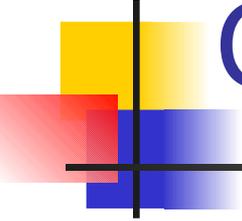
Host = 12



# Decimal vs binary notation

---

- To simplify “human” reading of IP addresses, we use decimal notation:
  - 130.88.55.12
- The routers view an IP address as binaries:
  - 10000010.01011000.00110111.00001100

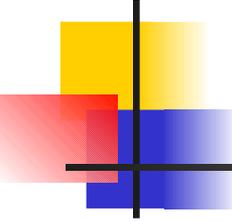


# Cheat sheet #1

---

- A set of N bits can store  $2^N$  different values

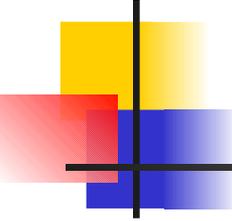
$2^0 = 1$	$2^1 = 2$	$2^2 = 4$
$2^3 = 8$	$2^4 = 16$	$2^5 = 32$
$2^6 = 64$	$2^7 = 128$	$2^8 = 256$



# In class examples

---

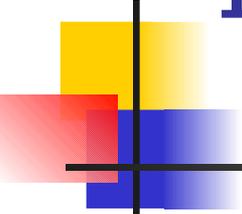
- Translate to decimal:
  - 00000001.00000010.00000011.00000111
  - 10010110.00100011.11111110.00010011
- Translate to binary:
  - 3.5.1.64
  - 128.65.49.6
  - 195.97.66.227



# IP Classes

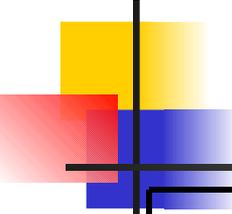
---

- **IP Networks** are of various sizes based on whether they are Class A, Class B or Class C.
- Address Class can be determined by value in 1<sup>st</sup> byte of the IP address
  - Class A = First half of all IP addresses.
    - 1<sup>st</sup> byte in range [0 - 127]
  - Class B = Half of remaining addresses.
    - 1<sup>st</sup> byte in range [128 - 191]
  - Class C = Half of remaining addresses.
    - 1<sup>st</sup> byte in range [192 - 223]
- In classfull IP addressing, you know what portion of IP address is the network address by just looking at the IP address the first octet will tell you the class.
- Important note: In today's network, you will almost never see classfull networks, you will need to rely on the subnet mask (more on that later) to know what part of the IP address is the network portion and what part is the host portion.



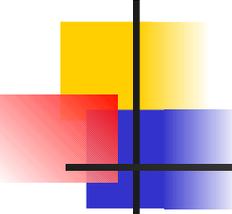
# IP Address Classes

Address Class	Network Prefix Size	Value in 1 <sup>st</sup> byte
A	1 byte	1-127
B	2 bytes	128-191
C	3 bytes	192-223



# IP Network Sizes

Address Class	Number of Networks (A)	Maximum Hosts per Network (B)	Total IP Addresses (A*B)
A	128	$256^3 = \sim 16$ million	$\sim 2$ billion
B	$64 * 256 = 16,384$	$256^2 = \sim 65,000$	$\sim 1$ billion
C	$32 * 256 * 256 = \sim 2$ million	256	$\sim 1/2$ billion



# Network address/host address

---

- IP addresses have 2 (or 3 depending how you look at it) parts:
  - Network and subnet number
  - Host number
- If I have an IP address, how do I know what part is the network or subnet @ and what part is the host @?
  - By looking at the subnet mask.
- 2 ways of documenting it:
  - Classic Mask notation: 255.255.0.0 or 11111111.11111111.00000000.00000000 – The “1” designate the network/subnet ID and the “0” the host addresses
  - Canonical notation: /16 – means there are 16 “1” in the mask.
- The subnet mask can be either explicit or implied.
  - In classfull addressing it is implied:
    - Class A: first octet is network ID: 255.0.0.0 or /8
    - Class B: first two octets are network ID: 255.255.0.0 or /16
    - Class C: first three octets are network ID: 255.255.255.0 or /24
    - **No, in classful it is determined by the few few bits**
  - In classless: you need to know the mask
  - In classfull you main network may be subnetted: you then also need to know the mask

# Network address/host address

## Examples

---

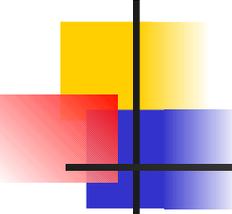
- Question: What is the network number and host number of the following:
- Classfull:
  - 10.5.6.8
  - 172.45.33.33
  - 200.100.99.88
- Classless
  - 12.4.20.30 – 255.255.255.0
  - 205.44.55.66 / 16
  - 199.56.89.54 / 8
  - 10.20.30.40 /8
  - 10.20.30.40 – 255.255.0.0
  - 10.20.30.40 /24

# Network address/host address

## Examples more tricky!

---

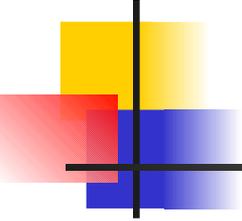
- Classless or classfull with subnets:
  - 16.24.44.129 – 255.255.255.128
  - 10.55.17.253 / 18
  - 2.3.4.5 – 255.255.255.252
  - More examples???
- More... In homeworks 😊



# IP Addresses

---

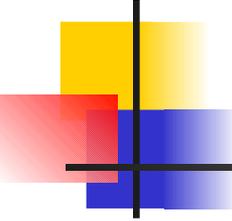
- How does every device on the Internet get a **different** IP address?
- **IP Network** prefixes are assigned to organizations by the ~~Internet Corporation for Assigned Names and Numbers (ICANN)~~ - **NO!** *Assigned by ISP, RIR or IANA if you qualify*
  - Who?: ISPs, Universities, large businesses
- These organizations then control all IP addresses starting with that IP Network prefix
- In the “Old times” (9+ years ago) Classfull networks were assigned. Today: classless.
- As an organization, can you get your own IP subnet today?



# Other Ways to get IP Addresses

---

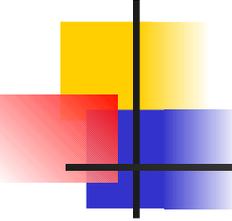
- Small businesses and individuals usually don't go through the ICANN - *nobody does, ICANN doesn't do that*
  - Small businesses are loaned a group of IP addresses by their ISP
  - Home PCs can get a dynamically assigned (DHCP) address from ISP on each connection.
  - SOHO routers can share a single DHCP address from ISP with multiple internal devices by providing a Network Address Translation (NAT) service.



