Lecture (06)
IP protocol

By:
Dr. Ahmed ElShafee

Agenda

- Introduction
- Network Layer Interaction with the Data Link Layer
- Network Layer (Layer 3) Addressing
- IP Routing and Routing Protocols

Introduction

Typical features of OSI layer 3

OSI layer 3 or network layer, or internet layer in case of TCP/IP protocol is a protocol that defines routing and addressing

Such as

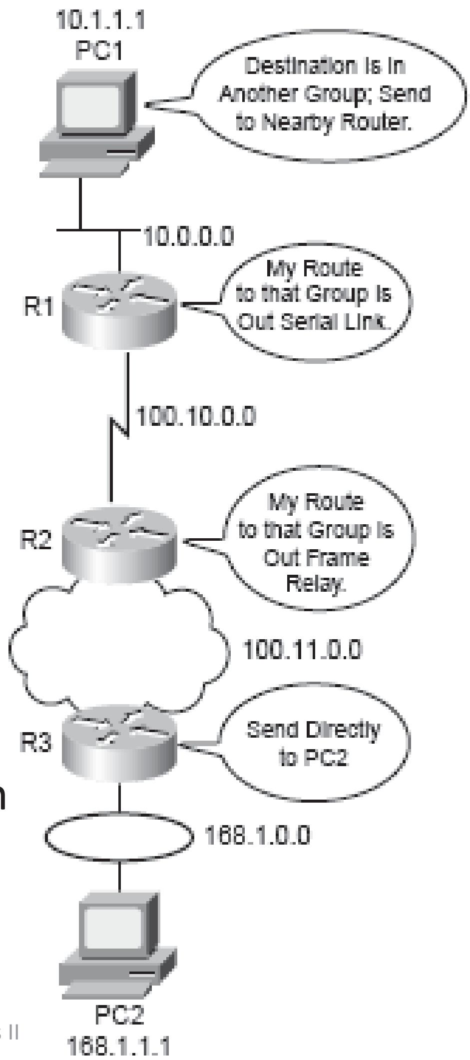
- Connectionless Network Services (CLNS),
- Internet Protocol (IP),
- Novell Internetwork Packet Exchange (IPX),
- or AppleTalk Dynamic Data Routing (DDR)

network layer protocols have many similarities.

Introduction (2)

Routing (Path Selection)

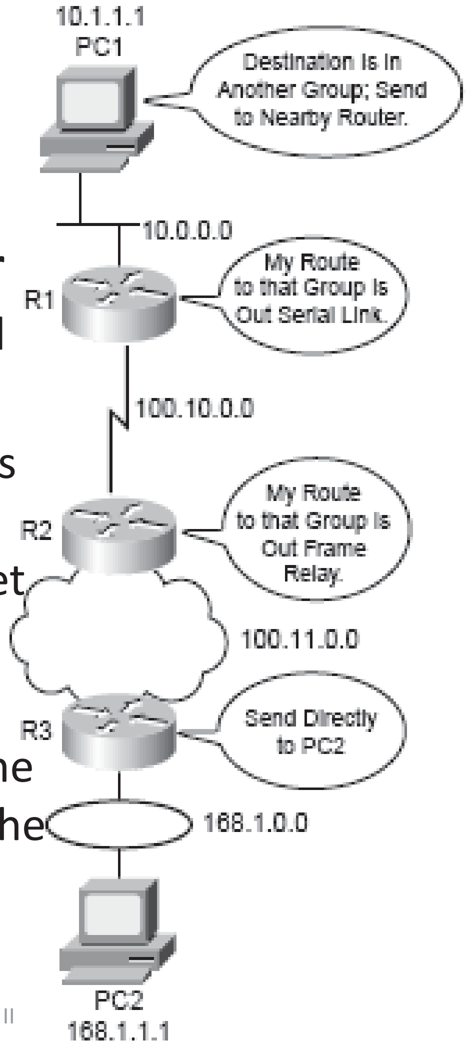
- Routing focuses on the end-to-end logic of forwarding data.
- Figure shows a simple example of how routing works.
- The logic seen in the figure is relatively simple.
- For PC1 to send data to PC2, it must send something to R1, when sends it to R2, then on to R3, and finally to PC2.



Introduction (3)

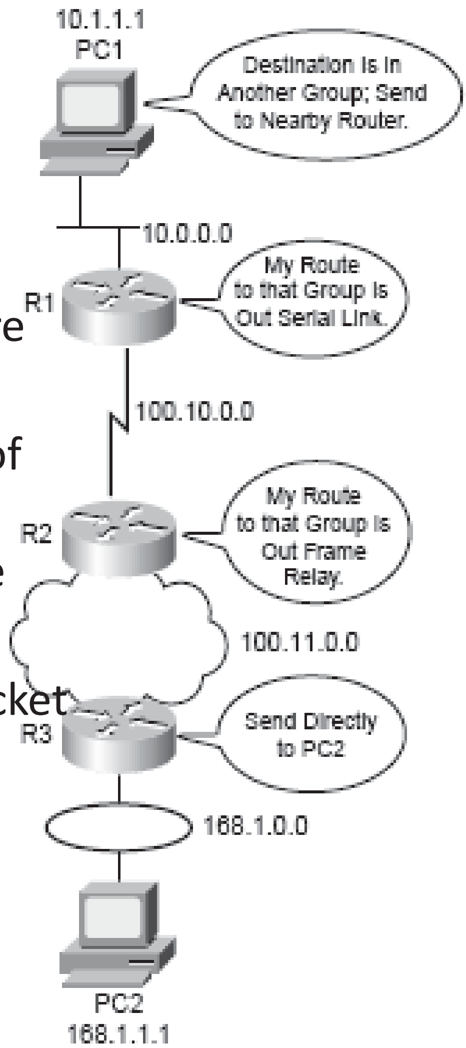
PC1's Logic: Sending Data to a Nearby Router

- In this example, PC1 has some data to send data to PC2.
- Because PC2 is not on the same Ethernet as PC1, PC1 needs to send the packet to a router that is attached to the same Ethernet as PC1.
- The sender sends a data-link frame across the medium to the nearby router; this frame includes the packet in the data portion of the frame.



Introduction (4)

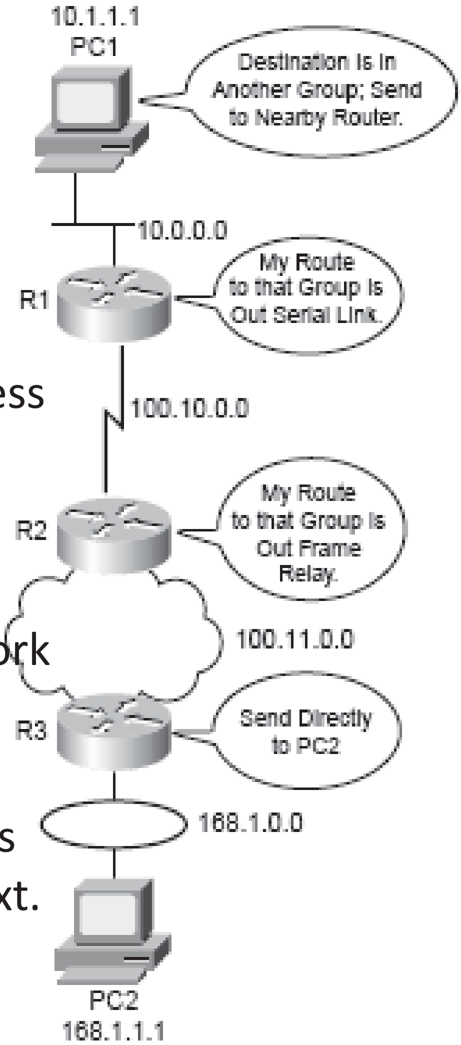
- That frame uses data link layer (Layer 2) addressing in the data-link header to ensure that the nearby router receives the frame.
- The main point here is that the originator of the data does not know much about the network—just how to get the data to some nearby router.
- PC1 needs to know only how to get the packet to R1.



Introduction (5)

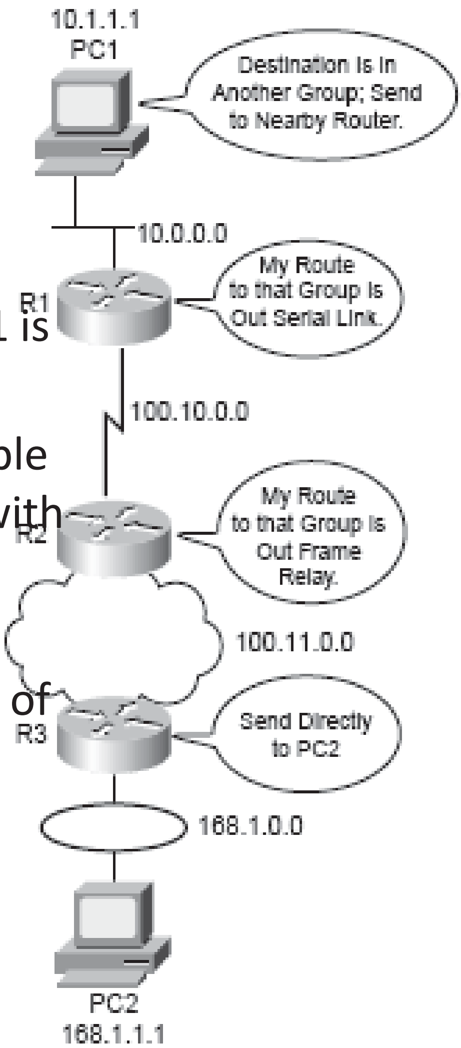
R1 and R2's Logic: Routing Data Across the Network

- R1 and R2 both use the same general process to route the packet
- The *routing table contains* one entry per group of addresses
- The router compares the destination network layer address in the packet to the entries in the routing table
- The matching entry in the routing table tells this router where to forward the packet next.



Introduction (6)

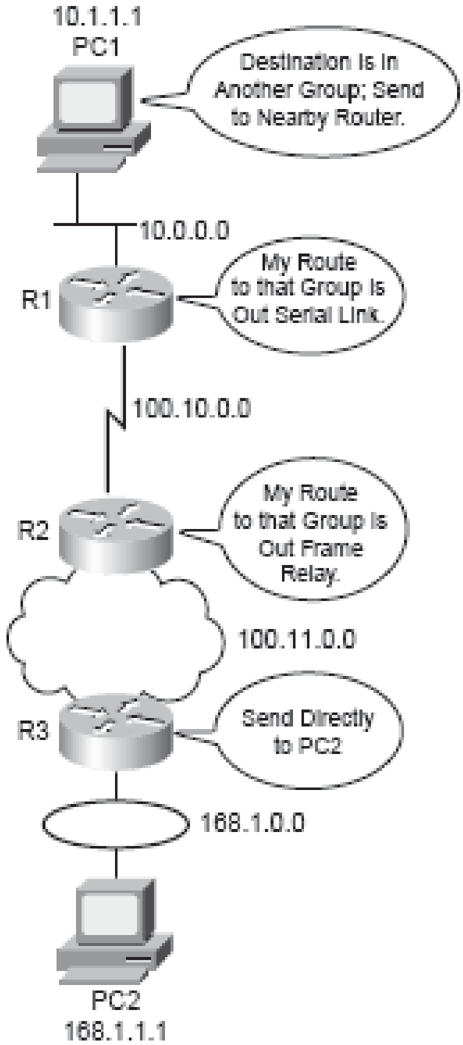
- Likewise, in Figure everyone in this the network whose IP address starts with 168.1 is on the Token Ring on which PC2 resides,
- So the routers can just have one routing table entry that means "all addresses that start with 168.1."
- Eventually, the packet is delivered to the router connected to the network or subnet of the destination host (R3),



Introduction (7)

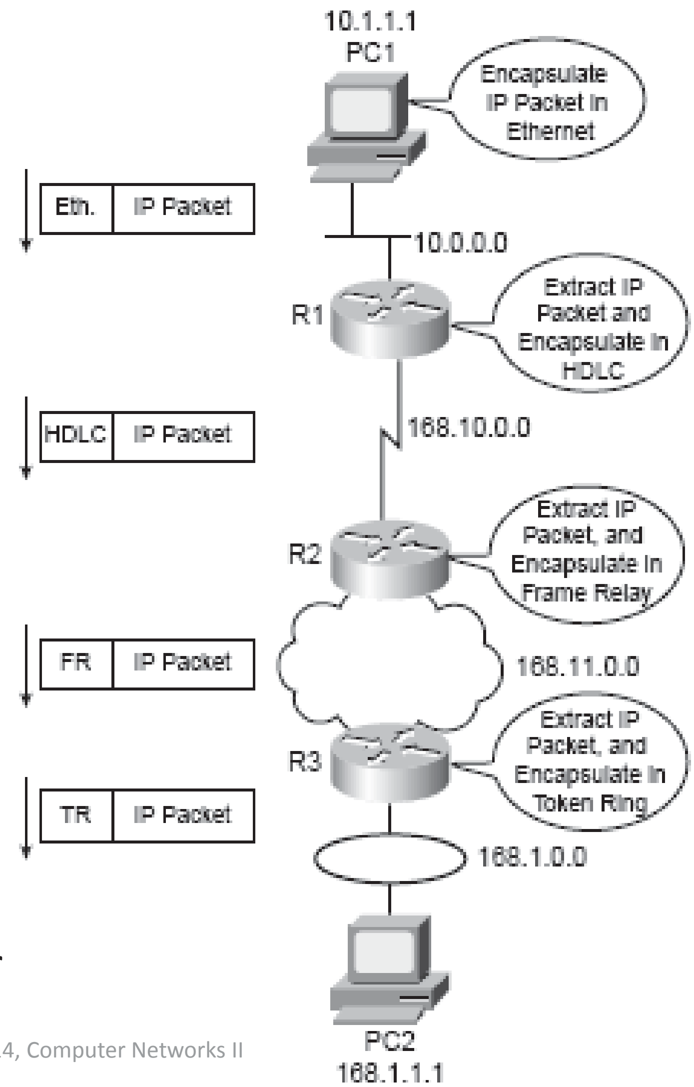
R3’s Logic: Delivering Data to the End Destination

- R3 needs to forward the packet directly t PC2, not to some other router.

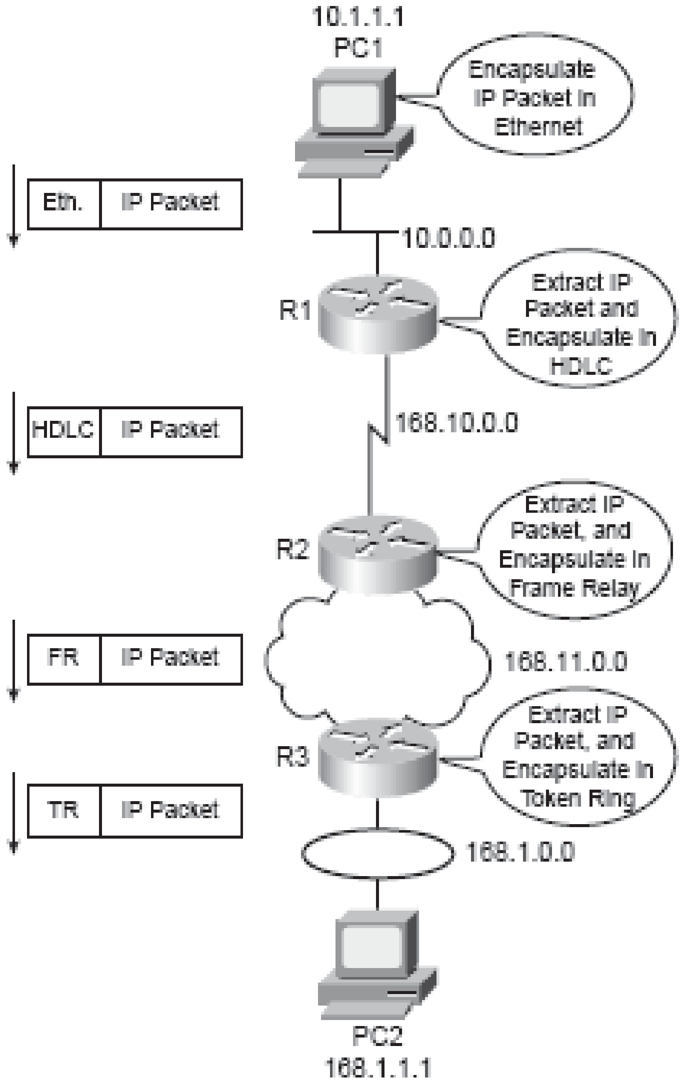


Network Layer Interaction with the Data Link Layer

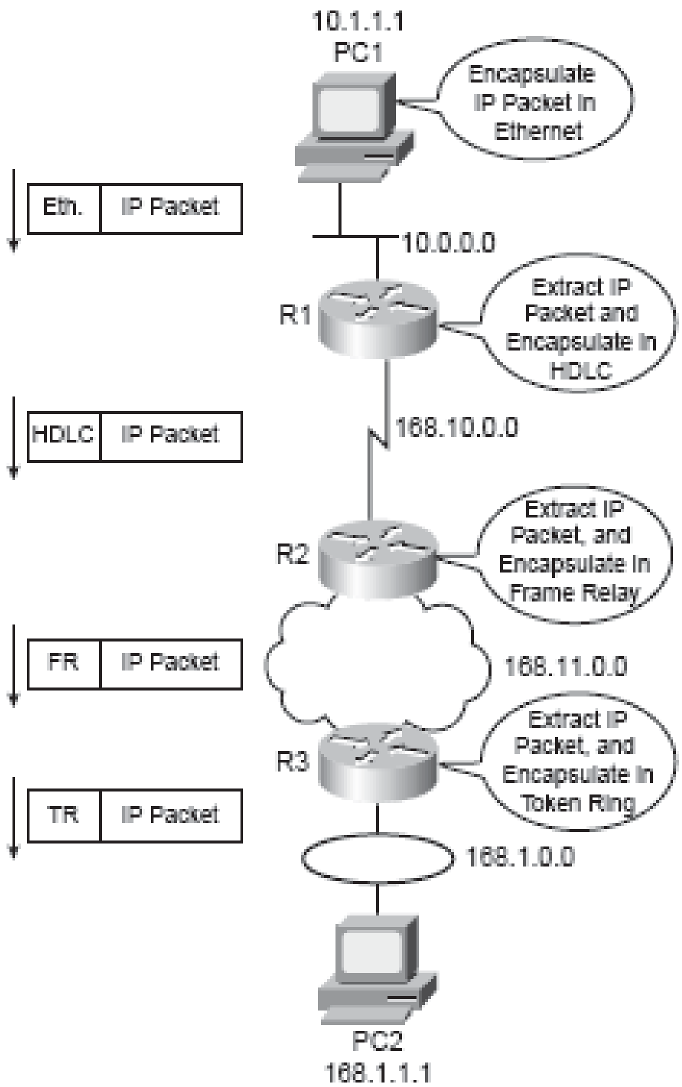
- four different types of data links were used to deliver the data
- When the network layer protocol is processing the packet, it decides to send the packet out the appropriate data link layer.
- the data link layer adds the appropriate header and trailer to the packet, creating a frame, before sending the frames over each physical network.