# Static IP Addresses

---

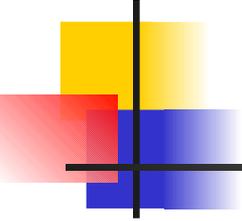
- A Static IP address is permanently assigned to a particular host device and typically must be manually configured into PC or device.
- Network/Subnet IP address values must be correct for the subnet on which this PC is located.
- Also, to be able to communicate outside its own network, a “default gateway” must be set correctly. Important aspect: the default gateway should be set to a value on your own subnet!
- Other parameters may need to be properly set: DNS, application specific values (phone # for example).



# IP Address Assignment

---

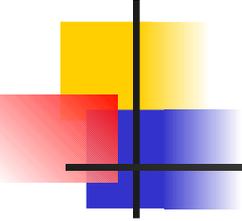
- DHCP (Dynamic Host Configuration Protocol) Server
  - When PC powers up, sends broadcast to DHCP server.
  - DHCP server dynamically allocates an IP address from pool of IP addresses for that subnet.
  - Address has a renewal time associated
  - Server can assign IP address, DNS server address, local router address, and other parameters to PC.



# DHCP Parameters

---

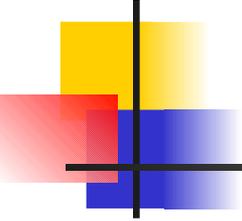
- DHCP server passes the following information to each host :
  - IP Address for the host machine
  - Subnet mask for its subnet
  - Gateway address for its nearest router
  - DNS server address for domain name resolution
  - Anything else that might be useful, such as telephone number and features for IP phone.



# Address Resolution Protocol (ARP)

---

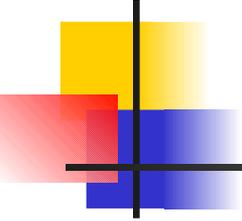
- **A broadcast protocol used to determine the MAC address corresponding to a known IP address**
  - *ARP Request* containing IP address is broadcast on a subnet (using the MAC all 1s broadcast destination address).
  - *ARP Reply* is sent by device that recognizes its IP address in the ARP Request.
  - IP Address/MAC Address pairs are stored in **ARP Table** by the sender so ARP Request does not need to be re-sent.



# PC Packet Delivery

---

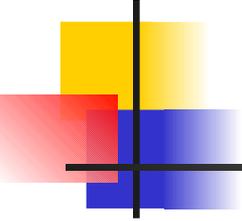
- When a PC has an IP packet to send, it does the following:
  - Check whether destination IP address is on the same subnet or a different subnet as the sender.
    - If destination IP on same subnet:
      - Look up destination IP address in **ARP Table**, get MAC address for destination
    - If destination on different subnet:
      - Look up default gateway address in **ARP table**, get MAC address for next router.
  - Forward the IP packet out appropriate Ethernet port with Ethernet destination found by ARP.



# What do routers do?

---

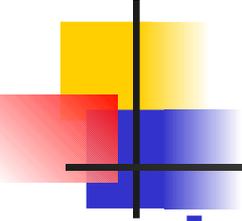
- **Store and Forward packets based on destination IP address**
- **Interconnect multiple IP subnets**
- Provide a gateway between different physical networks.
- Provide a boundary for network security and management
- Firewall services (packet filtering)



# Routers vs. Switches

---

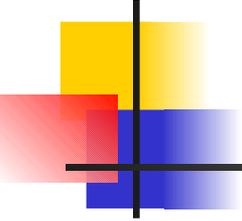
- Both IP routers and Ethernet switches receive data packets, store them in a buffer, and forward them towards their destination, but there are differences:
  - **Address:** switches look at destination Ethernet MAC address, routers look at destination IP address.
  - **Path learning:** switches build Forwarding table by looking at source addresses in passing packets, routers build Routing table by exchanging Route Table Update messages with other routers.



# Routers vs. Switches

---

- **Path complexity:** switches assume there is only one path to each destination; routers assume there are many paths and keep track of a distance metric for each to choose the best one.
- **Security:** switches assume that devices mainly trust one another, routers assume not and implement **Access Lists** which can be used to block any IP source/dest combinations. - *not quite right*
- **Broadcasts:** switches will forward broadcast packets out all ports, router do not typically re-send broadcasts. - *they can forward directed broadcasts*
- **Scalability:** switches can scale up to **thousands of devices** by storing one Forwarding Table address entry for each device, routers can scale up to **thousands of subnets** by storing one Route Table entry for each subnet (with hundreds or thousands of devices per subnet = millions of devices addressed).

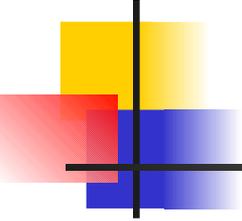


# Routers vs. Switches

---

- Switch Forwarding Table

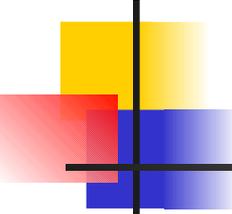
<b>Dest. Address</b>	<b>Port</b>	<b>Update Time</b>
1A:2C:33:44:2D:1E	1	15:01:0333 pm
1A:2C:33:48:26:34	1	15:01:0433 pm
1A:2C:33:2D:EE:29	3	15:01:0363 pm
1A:2C:33:46:2C:BB	2	15:01:2233 pm



# Routers vs. Switches

- Router Table (*last distance = connected?*)