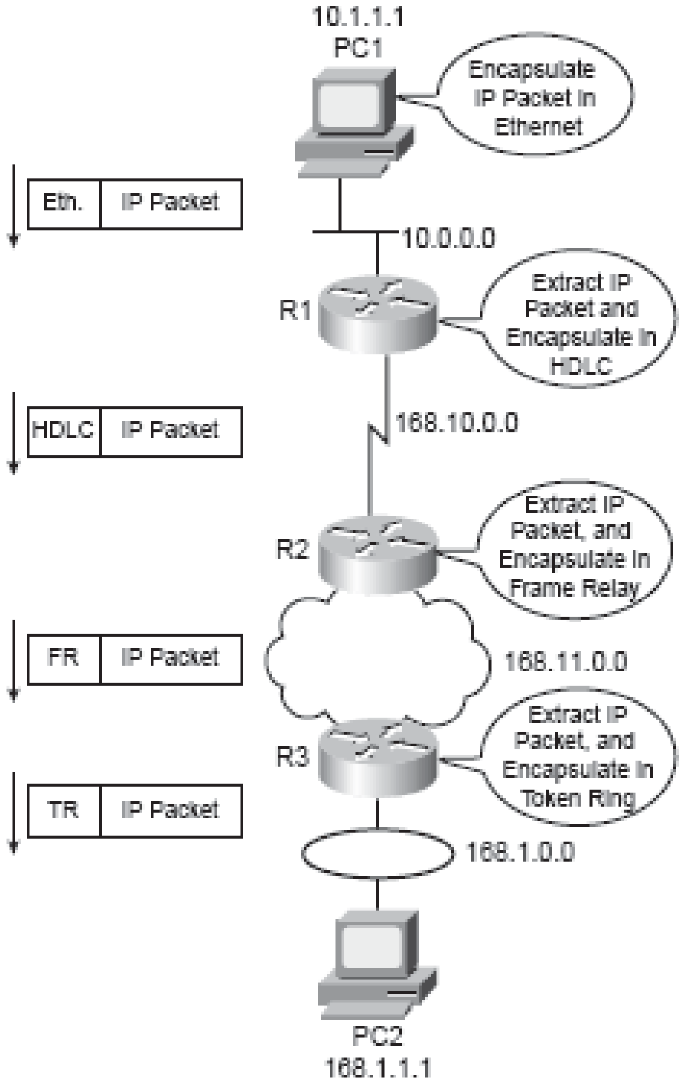
- The network layer processes deliver the packet end-to-end, using successive data-link headers and trailers just to get the packet to the next router or host in the path.
- the PCs and routers must have some way to decide what data-link addresses to use
- Address Resolution Protocol (ARP) is *used to dynamically learn the data-link address of an IP host connected to a LAN.*



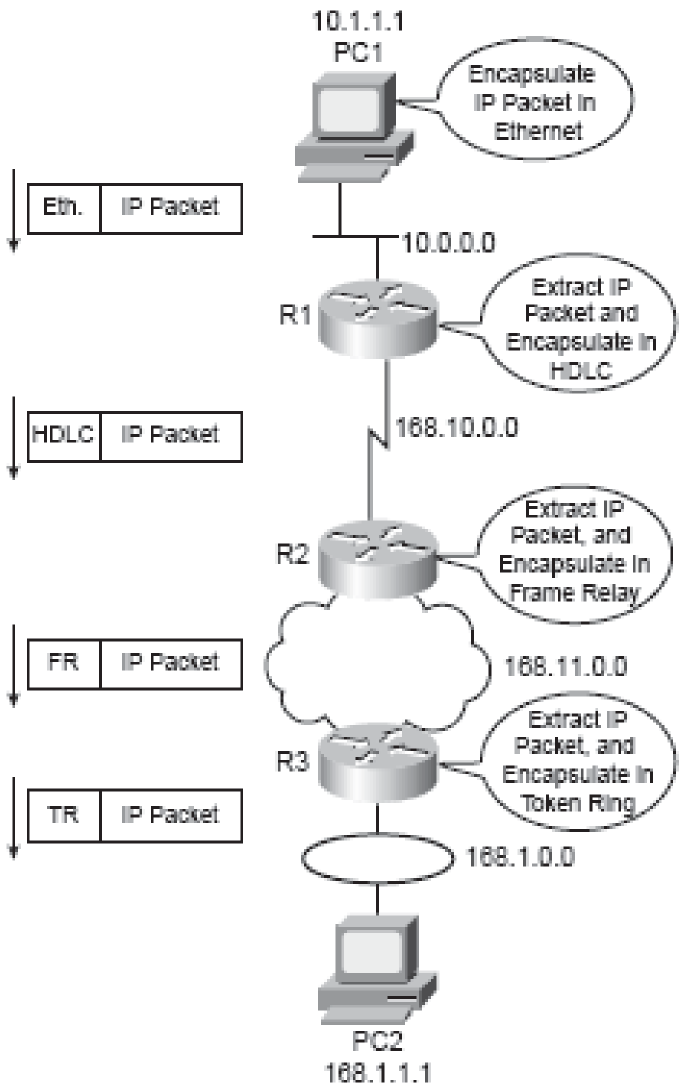
- One key feature of network layer addresses is that they were designed to allow logical grouping of addresses (routing process)
- In TCP/IP, this group is called a *network or a subnet.*
- *In IPX, it is called a network.*
- *In* AppleTalk, the grouping is called a *cable range.*
- Just like postal street addresses, network layer addresses are grouped based on physical location in a network



- In each of these network layer protocols, all devices on opposite sides of a router must be in a different Layer 3 group.
- The routing tables for each network layer protocol can have one entry for the group,
- Imagine an Ethernet with 100 TCP/IP hosts. A router needing to forward packets to any of those hosts needs only one entry in its IP routing table.
- This basic fact is one of the key reasons that routers can scale to allow tens and hundreds of thousands of devices

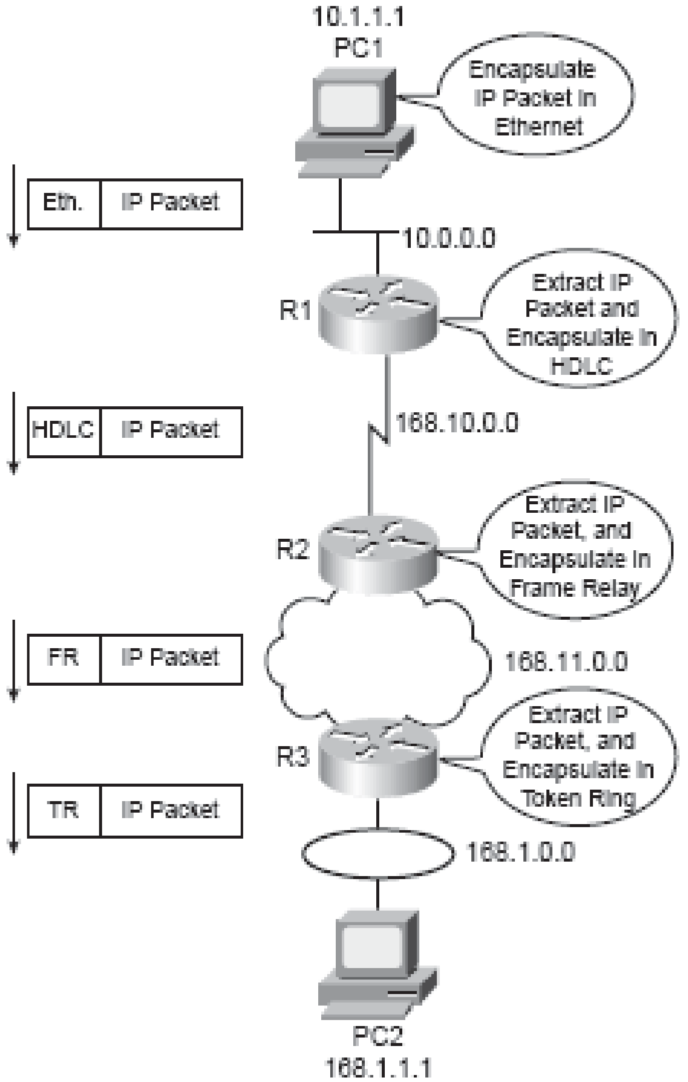


- The same logic of people in the same ZIP code live near to each other.
- If that local town wants to add streets, the rest of the post offices in the country already are prepared because they just forward letters based on the ZIP code, which they already know.
- The only postal employees who care about the new streets are the people in the local post office.
- Also, you can have duplicate local street addresses, as long as they are in different ZIP codes, and it all still works.



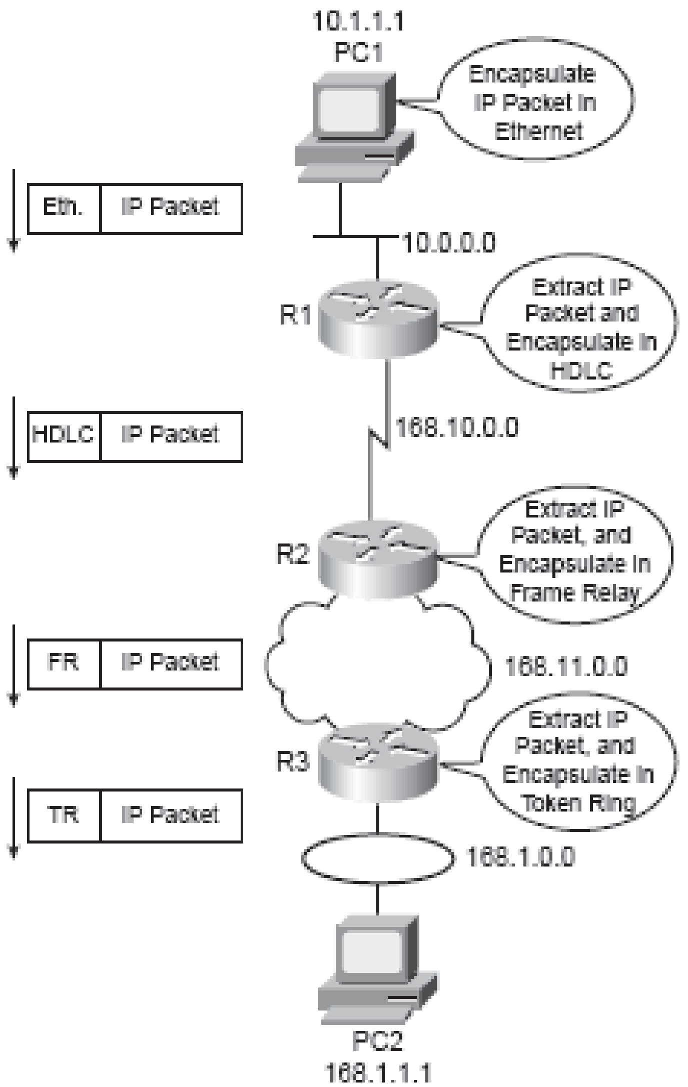
Example Layer 3 Address Structures

- Each Layer 3 address structure contains at least two parts. One (or more) part at the beginning of the address works like the ZIP code and essentially identifies the grouping.
- All addresses with the same value in the first part are in the same group—for example, the same IP subnet or IPX network or AppleTalk cable range.
- The last part of the address acts as a local address, uniquely identifying that device in that particular group

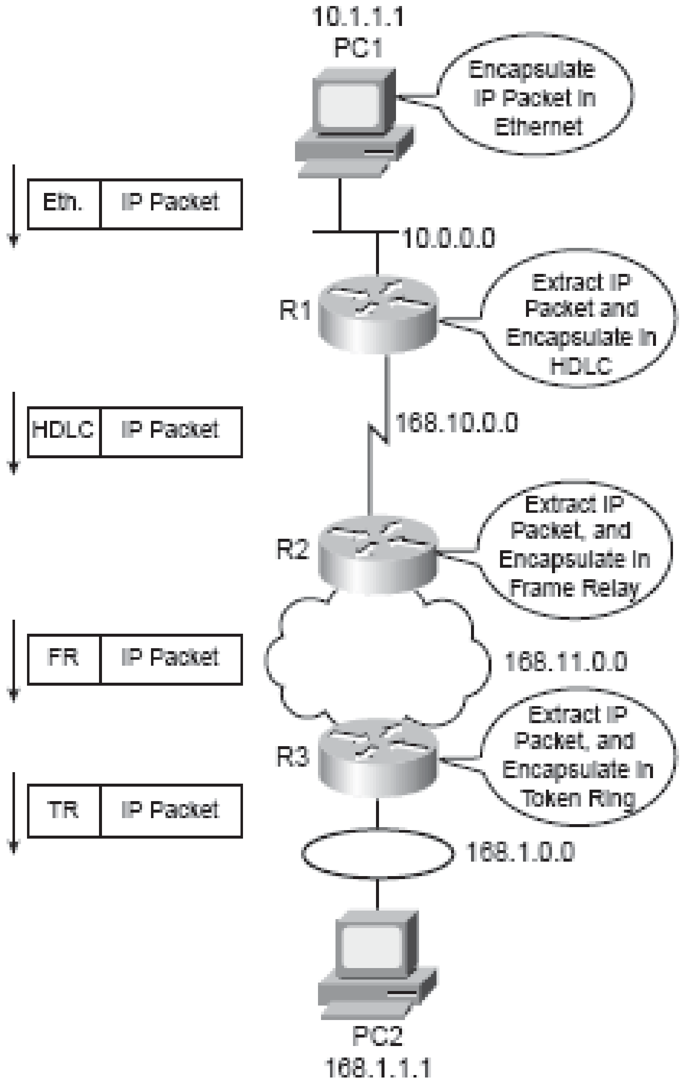


Routing Protocols

- In most cases, routing table entries are built dynamically by use of a routing protocol.
- Routing protocols learn about all the locations of the network layer "groups" in a network.
- Routing protocols define message formats and procedures to build the routing table.
- The end goal of each routing protocol is to fill the routing table with all known destination groups and with the best route to reach each group.



- routers uses the Routing Information Protocol (RIP) to learn the routes
- IP protocol called routed protocol, which is responsible of packet forwarded, or routed, through a network.



IP Addressing Definitions

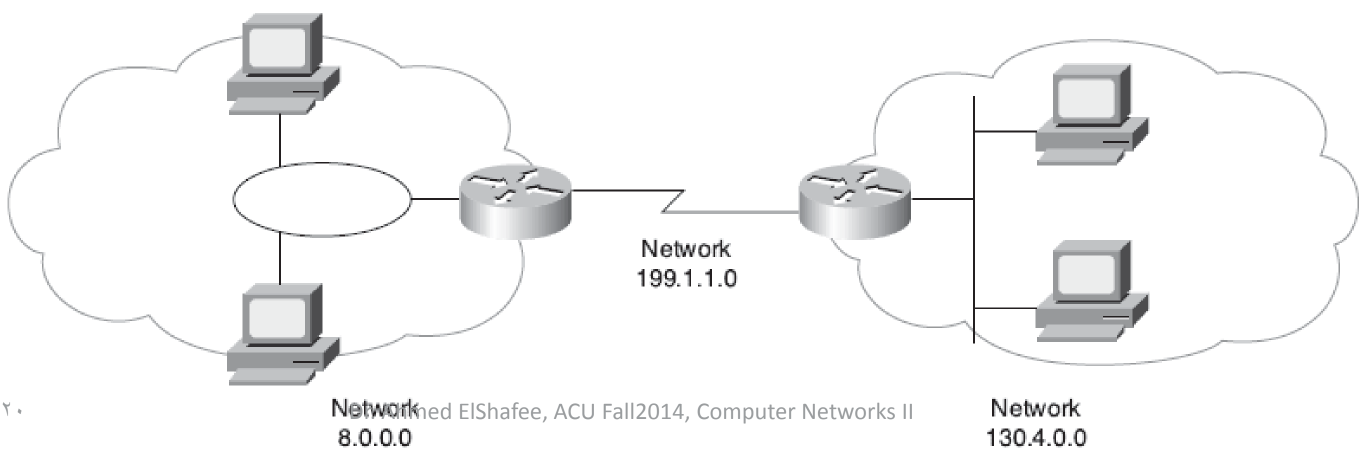
- If a device wants to communicate using TCP/IP, it needs an IP address.
- Any device that can send and receive IP packets is called an *IP host.*
- IP addresses consist of a 32-bit number, usually written in *dotted-decimal notation*
- For instance, 168.1.1.1 is an IP address written in dotted-decimal form
- Each of the decimal numbers in an IP address is called an *octet* which is a *byte.*
- The range of decimal numbers in each octet is between 0 and 255, inclusive.

- computer’s network card has an IP address.
- If you put two Ethernet cards in a PC to forward IP packets through both cards, they both would need unique IP addresses.
- Similarly, routers, which typically have many network interfaces that forward IP packets, have an IP address for each interface.

How IP Addresses Are Grouped Together

- For example, all IP addresses that begin with 8 are on the Token Ring on the left.
- Likewise, all IP addresses that begin with 130.4 are on the right.
- Along the same lines, 199.1.1 is the prefix on the serial link.
- By following this convention, the routers build a routing table with three entries, one for each prefix, or network number.

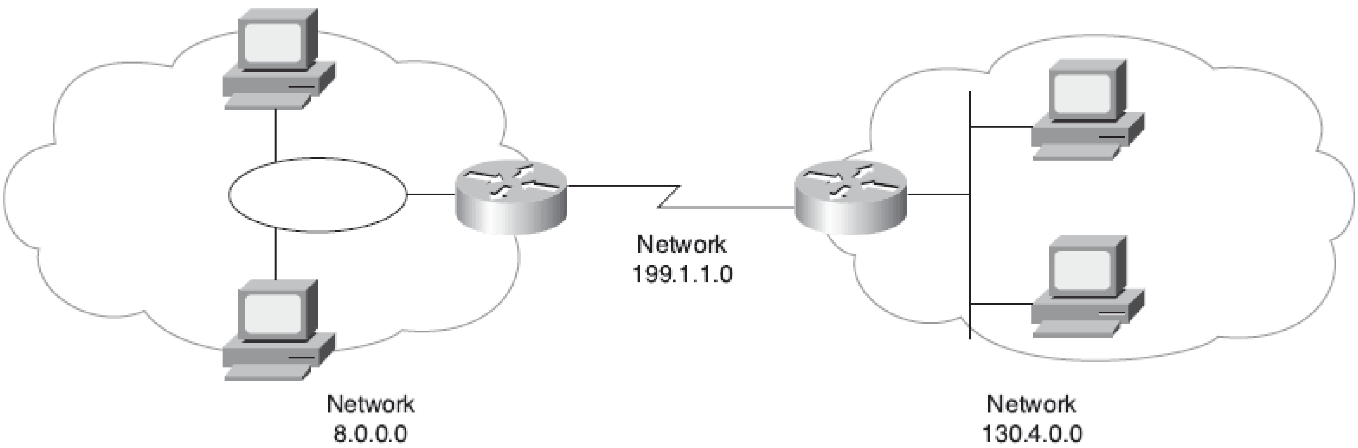
Sample Network Using Class A, B, and C Network Numbers



So, the general ideas about how IP address groupings can be summarized are as follows:

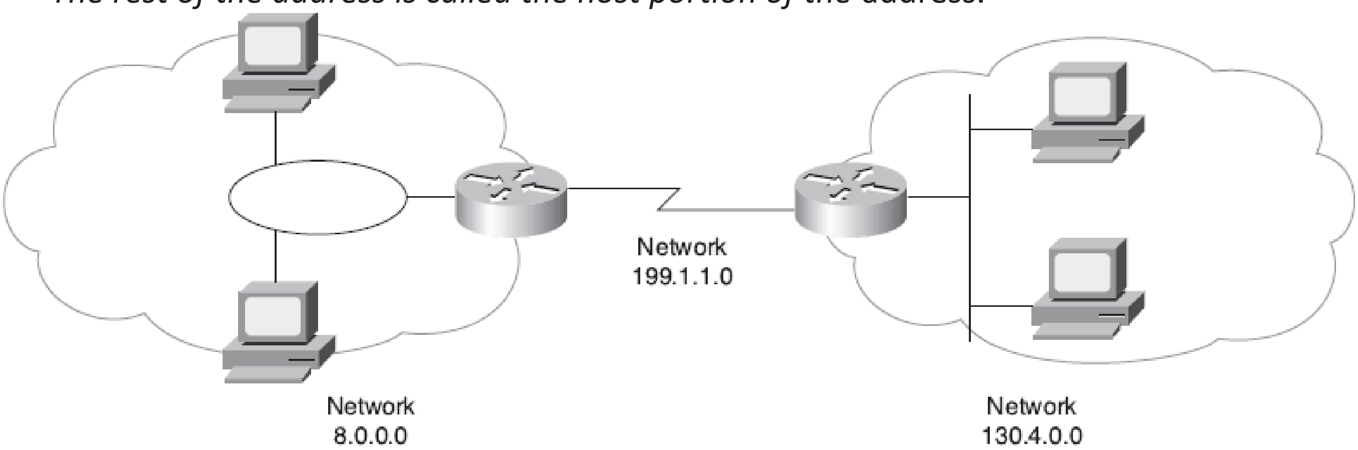
- All IP addresses in the same group must not be separated by a router.
- IP addresses separated by a router must be in different groups.

Sample Network Using Class A, B, and C Network Numbers



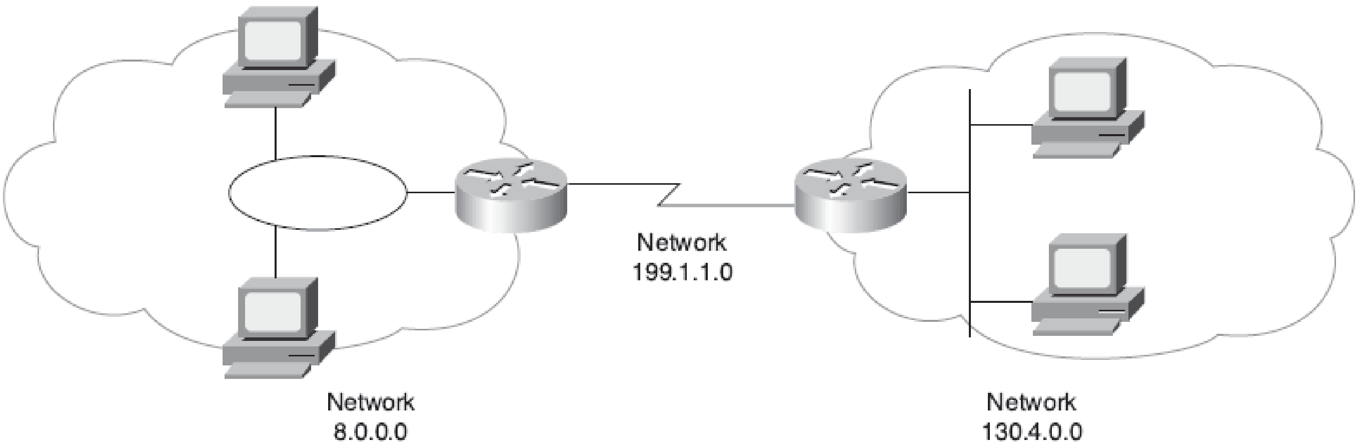
Classes of Networks

- IP defines three different network classes, called A, B, and C, from which individual hosts are assigned IP addresses.
- TCP/IP defines Class D (multicast) addresses and Class E (experimental) addresses as well.
- By definition, all addresses in the same Class A, B, or C network have the same numeric value *network portion of the addresses.*
- *The rest of the address is called the host portion of the* address.

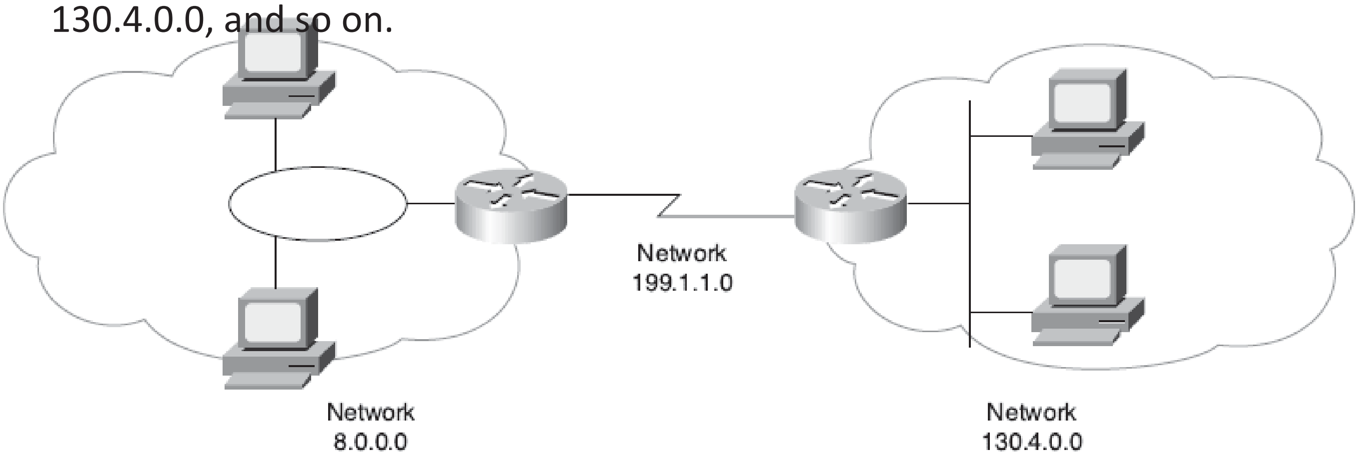


- Class A networks have a 1-byte-long network part. That leaves 3 bytes for the rest of the address, called the host part.
- Class B networks have a 2-byte-long network part, leaving 2 bytes for the host portion of the address.
- Class C networks have a 3-byte-long network part, leaving only 1 byte for the host part.

Sample Network Using Class A, B, and C Network Numbers



- network 8.0.0.0 next to the Token Ring. Network 8.0.0.0 is a Class A network, which means that only 1 byte is used for the network part of the address.
- Class B network 130.4.0.0 is listed next to the Ethernet; because it is Class B, 2 bytes define the network part, and all addresses begin with those same two bytes.
- So, Class A network “8” is written 8.0.0.0, Class B network 130.4 is written 130.4.0.0, and so on.



Sizes of Network and Host Parts of IP Addresses with No Subnetting

Any Network of This Class	Number of Network Bytes (Bits)	Number of Host Bytes (Bits)	Number of Addresses per Network*
A	1 (8)	3 (24)	$2^{24} - 2$
B	2 (16)	2 (16)	$2^{16} - 2$
C	3 (24)	1 (8)	$2^{8} - 2$

- Two numbers inside each Class A, B, or C network are reserved, one of the two reserved values is the network number itself

Example Network Numbers, Decimal and Binary

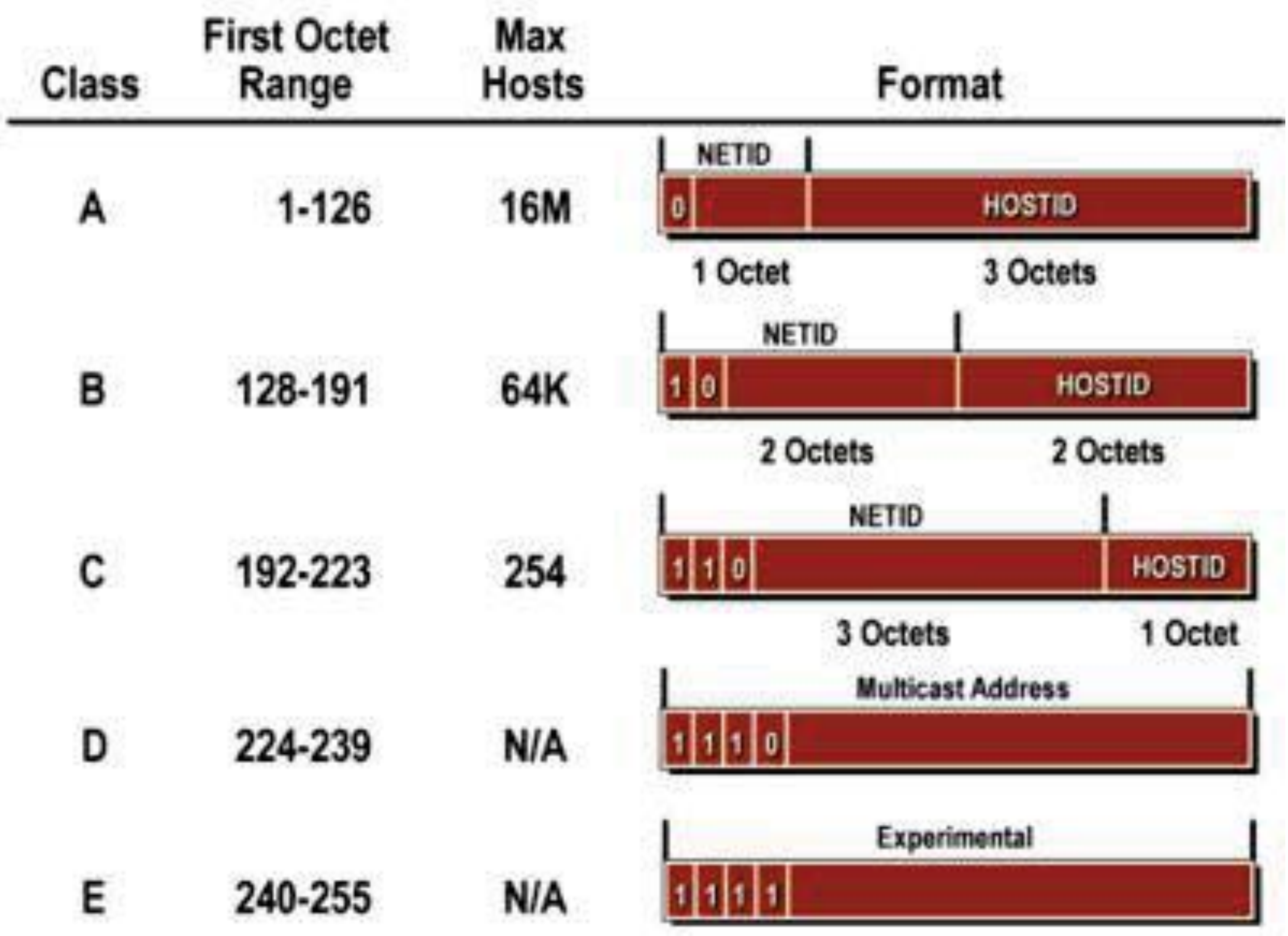
Network Number	Binary Representation, with Host Part Bold
8.0.0.0	00001000 00000000 00000000 00000000
130.4.0.0	10000010 00000100 00000000 00000000
199.1.1.0	11000111 00000001 00000001 00000000

- The other reserved value is the one with all binary 1s in the host part of the address—this number is called the *network broadcast or directed broadcast* address.
- network number is the lowest numerical value inside that network and the broadcast address is the largest, all the numbers between the network number and the broadcast address are the valid,

The Actual Class A, B, and C Network Numbers

*List of All Possible Valid Network Numbers**

Class	First Octet Range	Valid Network Numbers	Total Number of This Class of Network	Number of Hosts per Network
A	1 to 126	1.0.0.0 to 126.0.0.0	$2^7 - 2$	$2^{24} - 2$
B	128 to 191	128.1.0.0 to 191.254.0.0	$2^{14} - 2$	$2^{16} - 2$
C	192 to 223	192.0.1.0 to 223.255.254.0	$2^{21} - 2$	$2^8 - 2$