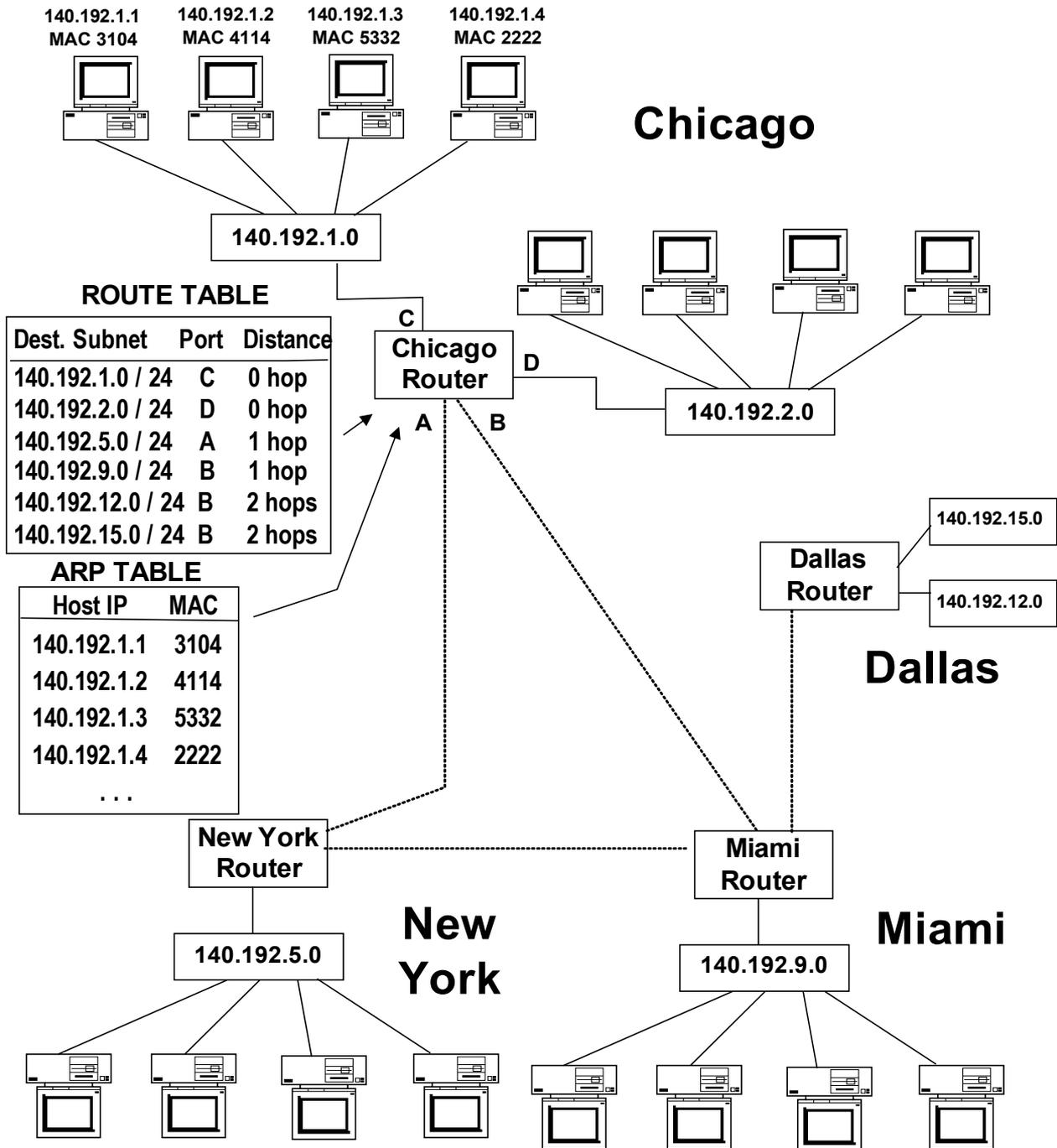
<b>Dest. Subnet</b>	<b>Port</b>	<b>Distance</b>	<b>Next Router</b>
150.193.0.0 / 16	S1	2 hops	135.16.22.1
140.192.13.0 / 24	E1	2 hops	114.38.220.2
140.192.14.0 / 25	E2	1 hop	140.192.14.1
140.192.14.128 / 25	E3	1 hops	140.192.14.129
0.0.0.0 / 0	S2		162.22.39.1

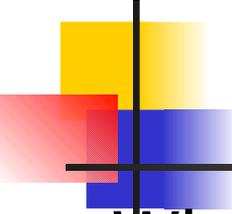


# IP Router Tables

---

- Routers *might* keep two types of address tables:
  - ARP Table for each port
    - Contains (IP addr, MAC addr) for each host on local subnets
    - Used to look up the matching Ethernet MAC address for a give IP destination address
    - If not found in ARP Table, then MAC values filled in using ARP Broadcast
    - *Depending on link technology ARP may not be needed, ARP isn't inherently necessary, but probably is in most networks*
  - Route Table
    - Contains (Network/Subnet ID, Router Port, Distance) for other reachable subnets

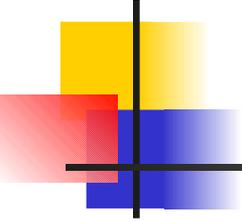




# Router Packet Delivery

---

- When a router has an IP packet to send, it does the following:
  - Find Route Table entry with the longest match of IP network/subnet prefix entry with packet destination address
    - If destination IP on local subnet (distance = 0):
      - Look up Destination IP address in **ARP Table**, get MAC address for destination
    - If destination on nonlocal subnet (distance > 0):
      - Look up Next Router address in **ARP table**, get MAC address for next router.
    - *What about interfaces (e.g. serial point-to-point) w/o a MAC address? They don't need ARP.*
  - Forward the IP packet out appropriate port with appropriate header for outgoing network.