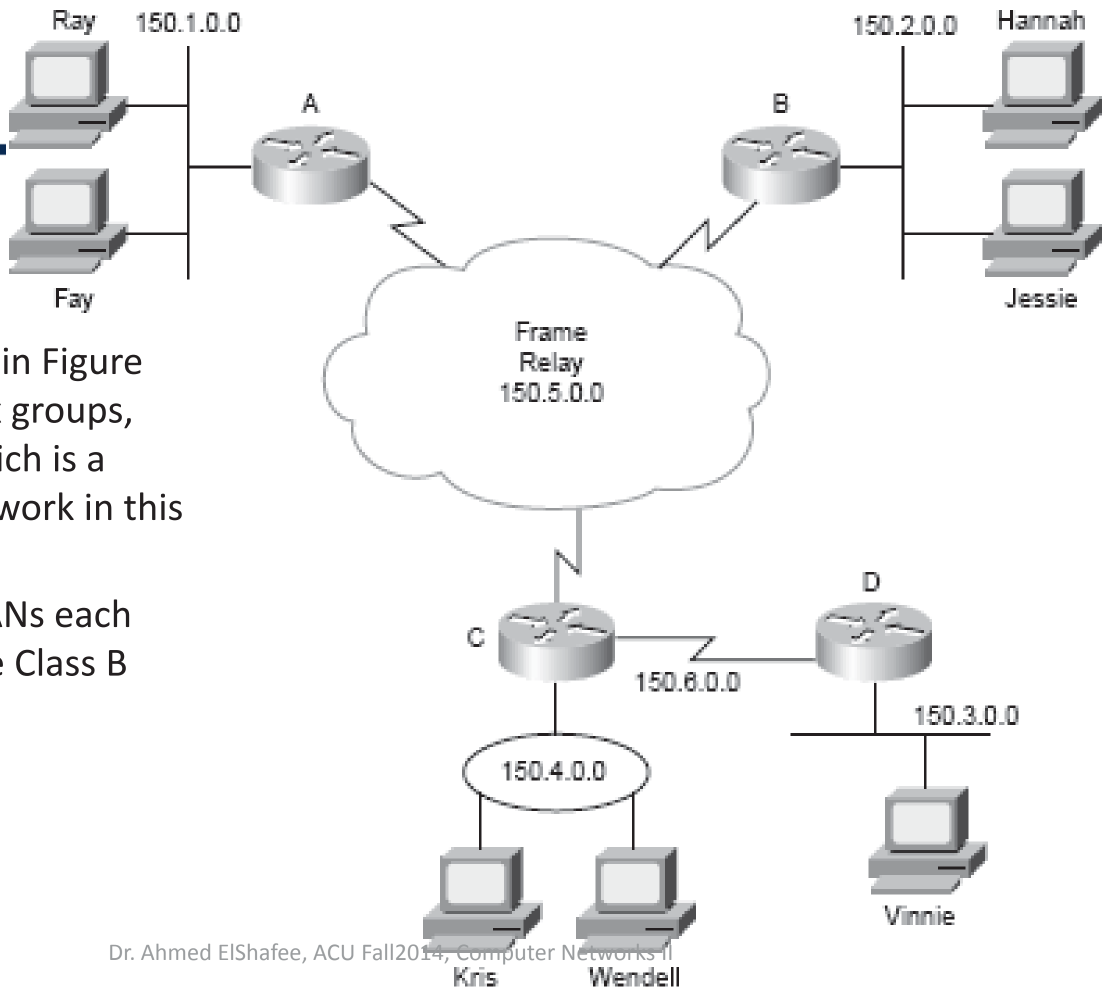


- The Valid Network Numbers column shows actual network numbers.
- There are several reserved cases.
- For example,
  - networks 0.0.0.0 (originally defined for use as a broadcast address) and
  - 127.0.0.0 (still available for use as the loopback address) are reserved networks
  - 128.0.0.0,
  - 191.255.0.0,
  - 192.0.0.0, and
  - 223.255.255.0 also are reserved.

IP Subnetting

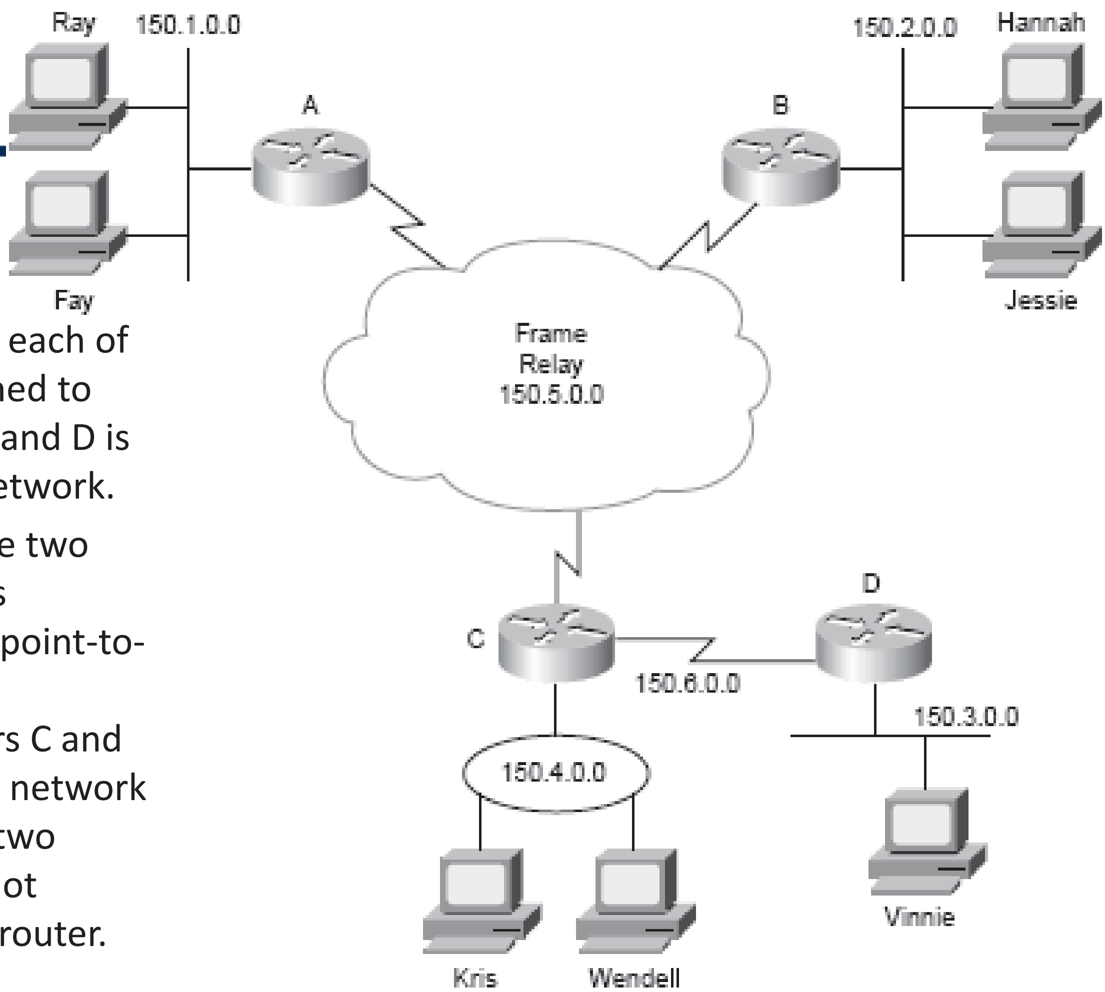
- IP subnetting creates vastly larger numbers of smaller groups of IP addresses, compared with simply using Class A, B, and C conventions.
- The Class A, B, and C rules still exist—but now, a single Class A, B, or C network can be subdivided into many smaller groups.
- By doing so, a single Class A, B, or C network can be subdivided into many non-overlapping subnets.

Backdrop for Discussing Numbers of Different Networks/Subnetworks

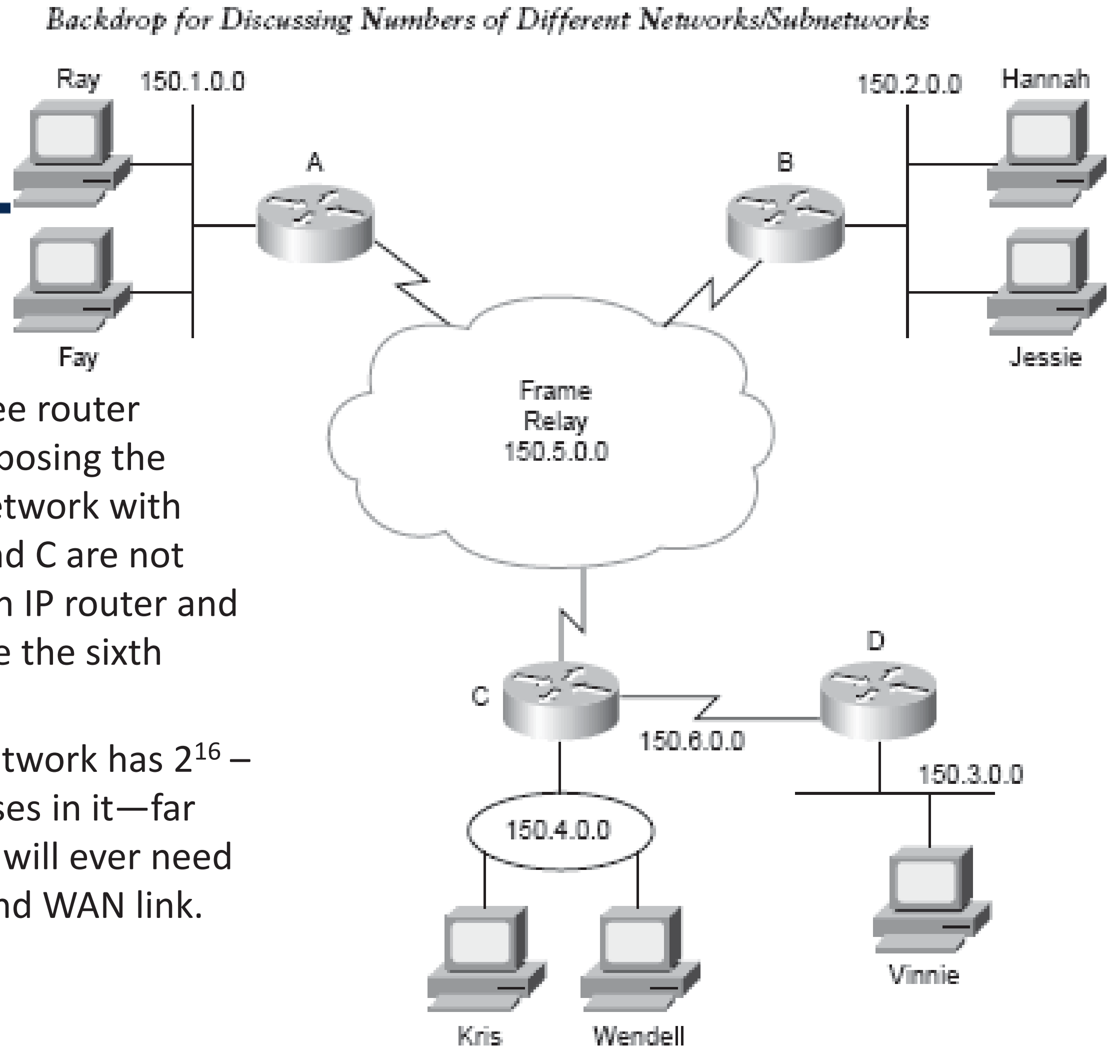


- The design in Figure requires six groups, each of which is a Class B network in this example.
- The four LANs each use a single Class B network.

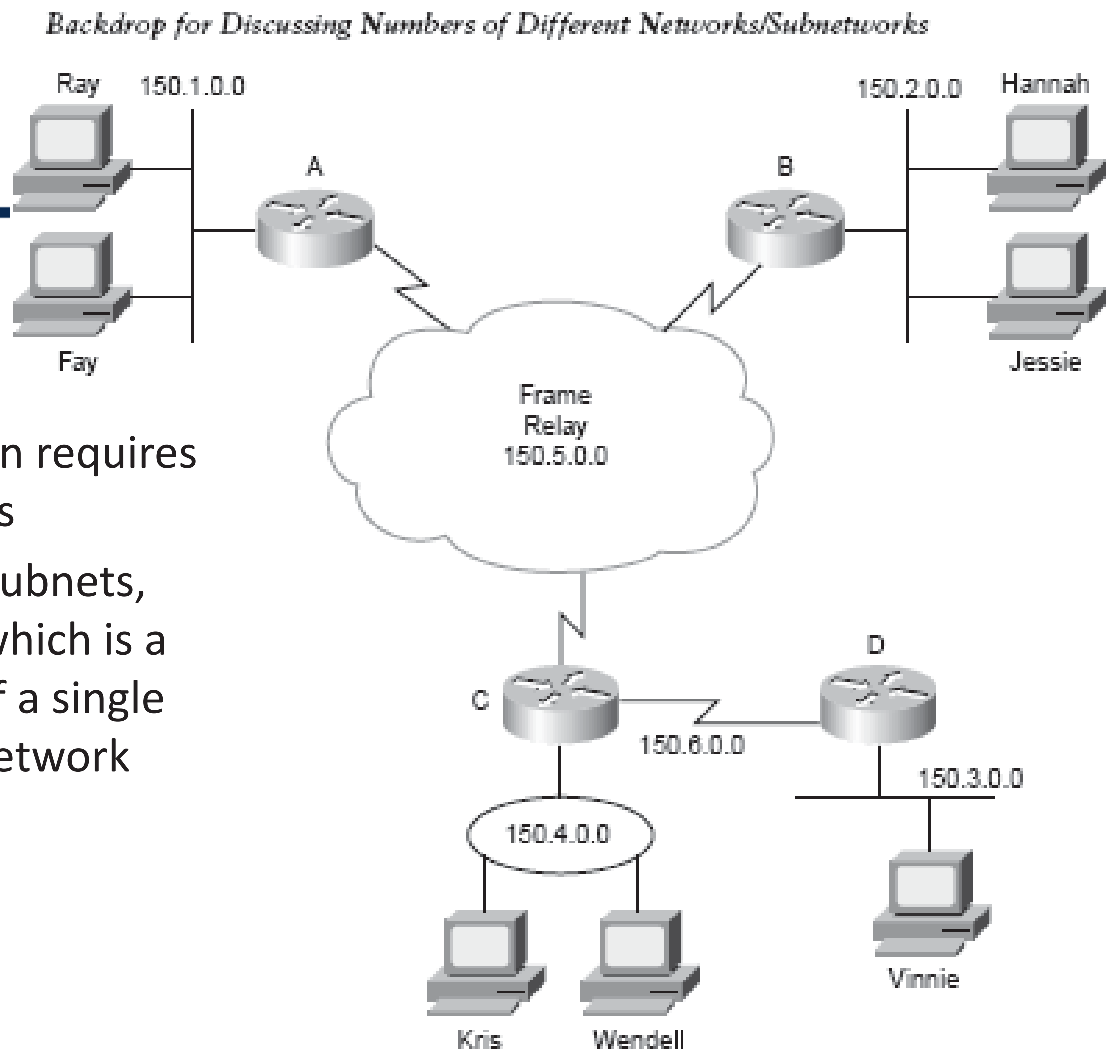
Backdrop for Discussing Numbers of Different Networks/Subnetworks



- In other words, each of the LANs attached to routers A, B, C, and D is in a separate network.
- Additionally, the two serial interfaces composing the point-to-point serial link between routers C and D use the same network because these two interfaces are not separated by a router.



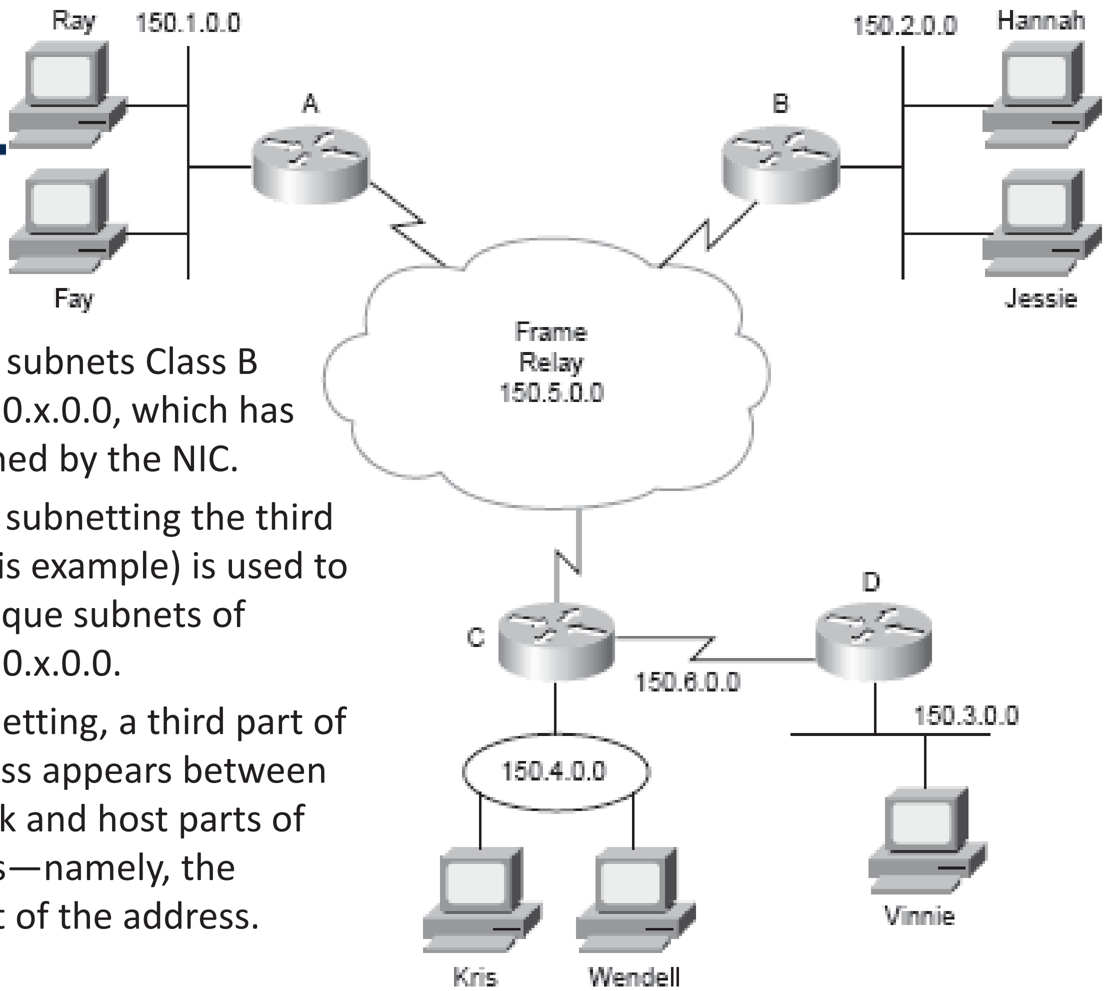
- Finally, the three router interfaces composing the Frame Relay network with routers A, B, and C are not separated by an IP router and would compose the sixth network.
- Each Class B network has $2^{16} - 2$ hosts addresses in it—far more than you will ever need for each LAN and WAN link.



- the design requires six groups
- uses six subnets, each of which is a subnet of a single Class B network

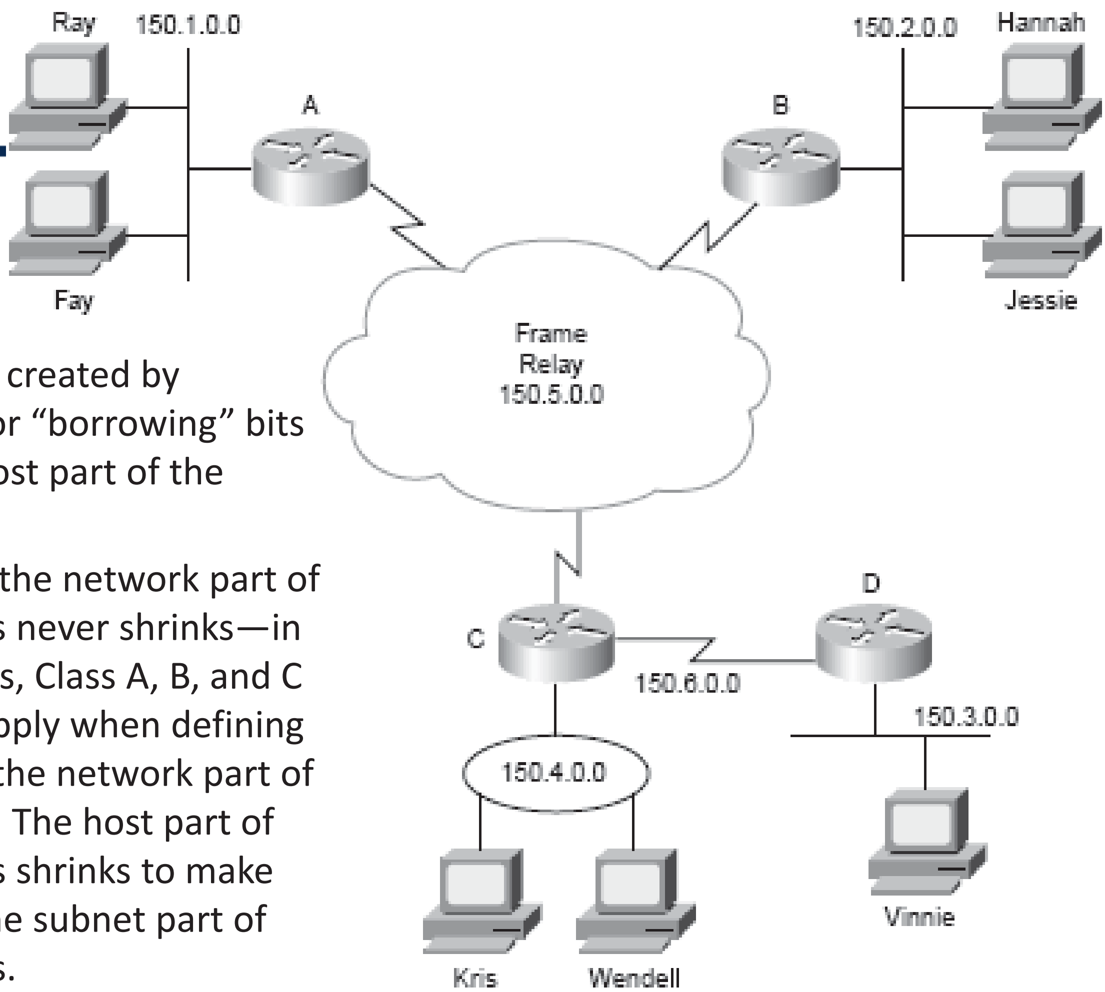
Netwo (24)

Backdrop for Discussing Numbers of Different Networks/Subnetworks



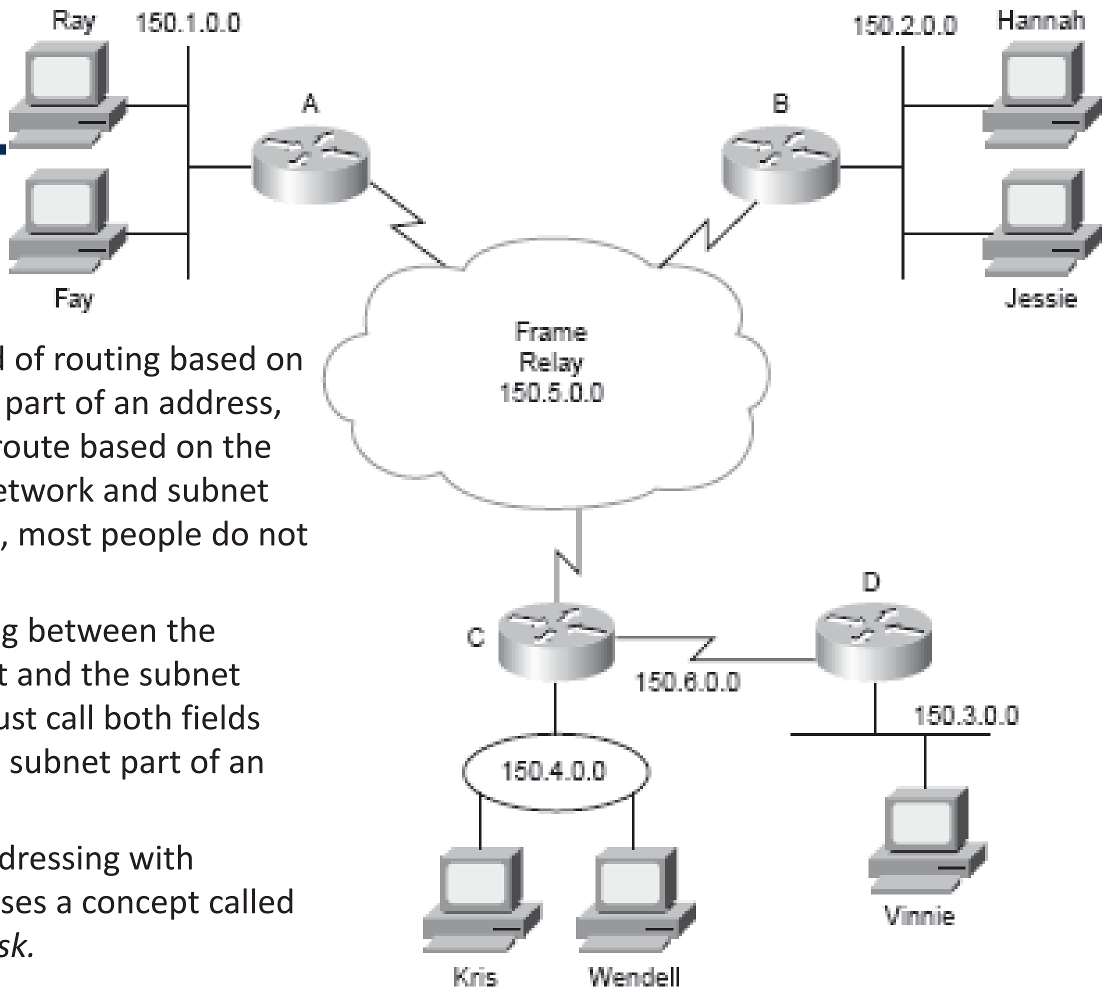
- This design subnets Class B network 150.x.0.0, which has been assigned by the NIC.
- To perform subnetting the third octet (in this example) is used to identify unique subnets of network 150.x.0.0.
- When subnetting, a third part of an IP address appears between the network and host parts of the address—namely, the subnet part of the address.

Backdrop for Discussing Numbers of Different Networks/Subnetworks



- This field is created by "stealing" or "borrowing" bits from the host part of the address.
- The size of the network part of the address never shrinks—in other words, Class A, B, and C rules still apply when defining the size of the network part of an address. The host part of the address shrinks to make room for the subnet part of the address.

Backdrop for Discussing Numbers of Different Networks/Subnetworks



- Now, instead of routing based on the network part of an address, routers can route based on the combined network and subnet parts. In fact, most people do not even bother
- distinguishing between the network part and the subnet part—they just call both fields together the subnet part of an address.
- Finally, IP addressing with subnetting uses a concept called a *subnet mask.*

Network Layer Utilities

Network Layer Utilities

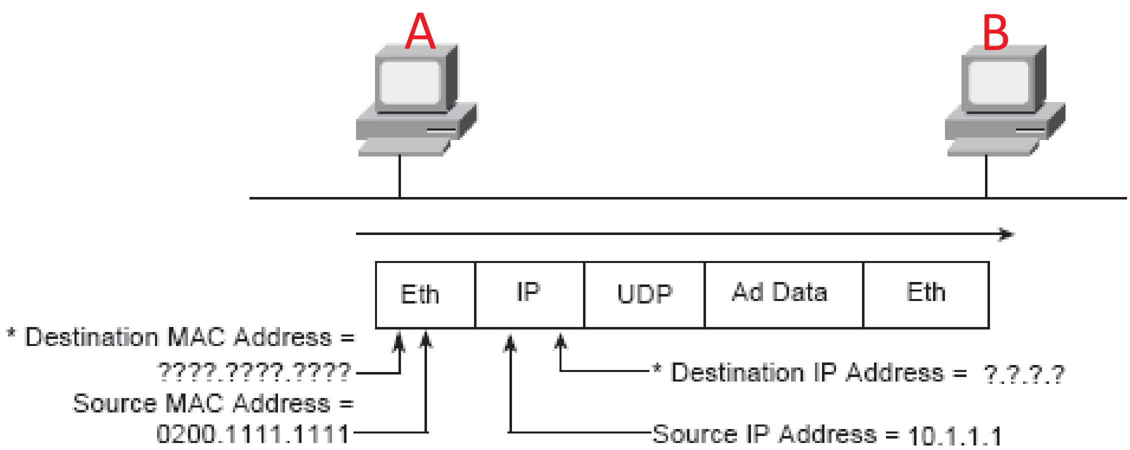
The TCP/IP network layer uses several utility protocols to help it complete its task

1. Address Resolution Protocol (ARP) and the Domain Name System
2. ICMP Echo and the ping Command
3. RARP, BOOTP, and DHCP

1. Address Resolution Protocol and the Domain Name System

- TCP/IP needs a way to find MAC addresses associated with other computers on the same LAN subnet.

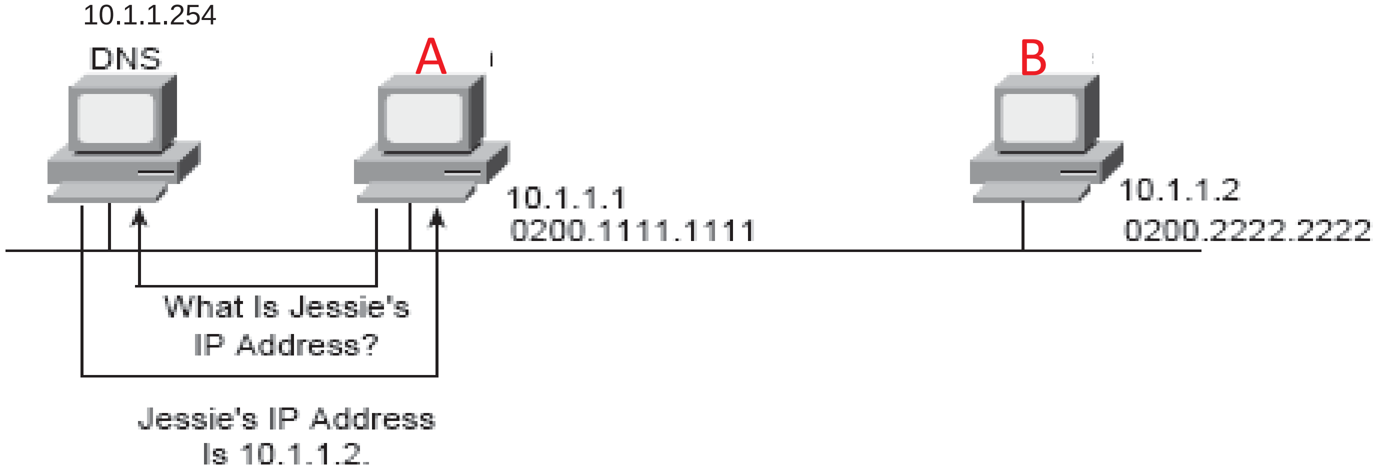
Hannah Knows Jessie's Name, Needs IP Address and MAC Address



* Information that Hannah Needs to Learn

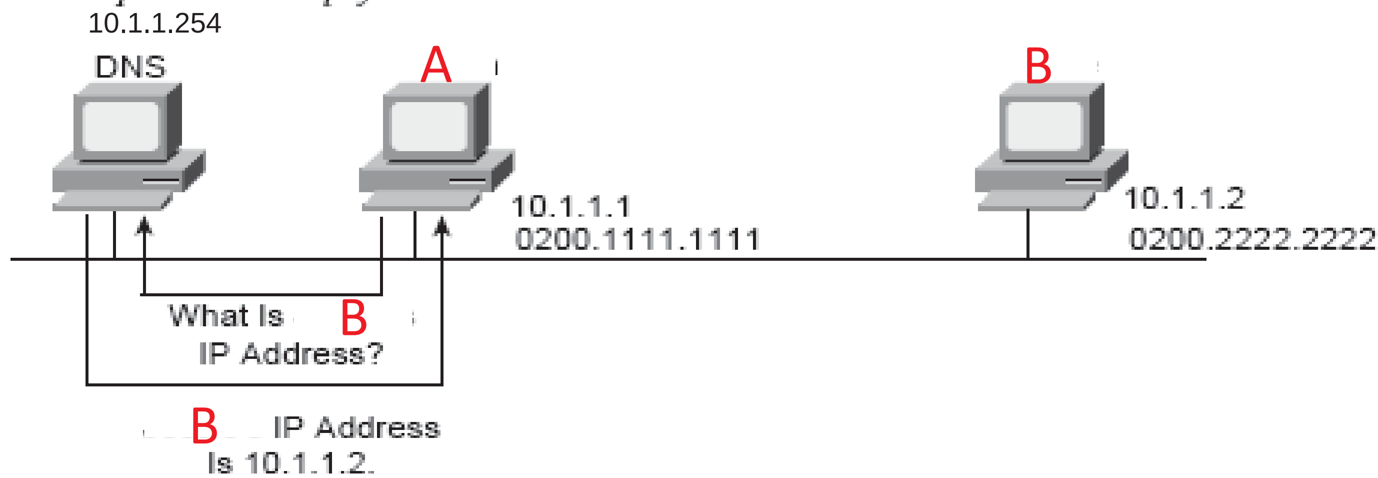
- "A" knows her own name, IP address, and MAC address because those things are configured in advance.
- *What "A" does not know are "B"'s IP and MAC addresses.*
- *To* find the two missing facts, **"A"** uses the Domain Name System (DNS) and the Address Resolution Protocol (ARP).

DNS Request and Reply



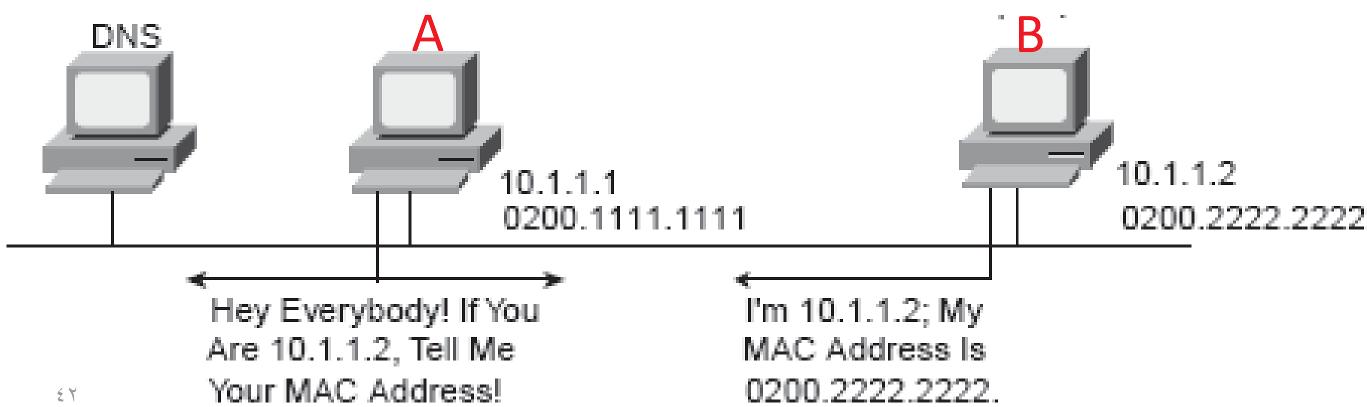
- "A" knows the IP address of a DNS server because the address was preconfigured on "A"'s machine or learned using Dynamic Host Configuration Protocol (DHCP) (will be discussed later)
- "A" now sends a *DNS request to the* DNS, asking for "B"'s IP address.
- The DNS replies with the address, 10.1.1.2.