# What do routers do?

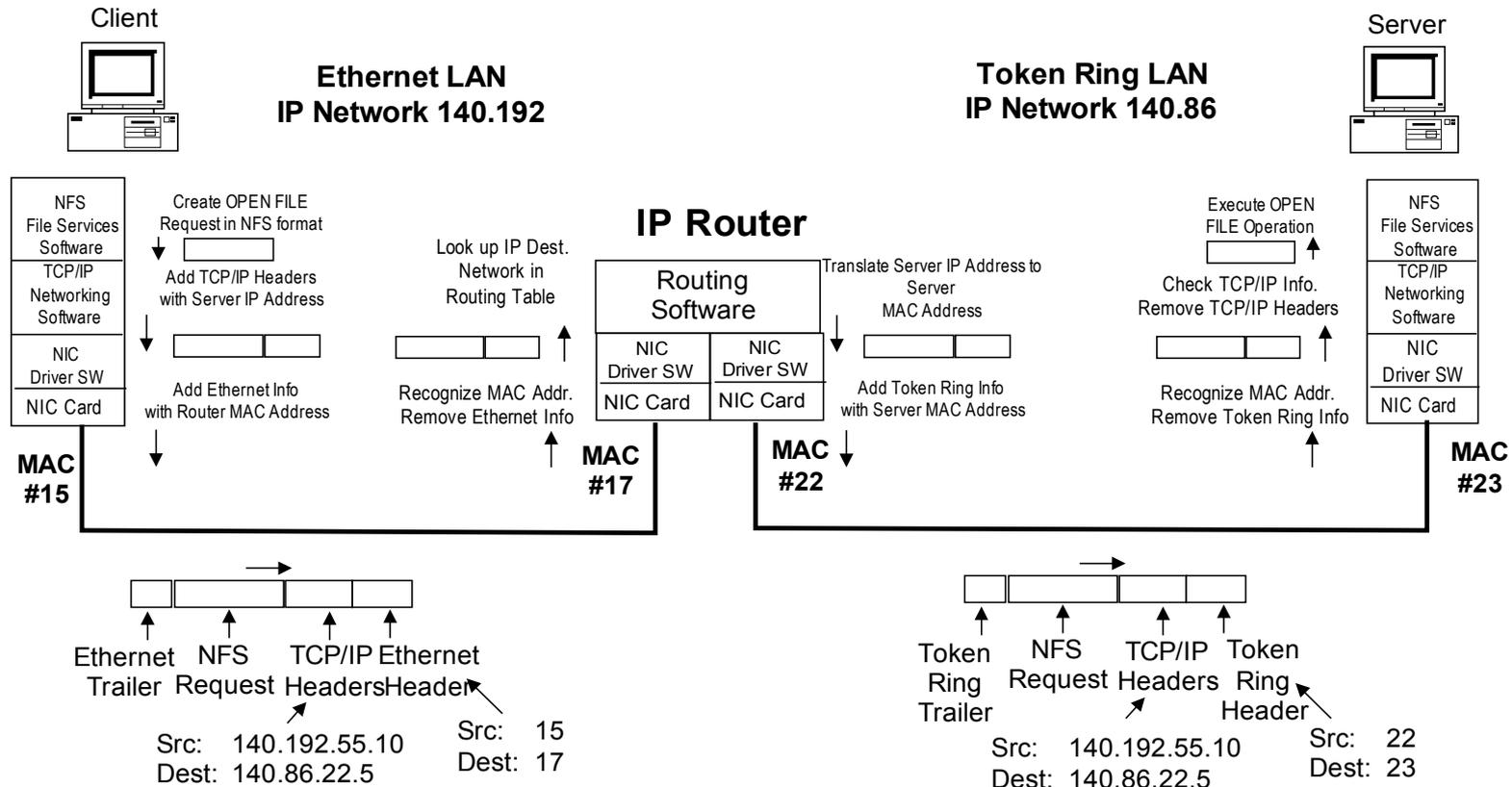
---

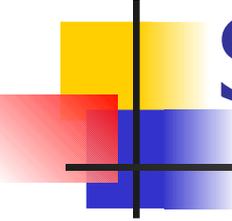
- Store and Forward packets based on destination IP address
- Interconnect multiple IP subnets
- **Provide a gateway between different physical networks.**
- Provide a boundary for network security and management
- Firewall services (packet filtering)

# IP Routing Example

IP Address 140.192.55.10

IP Address 140.86.22.5

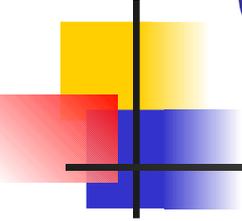




# Subnetting IP Networks

---

- You can break an IP Network (or subnet) into smaller subnets by choosing some additional bits in the address to be used as Subnet bits.
- IP addresses of all hosts within a subnet have identical IP network/subnet prefix.
- Key issue: how many subnets are needed?
  - By using  $X$  Subnet bits per address, you get  $2^X$  unique subnets
  - Example: if Subnet field is 6 bits long, then  $2^6 = 64$  subnets can be addressed by internal routers.

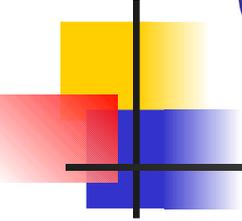


# Cheat sheet #2

---

- Equivalent values: #routing bits vs subnet mask

/16 = 255.255.0.0	/17 = 255.255.128.0
/18 = 255.255.192.0	/19 = 255.255.224.0
/20 = 255.255.240.0	/21 = 255.255.248.0
/22 = 255.255.252.0	/23 = 255.255.254.0

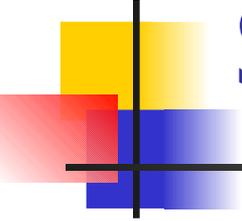


# Cheat sheet #2b

---

- Equivalent values: #routing bits vs. subnet mask

/24 = 255.255.255.0	/25 = 255.255.255.128
/26 = 255.255.255.192	/27 = 255.255.255.224
/28 = 255.255.255.240	/29 = 255.255.255.248
/30 = 255.255.255.252	/31 = 255.255.255.254

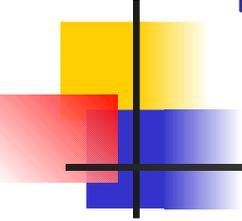


# Subnet Mask Example

---

- A Company is assigned Class B address 130.88 and decide they want 256 subnets.
- What subnet mask should they use?

# Subnet Mask Example

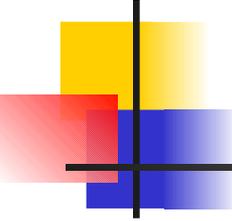


---

	Network		Subnet		Host		
	<b>130</b>		<b>88</b>		<b>30</b>		<b>100</b>
<u>Subnet Mask:</u>	<u>11111111</u>		<u>11111111</u>		<u>11111111</u>		<u>00000000</u>

**Subnet Mask: 255.255.255.0**

**Full Subnet: 130.88.30.0 / 24**

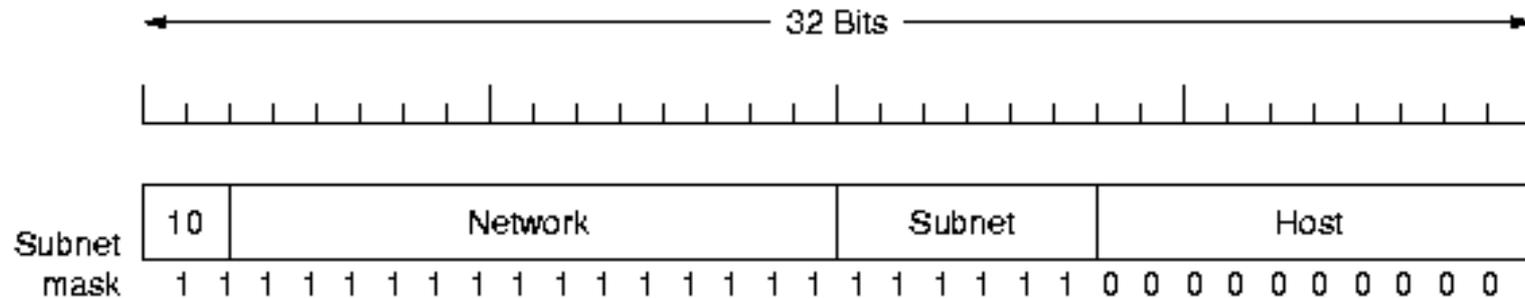


# Subnet Mask Example

---

- Another example. An ISP owns a Class B address. One of their customers, JoyCo, wants a block of 1000 IP addresses.
- What subnet mask does the ISP program into their routers so that JoyCo's block of addresses is routed correctly?

# Subnet Mask Example



**Subnet Mask: 255.255.252.0**

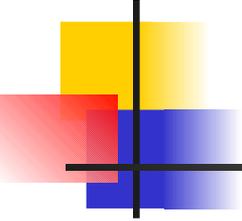
16 Network bits, 6 Subnet bits, 10 host bits

Number of Subnets =  $2^6 = 64$

Number of Hosts per Subnet =  $2^{10} - 2 = 1022$

Cisco Notation: X.Y.Z.0 / 22

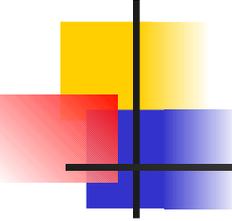
*Which leaves almost no room for expansion and further subnetting*



# Another Example

---

- ISP X is allocated Class C IP Network 202.55.16.0 / 24 by the IANA.
  - *They would get it via an RIR today*
- ISP X splits these 256 addresses as follows:
  - 202.55.16.0 / 25 – 128 IP addresses for DSL customer DHCP
  - 202.55.16.128 / 26 – 64 IP addresses for SOHO customer DHCP
  - 202.55.16.192 / 26 – 64 static IP addresses allocated to JonesCo.
    - *Each time they split, they lose two to network and broadcast address*
- JonesCo further splits its 64 addresses into 4 subnets:
  - 202.55.16.192 / 28 – 16 static IP addresses for company servers
  - 202.55.16.208, 224, 240 / 28 – 3 subnets of 16 each for 3 internal departments – Product Design, Marketing and Sales.



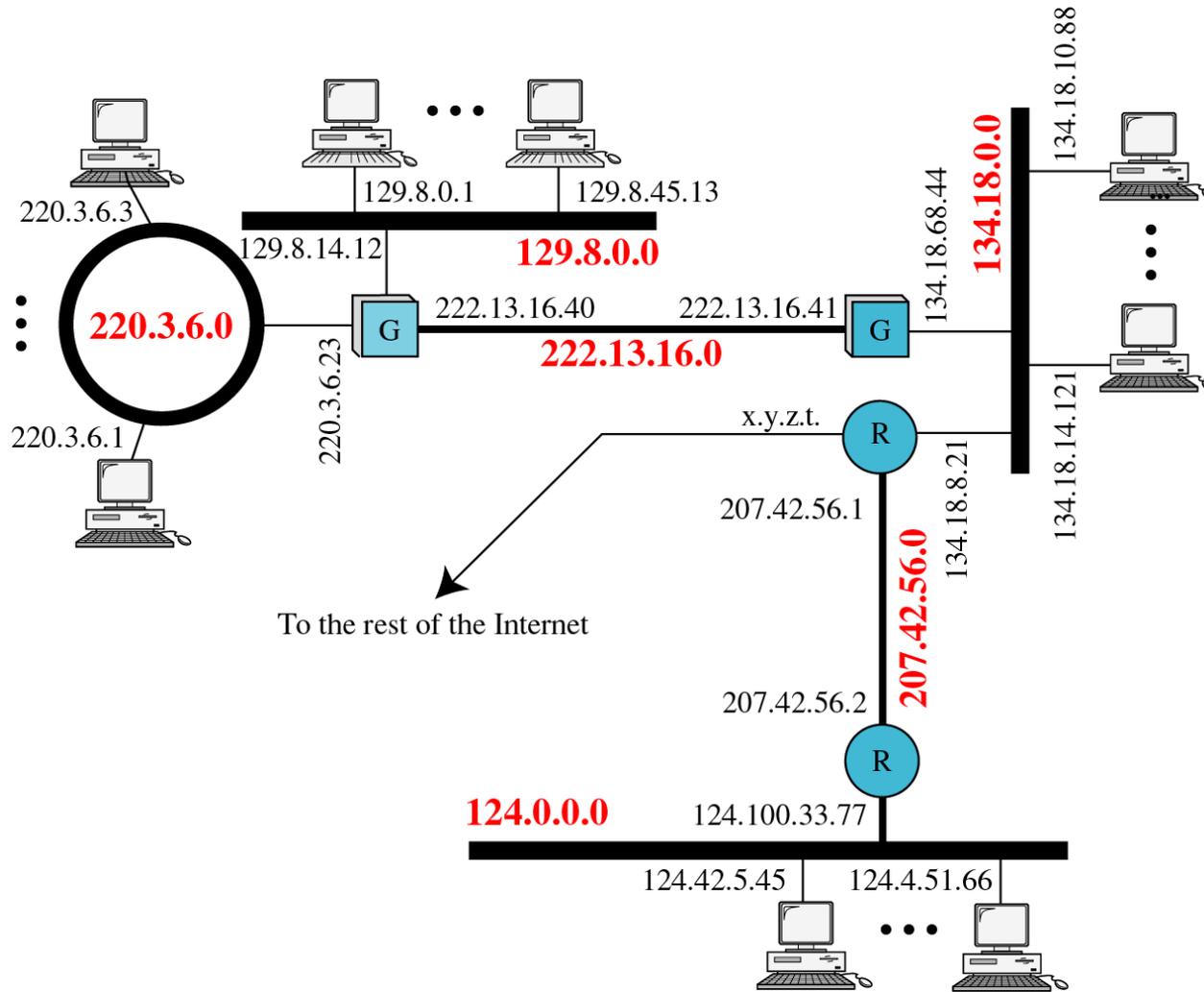
# Router Packet Delivery - again

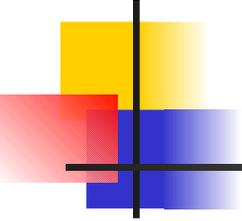
---

- Key operation in Route packet delivery:
  - Determine whether Destination IP address matches with a subnet in Route Table entry.
- How is this done?
- For one entry in route table, let DN = Destination subnet, SM = Subnet mask)
- If  $DN = [(Dest. IP address) AND (SM)]$   
then this Route Table entry matches Destination address.

Note: if there are no exact match, the router will match the IP to the longest possible prefix and forward the packet based on that.

# Routing Example: Let's write the routing tables in class

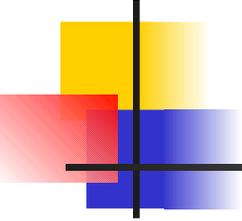




# What do routers do?

---

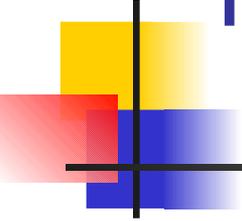
- Store and Forward packets based on destination IP address
- Interconnect multiple IP subnets
- Provide a gateway between different physical networks.
- **Provide a boundary for network security and management**
- Firewall services (packet filtering)



# IP Private Addresses

---

- Some IP Address blocks are set aside for private use.
- Private IP Networks :
  - Class A: 10.0.0.0/8
  - Class B: 172.16.0.0/16 to 172.31.0.0/16 (*172.16.0.0/12*)
  - Class C: 192.168.0.0/24 to 192.168.255.0/24 (*192.168.0.0/16*)
- Cannot be used on the public Internet backbone. Backbone routers do not have any route table entries for these networks. - *hehe, sometimes they leak*