DNS Request and Reply



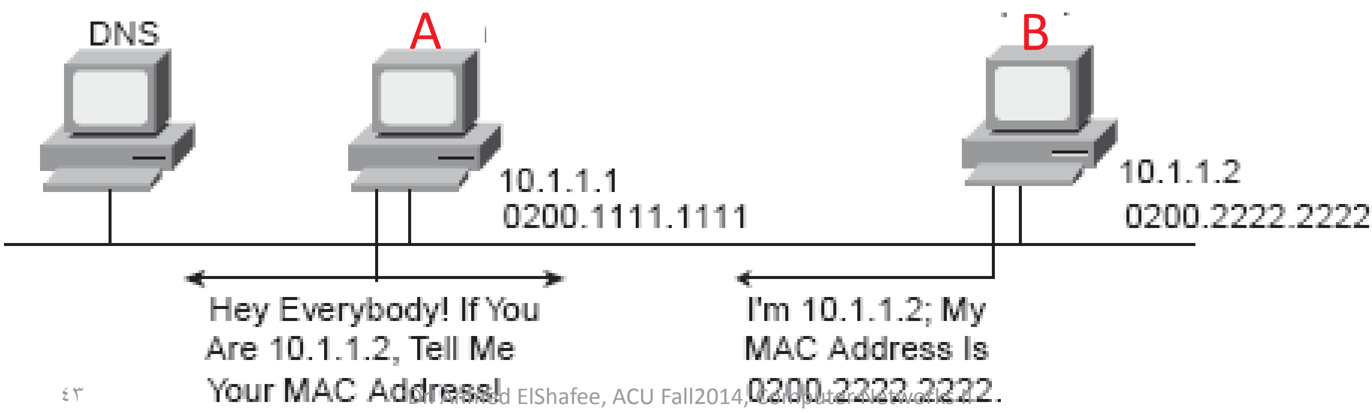
- "A" still needs to know the Ethernet MAC address used by 10.1.1.2
- "A" issues something called an *ARP broadcast.*
- An ARP broadcast is sent to a broadcast Ethernet address, so everyone on the LAN receives it.

Sample ARP Process



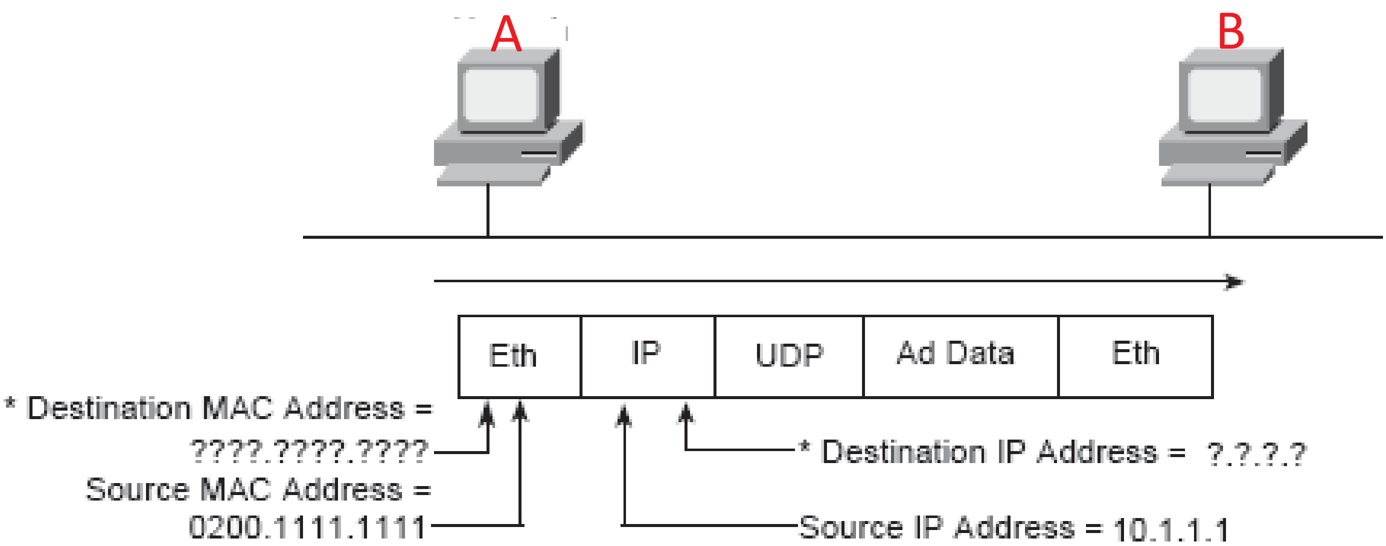
- Because “B” is on the LAN, “B” receives the ARP broadcast.
- Because “B”’s IP address is 10.1.1.2 and the ARP broadcast is looking for the MAC address associated with 10.1.1.2, “B” replies with his own MAC address.

Sample ARP Process



- Now “A” knows the destination IP and Ethernet addresses that she should use when sending frames to “B” ,

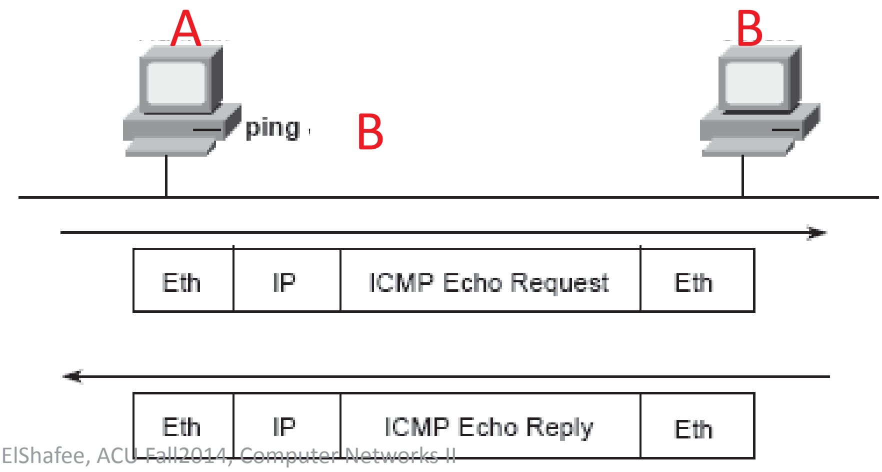
Hannah Knows Jessie’s Name, Needs IP Address and MAC Address



2. **ICMP Echo and the ping Command**

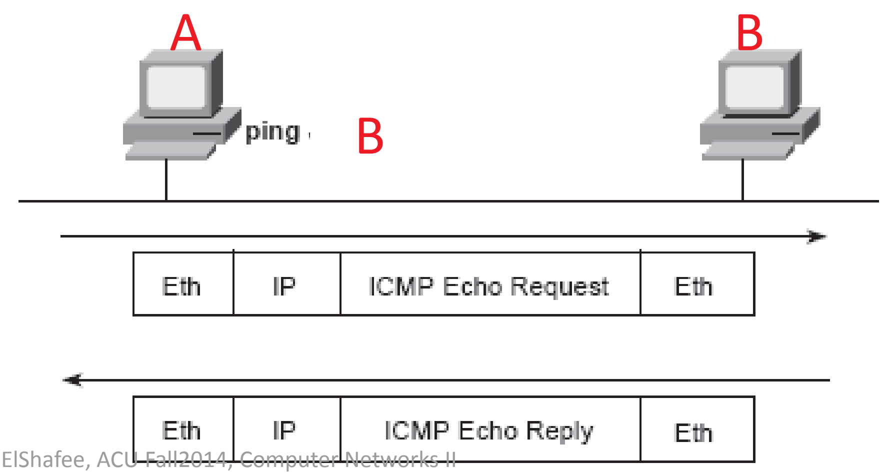
- IP needs to have a way to test basic IP connectivity, without relying on any applications to be working.
- can test basic network connectivity using the **ping command.**

Sample Network, ping Command



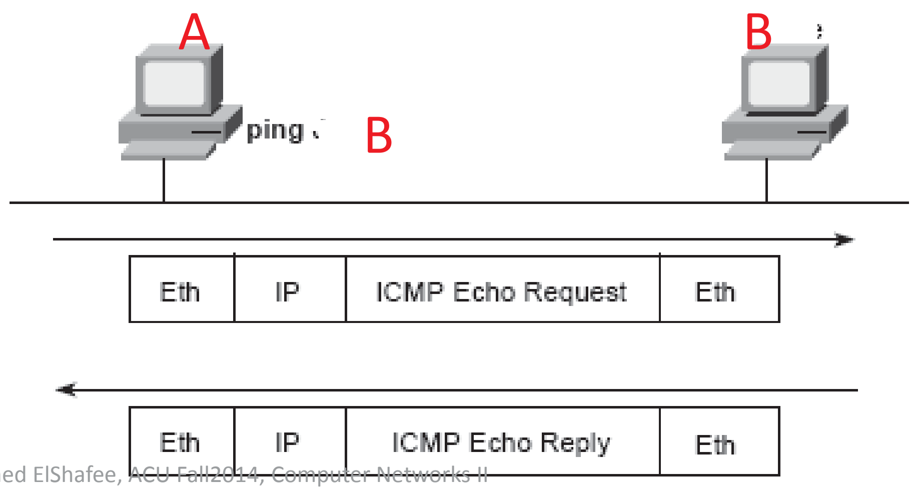
- **ping (Packet** INternet Groper) uses the *Internet Control Message Protocol (ICMP), sending a message* called an *ICMP echo request to another IP address.*
- The computer with that IP address should reply with an *ICMP echo reply.*

Sample Network, ping Command



- *If that works, you successfully have tested the IP* network.
- ICMP does not rely on any application, so it really just tests basic IP connectivity— Layers 1, 2, and 3 of the OSI model.

Sample Network, ping Command



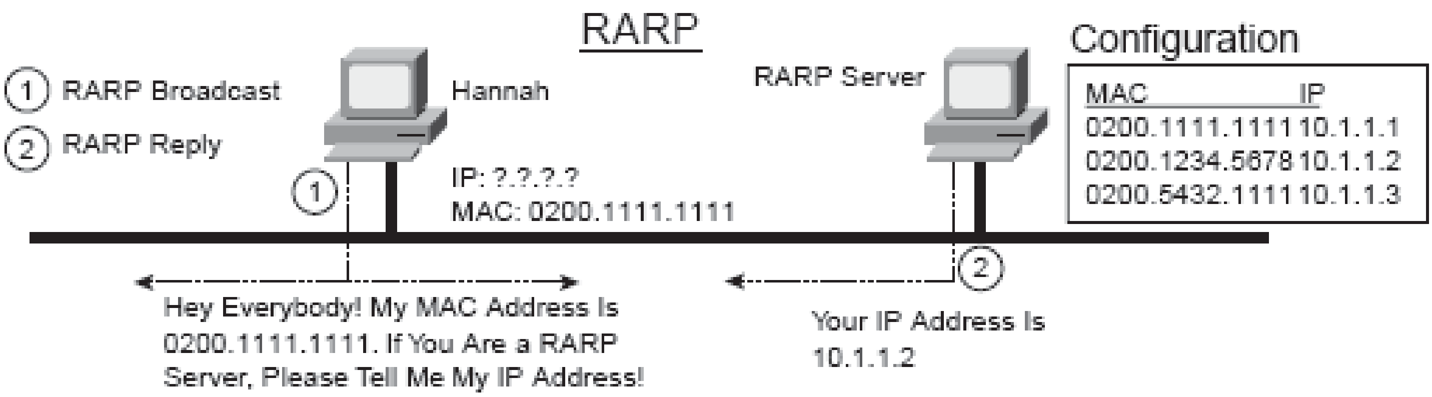
3. **RARP, BOOTP, and DHCP**

- Over the years, three protocols have been popular to allow a host computer to discover the IP address it should use:

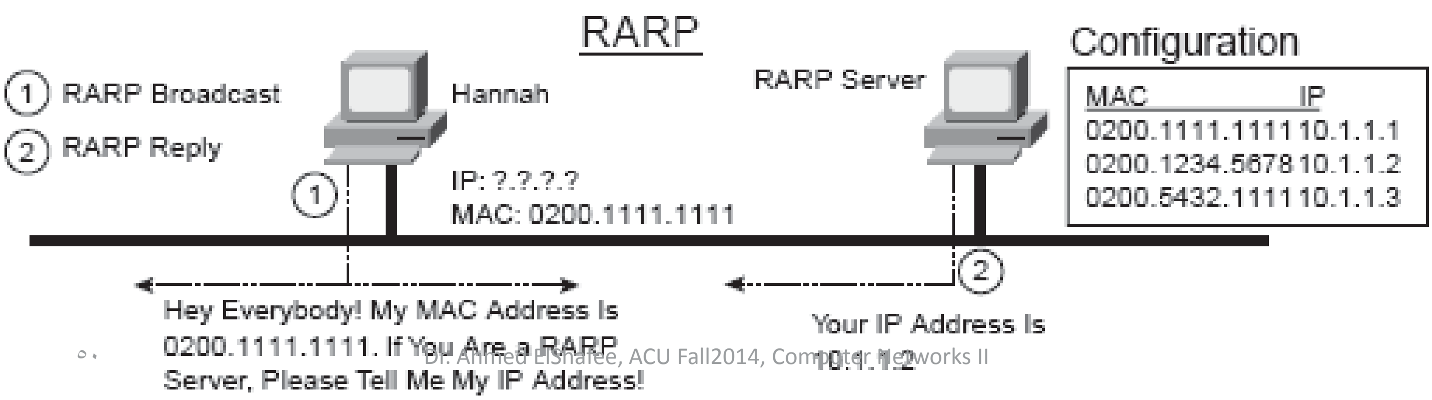
1. Reverse ARP (RARP)
2. Boot Protocol (BOOTP)
3. Dynamic Host Configuration Protocol (DHCP)

Network Layer (Layer 3) Addressing (cont,..) (13)

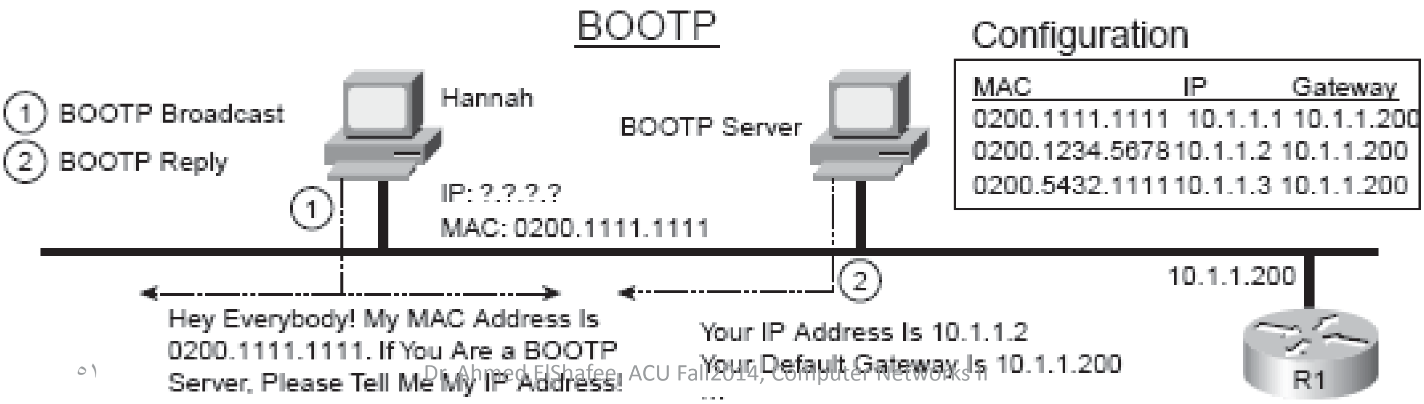
- RARP and BOOTP work using the same basic process. To use either protocol, a PC needs a LAN interface card.
- The computer sends a LAN broadcast frame announcing its own MAC address and requests that someone assign it an IP address.



- Both protocols allow for IP address assignment, but that is all that RARP can ask for—it can’t even ask for the subnet mask used on the LAN.



- BOOTP allows many more information to be announced to a BOOTP client—its IP address, its subnet mask, its default gateway IP addresses, its other server IP addresses, and the name of a file that the computer should download.



Why?

- Both RARP and BOOTP were created with the motivation to allow a diskless workstation to come up and start operating.

RARP?

With RARP, the creators of the protocol just want to get the machine an IP address so that a knowledgeable user could type in commands and copy the correct files from a server onto the diskless computer’s RAM memory so that they could be used

BOOTP?

The creators wanted to automate as much of the process as possible—including the dynamic assignment of a default gateway (router) IP address.

- BOOTP's name really comes from the feature in which BOOTP supplies the name of a file to the BOOTP client.
- Typically, the diskless workstations had enough permanent memory to boot a very simple operating system, with the expectation that the computer would use a simple protocol, such as the Trivial File Transfer Protocol (TFTP), to transfer a file containing a more sophisticated operating system into RAM.

- So, with the ultimate goal being to let a diskless computer complete the processing of initializing, or *booting, a full operating system,* BOOTP was aptly named.
- Both RARP and BOOTP clients require a computer to act as a server, and the server was required to know the MAC address of every computer and the corresponding configuration parameters that each computer should be told.
- So, administration in a network of any size was painful.

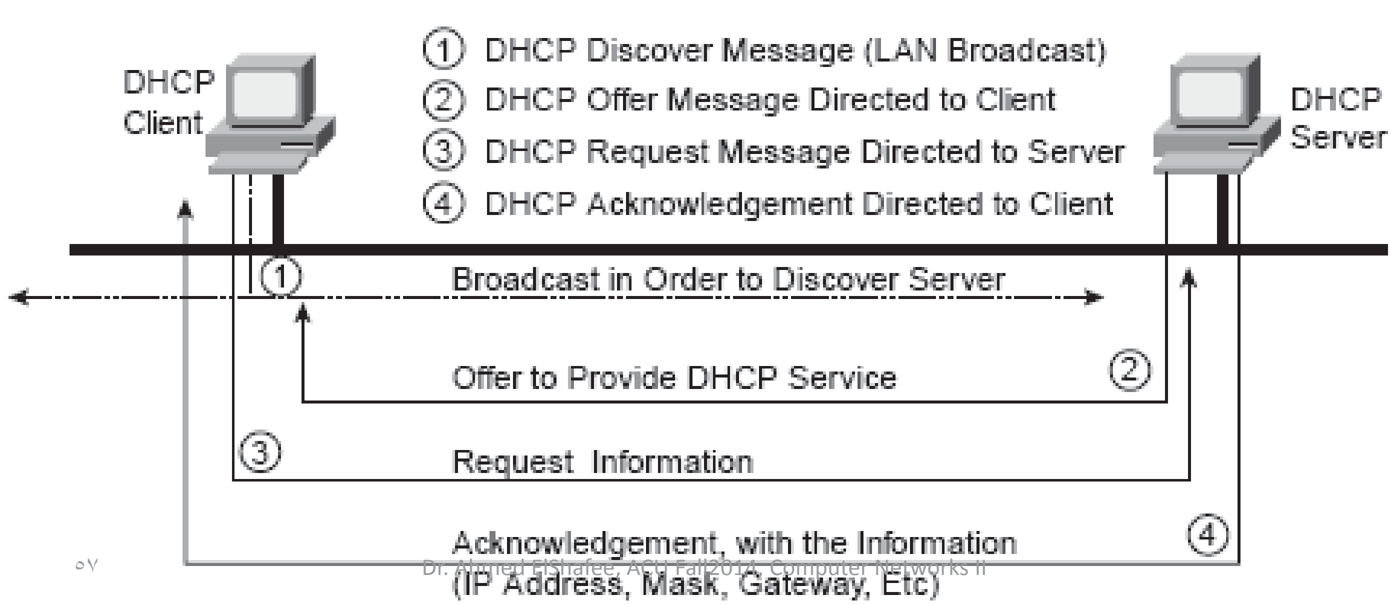
DHCP

- Like BOOTP, DHCP uses the concept of the client making a request and the server supplying the IP address to the client, plus other information such as the default gateway, subnet mask, DNS IP address, and other information.
- The biggest advantage of DHCP compared to BOOTP and RARP is that DHCP does not require that the DHCP server be configured with all MAC addresses of all clients.
- DHCP defines a process by which the server knows the IP subnet in which the DHCP client resides, and it can assign an IP address from a pool of valid IP addresses in that subnet.

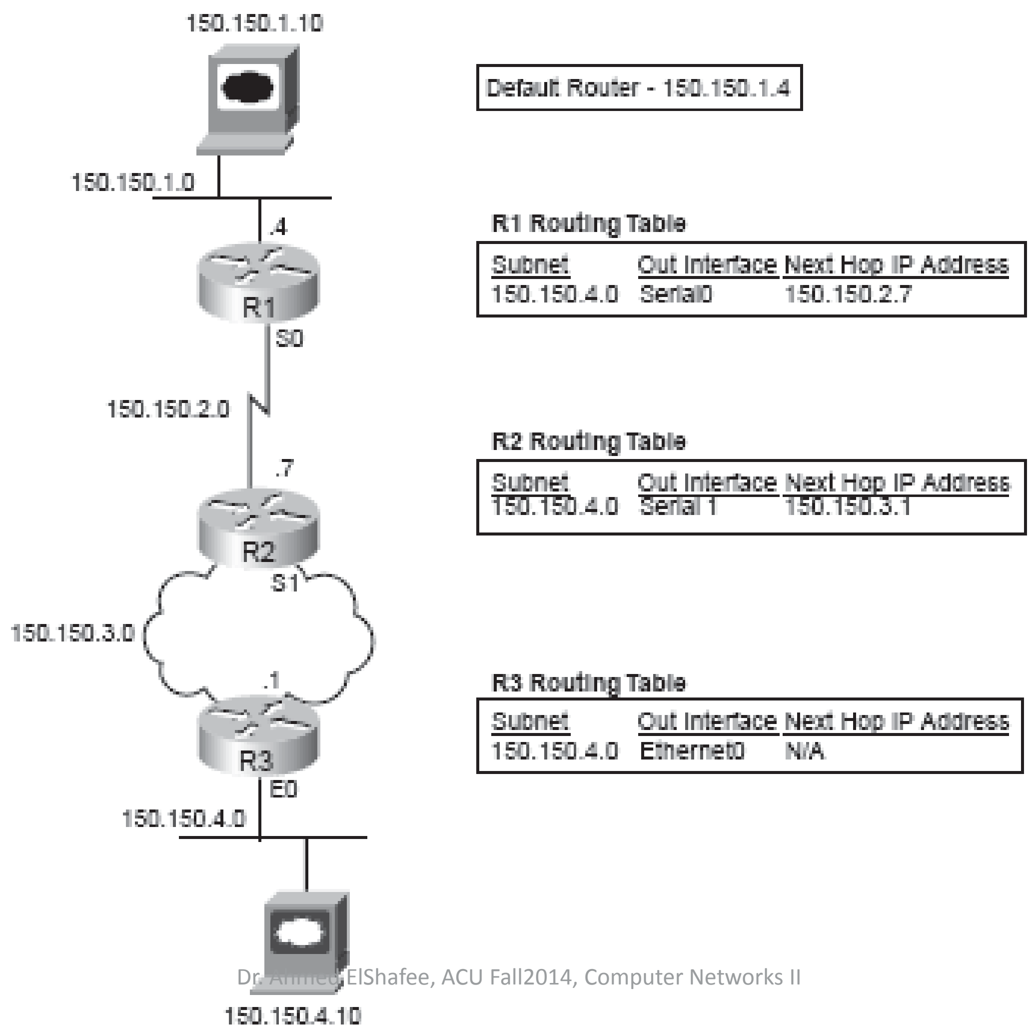
- So, the DHCP server does not need to know the MAC address ahead of time.
- Also, most of the other information that DHCP might supply, such as the default router IP address, is the same for all hosts in the same subnet, so DHCP servers simply can configure information per subnet rather than per host and save a lot of administrative hassle compared to BOOTP.

- DHCP has become a very popular protocol, with most end-user hosts on LANs in corporate networks getting their IP addresses and other basic configuration via DHCP.

DHCP Messages to Acquire an IP Address



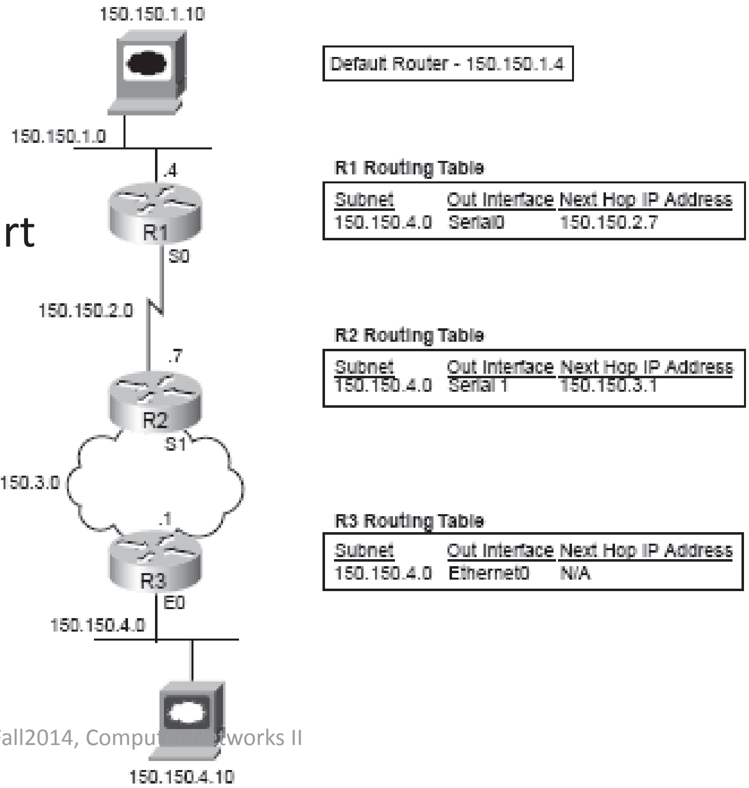
Simple Routing Example, with IP Subnets



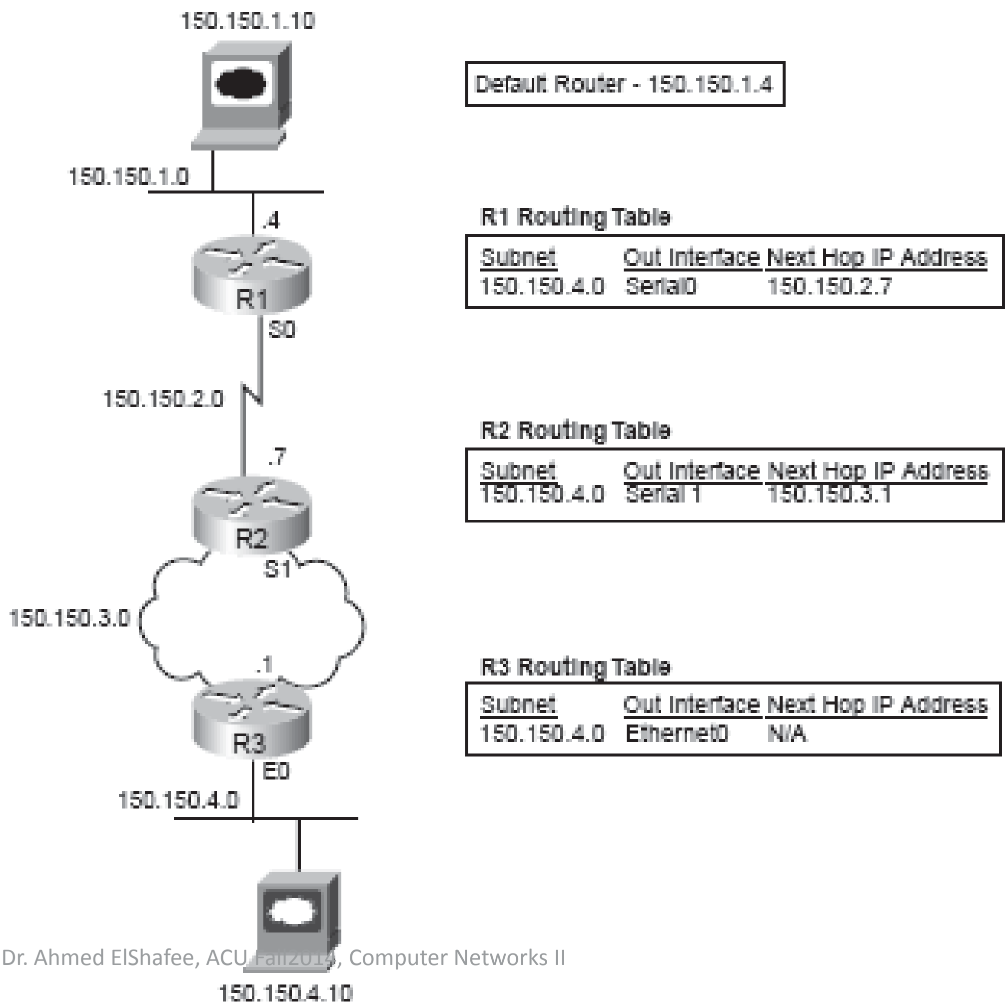
IP Routing and Routing Protocols

- First, a few detail about the figure need to be explained.
- The subnet numbers are shown, with the whole third octet used for the subnet part of the addresses.
- The actual IP addressed for PC1 and PC2 are shown.
- the full IP addresses of the routers are not shown
- only the host part of the address is listed

Simple Routing Example, with IP Subnets



Simple Routing Example, with IP Subnets

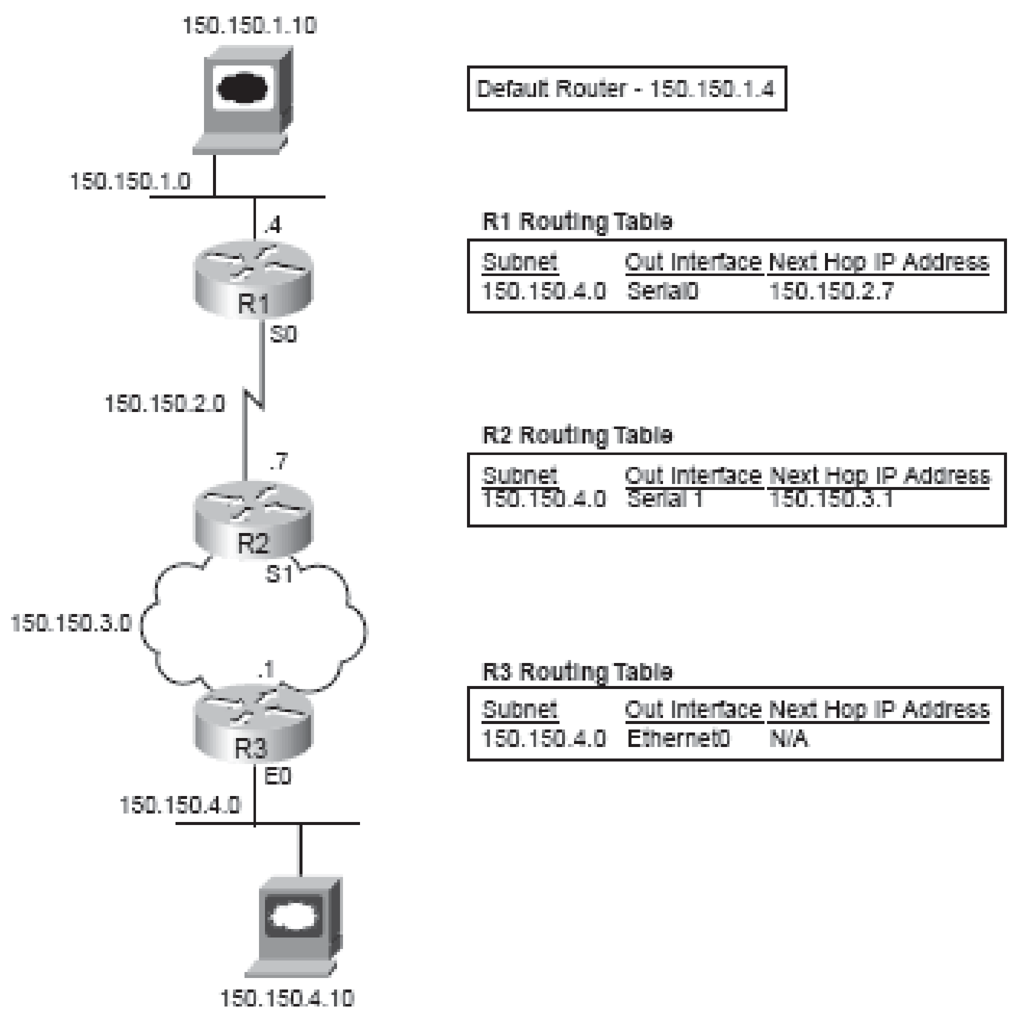


- For instance, R2's IP address on the serial link to R1 is 150.150.2.7.
- The subnet is 150.150.2.0, and the .7 shown beside R2 in the figure represents the host part of the address, which is the fourth octet in this case.