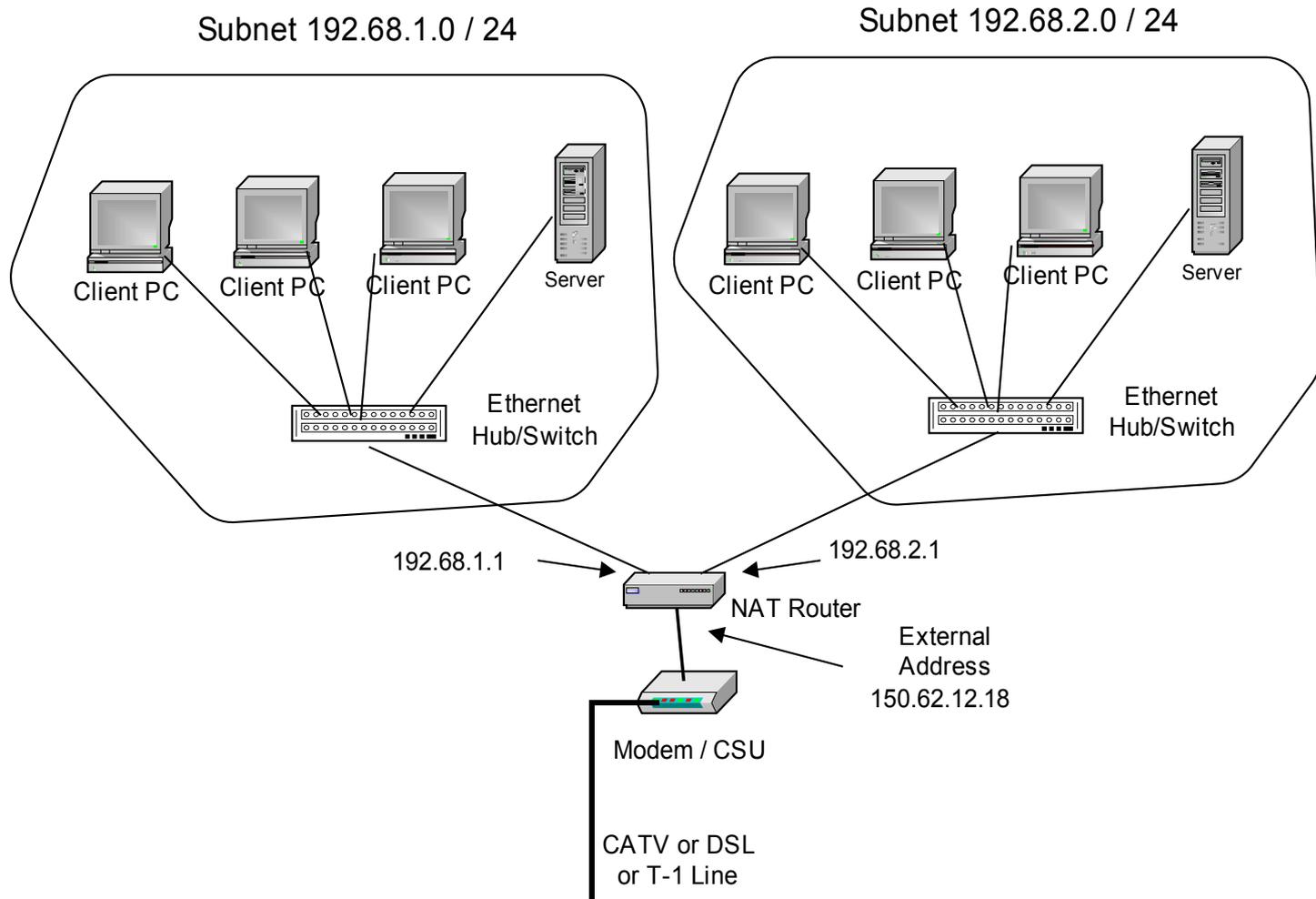


# Network Address Translations

---

- **Network Address Translation** allows a single public IP address to be shared by multiple internal hosts who use private IP addresses internally. - *NO! That requires "port address translation too" - NAT technically maps between one external address and one internal address*
- The access router or firewall swaps IP addresses in the headers of incoming and outgoing IP packets.

# SOHO NAT router



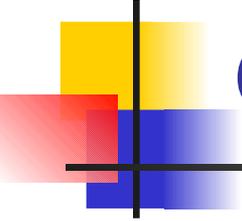
# NAT Table in Router

ports here refer to source port

<u>Internal</u> IP & Port	<u>External</u> IP & Port
192.168.1.100 80	150.62.12.18 80
192.168.1.100 1104	150.62.12.18 2200
192.168.2.105 80	150.62.12.18 2201
192.168.2.110 25	150.62.12.18 25

# Network (port) Address Translation

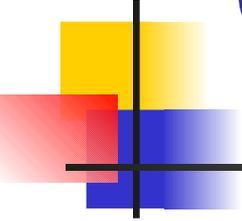
- For each outgoing packet: NAT router will look up Source IP, port in Internal part of NAT table
  - If found, swap in the External IP, port, send out to Internet.
  - If not found, choose a new, unused port number and create a new NAT table entry with
    - Internal IP, port = packet Source IP, port
    - External IP, port = Fixed external IP (i.e. 150.62.12.18), new port #.
- For each incoming packet: NAT router will look up Dest IP, port in External part of NAT table:
  - If found, swap in the Internal IP, port, send into local subnets.
  - If not found, log error (packet in w/o previous pkt out??) and send back Packet Undeliverable error message using ICMP (or not: depends on NAT/firewall).



# Some more IP subnetting examples and exercises

---

- First: subnetting is easy. We can even make it easier.

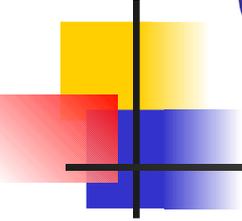


# Cheat sheet #3

---

- Subnet, host bits per byte based on subnet mask value

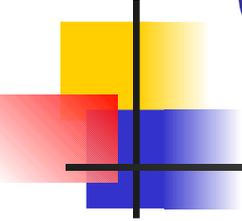
Mask Value (Decimal)	Mask Value (8-bit Binary)	Subnet Bits	Host bits
0	00000000	0	8
128	10000000	1	7
192	11000000	2	6
224	11100000	3	5



# Cheat sheet #3

- Subnet, host bits per byte based on subnet mask value

Mask Value (Decimal)	Mask Value (8-bit Binary)	Subnet Bits	Host bits
240	11110000	4	4
248	11111000	5	3
252	11111100	6	2
254	11111110	7	1
255	11111111	8	0

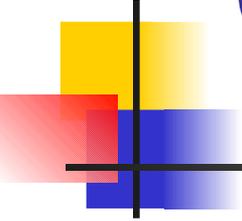


# Cheat sheet #4

---

- Value splits per byte based on mask value other than 0 or 255.

Mask Value	#Subnets	#Values per subnet	Ranges
128	2	128	0-127, 128-255
192	4	64	0-63, 64-127, 128-191, 192-255
224	8	32	0-31, 32-63, ... 192- 223, 224-255
240	16	16	0-15, 16-31, ... 224- 239, 240-255

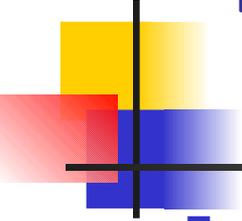


# Cheat sheet #4

---

- Value splits per byte based on mask value other than 0 or 255.

Mask Value	#Subnets	#Values per subnet	Ranges
248	32	8	0-7, 8-15, ... 240-247, 248-255
252	64	4	0-3, 4-7, 8-11, ..., 248-251, 252-255
254	128	2	0-1, 2-3, 4-5, ..., 252-253, 254-255

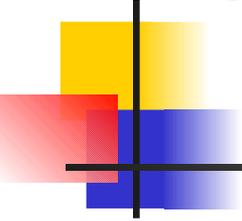


# Sample Problem

---

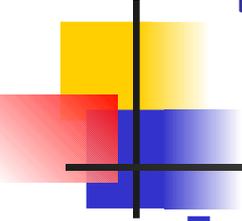
- Consider a PC with IP 205.18.45.123 and subnet mask 255.255.255.224.
  - What IP network is this address on? What is the subnet ID for its subnet?
  - List all IP addresses on same subnet as this one.
  - What is maximum number of hosts in this subnet?
  - How many different subnets exist in this IP network using this subnet mask?

# Solution



---

- Consider a PC with IP 205.18.45.123 and subnet mask 255.255.255.224.
  - What classfull IP network is this address on?
    - **Class C IP network – 205.18.45.0 / 24.**
    - **Classful (A, B, C) addressing is obsolete, almost always don't say that anymore**
  - What is the subnet ID for its subnet?
    - **Cheat sheet #3 – 3 more subnet bits gives /27.**
    - **Cheat sheet #4 – falls into IP range 96-127, so Subnet ID = 205.18.45.96 / 27.**
  - List all IP addresses on same subnet as this one.
    - **Cheat sheet #4 – falls into IP range 96-127, so range of addresses is 205.18.45.96 to 205.18.45.127.**
  - What is maximum number of hosts in this subnet?
    - **IP range has 32 addresses per subnet. But we subtract two because we can't assign 205.18.45.96 (subnet ID) or 205.18.45.127 (broadcast address) to any host machine. Maximum number of hosts = 30.**
  - How many different subnets exist in this IP network using this subnet mask?
    - **Cheat sheet #4 – 8 subnets exist within this Class C IP network using this subnet mask.**

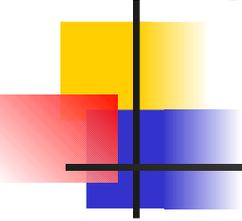


# Sample Problem

---

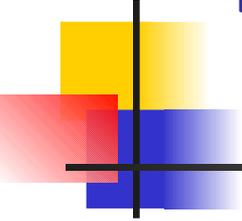
- Consider a PC with IP 133.29.150.68 and subnet mask 255.255.240.0.
  - What IP network is this address on? What is the subnet ID for its subnet?
  - List all IP addresses on same subnet as this one.
  - What is maximum number of hosts in this subnet?
  - How many different subnets exist in this IP network using this subnet mask?

# Solution



---

- Consider a PC with IP 133.29.150.68 and subnet mask 255.255.240.0.
  - What IP network is this address on?
    - **Class B IP network – 133.29.0.0 / 16.**
    - **Classful (A, B, C) addressing is obsolete, almost always don't say that anymore**
  - What is the subnet ID for its subnet?
    - **Cheat sheet #3 – 4 more subnet bits gives /20.**
    - **Cheat sheet #4 – falls into IP range 144 - 159, so Subnet ID = 133.29.144.0 / 20.**
  - List all IP addresses on same subnet as this one.
    - **Cheat sheet #4 – 3<sup>rd</sup> byte has IP range 144 - 159, 4<sup>th</sup> byte has range 0 – 255, so range of addresses is 133.29.144.0 to 133.29.159.255.**
  - What is maximum number of hosts in this subnet?
    - **IP range has 16 addresses per subnet in 3<sup>rd</sup> byte. But there are also 256 possible addresses in 4<sup>th</sup> byte. Total hosts = 16\*256 = 4096. But we subtract two because we can't assign 133.29.144.0 (subnet ID) or 133.29.159.255 (broadcast address) to any host machine. Maximum number of hosts = 4094.**
  - How many different subnets exist in this IP network using this subnet mask?
    - **Cheat sheet #4 – 16 subnets exist within this Class B IP network using this subnet mask.**



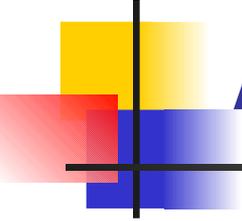
# Sample Problem

---

- Consider a PC with IP 39.62.100.112 and subnet mask 255.192.0.0. This PC wishes to transmit a data packet to destination IP 39.65.13.92.
  - *Does this packet need to go through a router or not?*

# Solution

- Consider a PC with IP 39.62.100.112 and subnet mask 255.192.0.0. This PC wishes to transmit a data packet to destination IP 39.65.13.92.
  - ***Does this packet need to go through a router or not?***
    - **What is the subnet ID for source IP address?**
      - Source IP address is on class A network 39.0.0.0/8
    - Classful (A, B, C) addressing is obsolete, almost always don't say that anymore - get the point yet?
      - Cheat sheet #3 – 2 more subnet bits gives /10.
      - Cheat sheet #4 – source address 2<sup>nd</sup> byte falls into IP range 0-63, so Subnet ID = 39.0.0.0 / 10.
    - **What is the subnet ID for destination IP address?**
      - Dest. IP address is on class A network 39.0.0.0/8
      - Cheat sheet #3 – 2 more subnet bits gives /10.
      - Cheat sheet #4 – source address 2<sup>nd</sup> byte falls into IP range 64-127, so Subnet ID = 39.64.0.0 / 10.
    - **These addresses are on *different subnets*.**
    - **So - YES – the packet must be sent through a router to be delivered.**



# A routing table problem

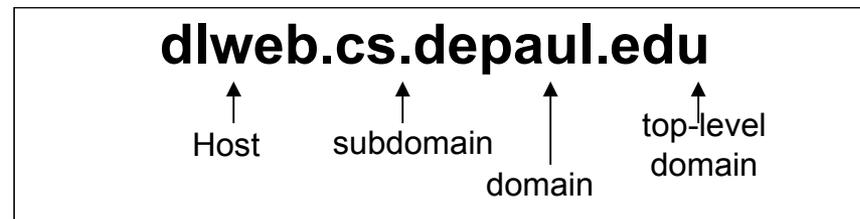
---

- See

<http://<removed>/IP-routing-table-example.xls>

# Domain Name System

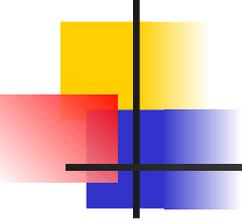
A system of **Domain Name System (DNS)** (someone went to the redundancy school of redundancy :-)) servers allows users to refer to any device by **DNS Name** (i.e. dlweb.cs.depaul.edu) rather than by **IP address** (i.e. 140.192.32.9)



# Domain Name System

This slide is pretty bad - don't use this

- ~~Top-Level~~ **Root DNS Servers** store address of each Domain Server for a particular **top-level domain** (i.e. ".edu")
- ~~Domain~~ **DNS Servers** store names and IP addresses for organization (i.e. depaul.edu)  
*name servers*
- Domain Names are ~~registered with the ICANN and associated with IP address prefixes.~~
  - ~~Example (depaul.edu <=> 140.192.x.x)~~

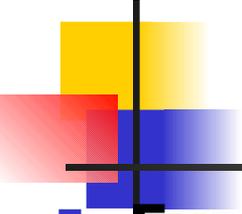


# DNS Top-Level Domains

---

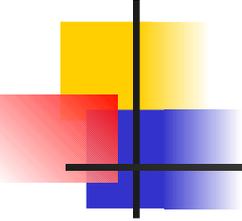
- Top-Level Domains (TLDs):
  - The original 7 “generic TLDs”
    - .edu, .com, .gov, .mil, .org, .net, .int.
  - The “.arpa” TLD for special lookups
  - New TLDs approved in 2002/3:
    - .info, .biz, .aero, .coop, .museum, .name, .pro
  - 2-letter top-level domains for each country - .fr for France, .de for Germany, etc.
- See <http://www.icann.org/tlds/>

# Domain Name System



---

- **Example: You type "www.ibm.com"**
  - Your PC sends DNS request to DePaul DNS server
  - DePaul DNS server sends request to ".com" top-level DNS server asking it to look up "www.ibm.com" server address.
  - Top-level DNS server sends back address *referral* for "ibm.com" Domain DNS Server at IBM Corp.
  - DePaul DNS server sends request packet to IBM DNS Server asking it to look up "www.ibm.com"
  - IBM DNS server looks up IP address and sends result back to DePaul DNS Server.
  - DePaul DNS server sends IP address back to your PC.

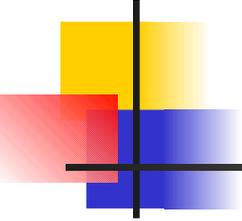


# Domain Name System

---

## ■ Caching

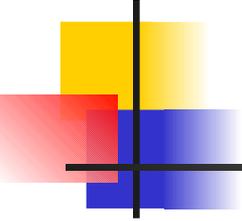
- All DNS servers keep Address Caches where they store all results that they have discovered lately. - *not all servers, authoritative-only servers don't keep caches*
- Thus, after DePaul DNS server has looked up “www.ibm.com” once, it stores address and responds locally when additional DePaul users ask about the same name.
- Cached address information is called “non-authoritative” and eventually times out.



# Dynamic DNS

---

- Traditional DNS databases are updated manually by a systems admin each time a new name is add or changed in a zone.
  - This is time-consuming and error-prone
- The Dynamic DNS protocol is designed to automate DNS zone database updates via a set of update messages exchanged between authenticated servers and clients.



# Useful TCP/IP Utility Programs

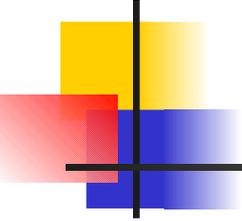
---

- ARP

- Can be used to display or modify ARP table at a host.
- Example: "arp -a" on *Windows* displays the ARP table

- PING

- Used to verify that a host is reachable and provide round-trip packet transfer time - *not so useful today, often filtered/disabled*
- Example: "ping [www.depaul.edu](http://www.depaul.edu)" - *exhibit A, ping won't work for this host*



# Useful TCP/IP Utility Programs

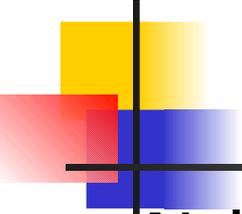
---

- Netstat

- Information about TCP/IP information stored on a host.
- Example: "netstat" provides listing of TCP connections currently in use, "netstat -r" displays the routing table

- Nbtstat

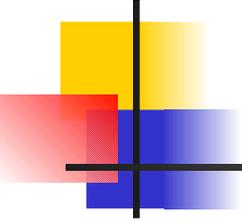
- NetBIOS information (station names and addrs)
- Example: "nbtstat -s" shows NetBIOS sessions



# Useful TCP/IP Utility Programs

---

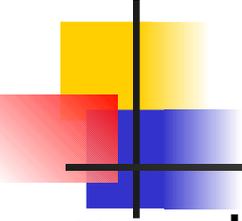
- **Nslookup** - *dig tool on unix is better*
  - Allows you to send a query to the local DNS server
  - Example: "nslookup www.microsoft.com" finds IP address for **www.microsoft.com**
  - Available only post-WinNT (i.e. WinXP, Win2000, Win2003 – not on Win95 or Win98).
- **Tracert** - *traceroute on unix*
  - Shows identity of each router on path to destination
  - Example: "tracert www.depaul.edu"



# Useful TCP/IP Utility Programs

---

- Ipconfig, winipcfg
  - Shows IP address and configuration information (winipcfg on Win98 and before)
- ifconfig
  - TCP/IP configuration utility on Linux/UNIX machines
  - Example: “ifconfig -a” shows status of all network interfaces.

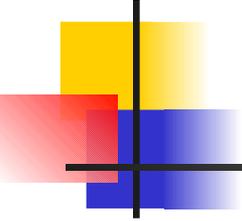


# Useful Utility Programs

---

- net

- The “net” command can be used to view and set network parameters for Windows file sharing.
- Type “net /?” at command prompt for list of net options.
- Examples:
  - net accounts – account information
  - net config workstation – configuration info
  - net statistics workstation – packet stats



# Useful Utility Programs

---

- netsh

- The “netsh” commands are a newer set of commands that can be used to view and set network parameters on Windows XP machines.
- Type “netsh /?” for list of netsh options.
- Examples:
  - netsh firewall show config – XP firewall settings
  - netsh interface show interface – network interface information
  - netsh routing ip nat show – NAT configuration