Step 1

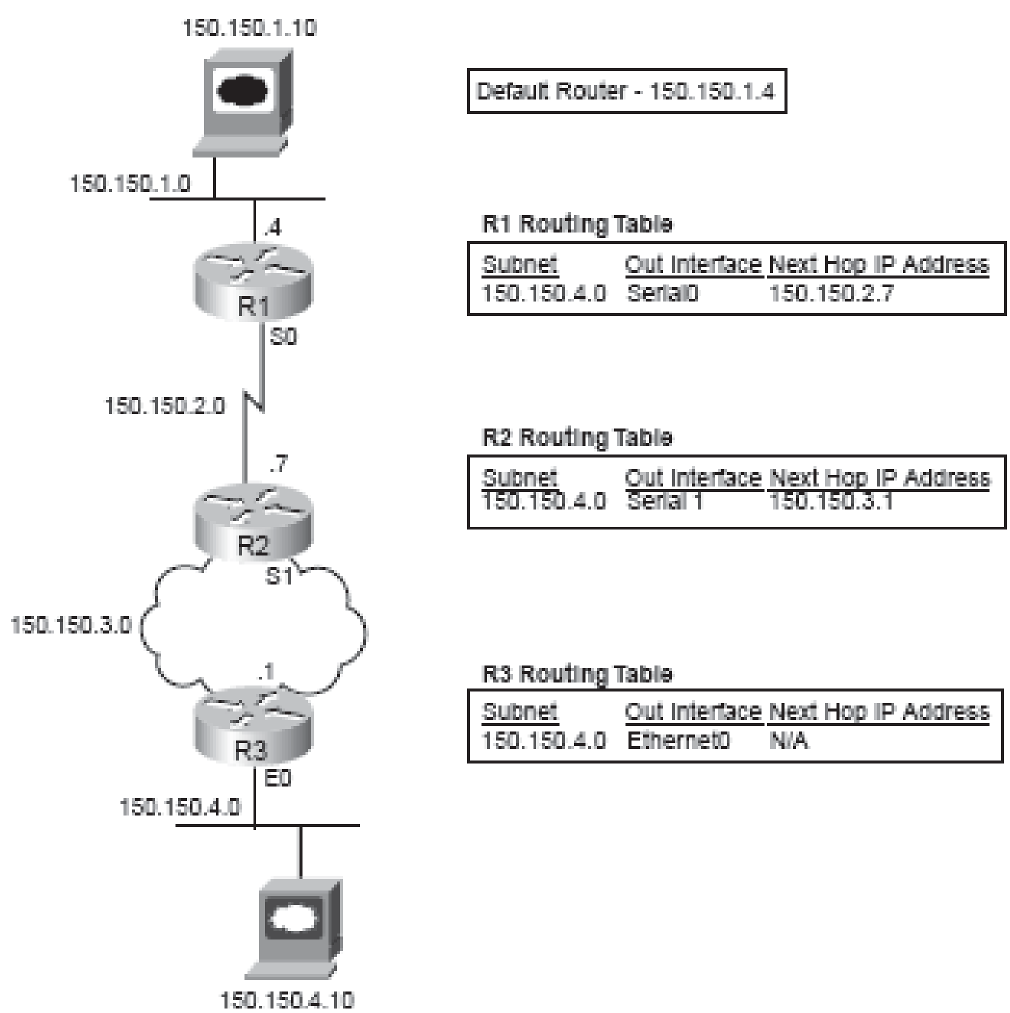
- PC1 sends the packet to R1—PC1 first builds the IP packet, with a destination address of PC2’s IP address (150.150.4.10). PC1 needs to send the packet to R1 because it knows that its default router is 150.150.1.4.
- PC1 first checks its ARP cache, hoping to find R1’s Ethernet MAC address.

Simple Routing Example, with IP Subnets



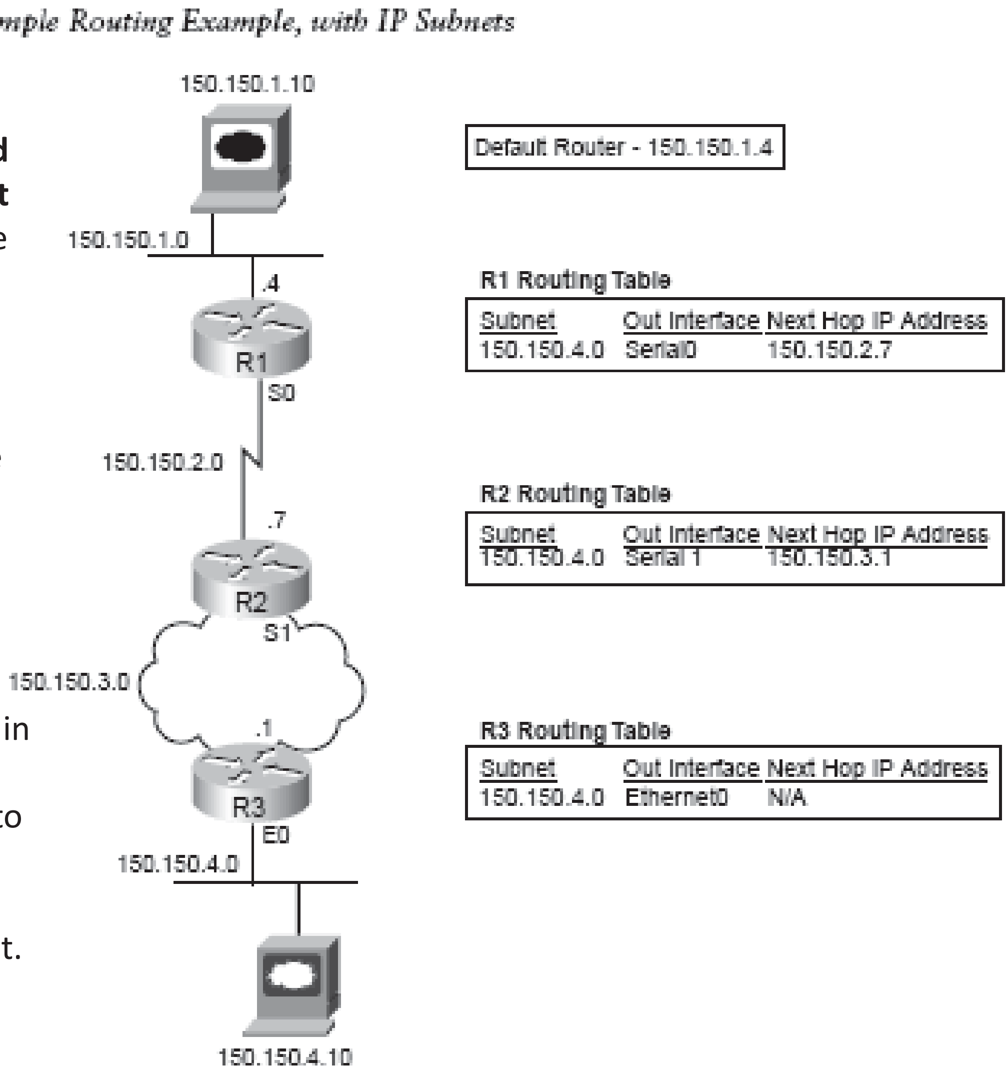
- If it is not found, PC1 ARPs to learn R1’s Ethernet MAC address.
- Then PC1 places the IP packet into an Ethernet frame, with a destination Ethernet address of R1’s Ethernet address.
- PC1 sends the frame onto the Ethernet.

Simple Routing Example, with IP Subnets

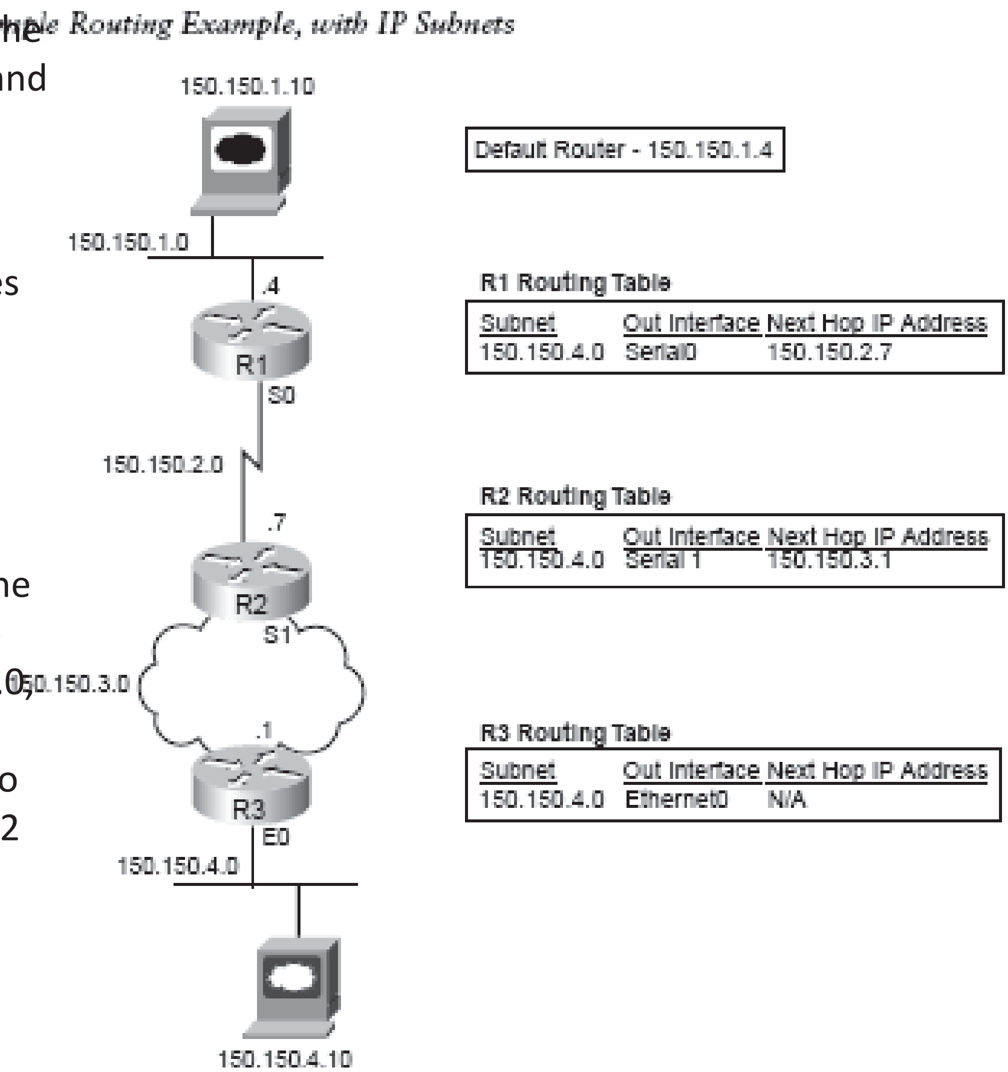


Step 2

- **R1 processes the incoming frame and forwards the packet to R2—** Because the incoming Ethernet frame has a destination MAC of R1's Ethernet MAC, R1 copies the frame off the Ethernet for processing.
- If the FCS passes, meaning that the Ethernet frame did not have any errors in it, R1 looks at the Protocol Type field to discover that the packet inside the frame is an IP packet.

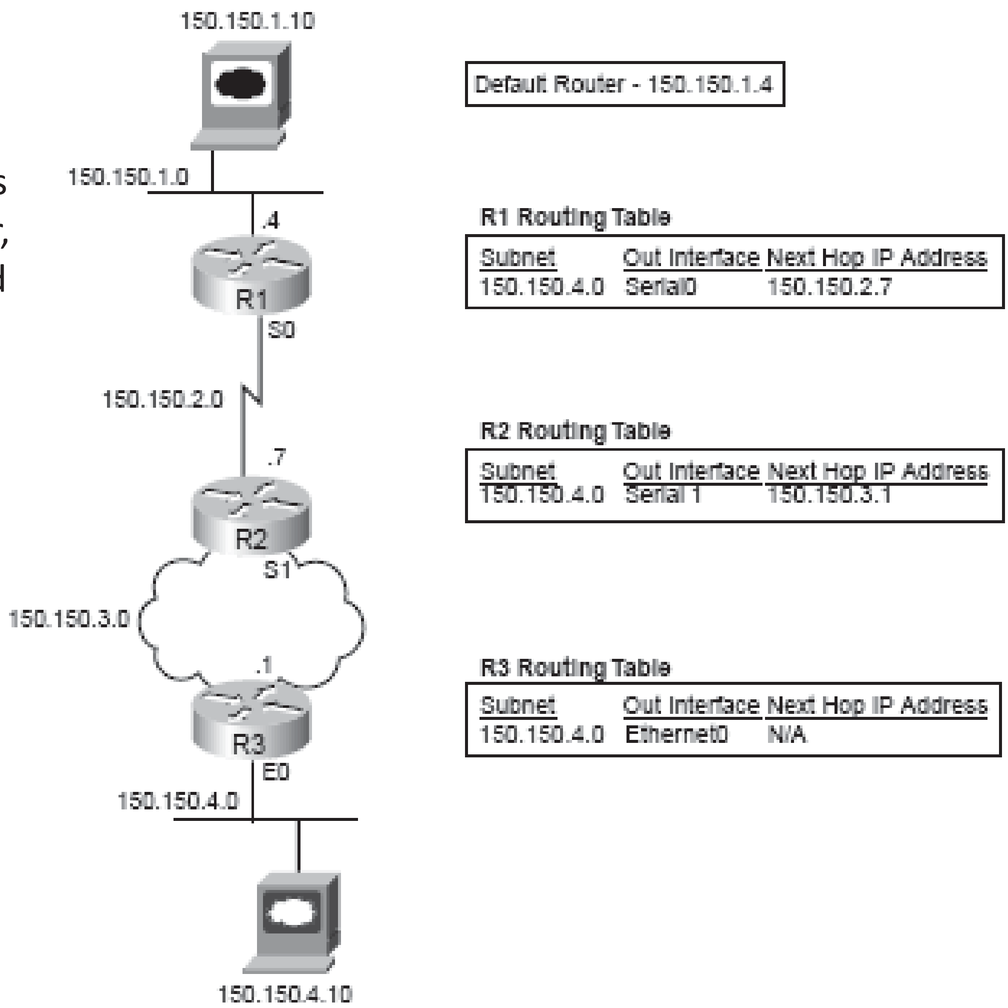


- R1 then discards the Ethernet header and trailer.
- Next, R1 looks for the routing table entry that matches the destination address in the packet, 150.150.4.10.
- The routing table entry is listed in the figure—a route to subnet 150.150.4.0, with outgoing interface Serial0 to next-hop router R2 (150.150.2.7).



Simple Routing Example, with IP Subnets

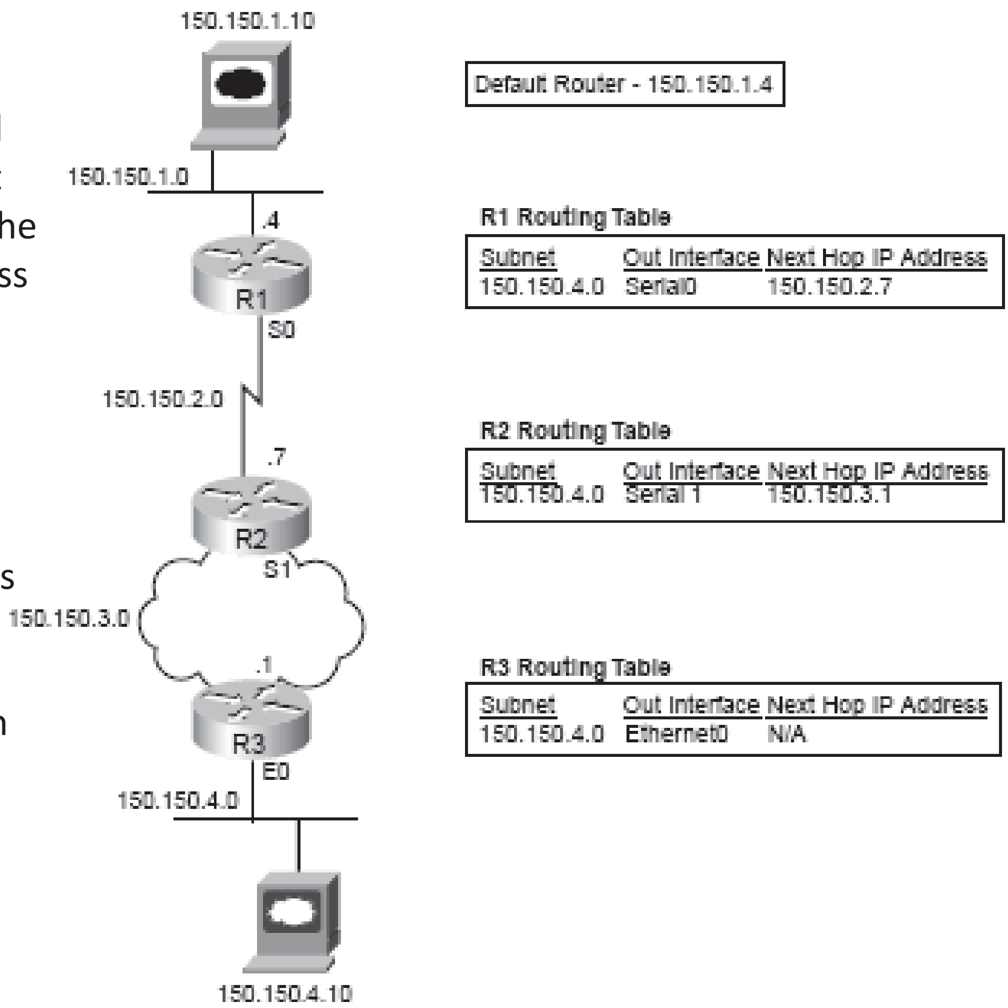
- Now R1 just needs to build an HDLC frame and send it out its Serial0 interface to R2. As mentioned earlier, ARP is not needed on a point-to point HDLC WAN link.
- R1 knows all the information necessary to out the packet inside an HDLC frame and send the frame.



Simple Routing Example, with IP Subnets

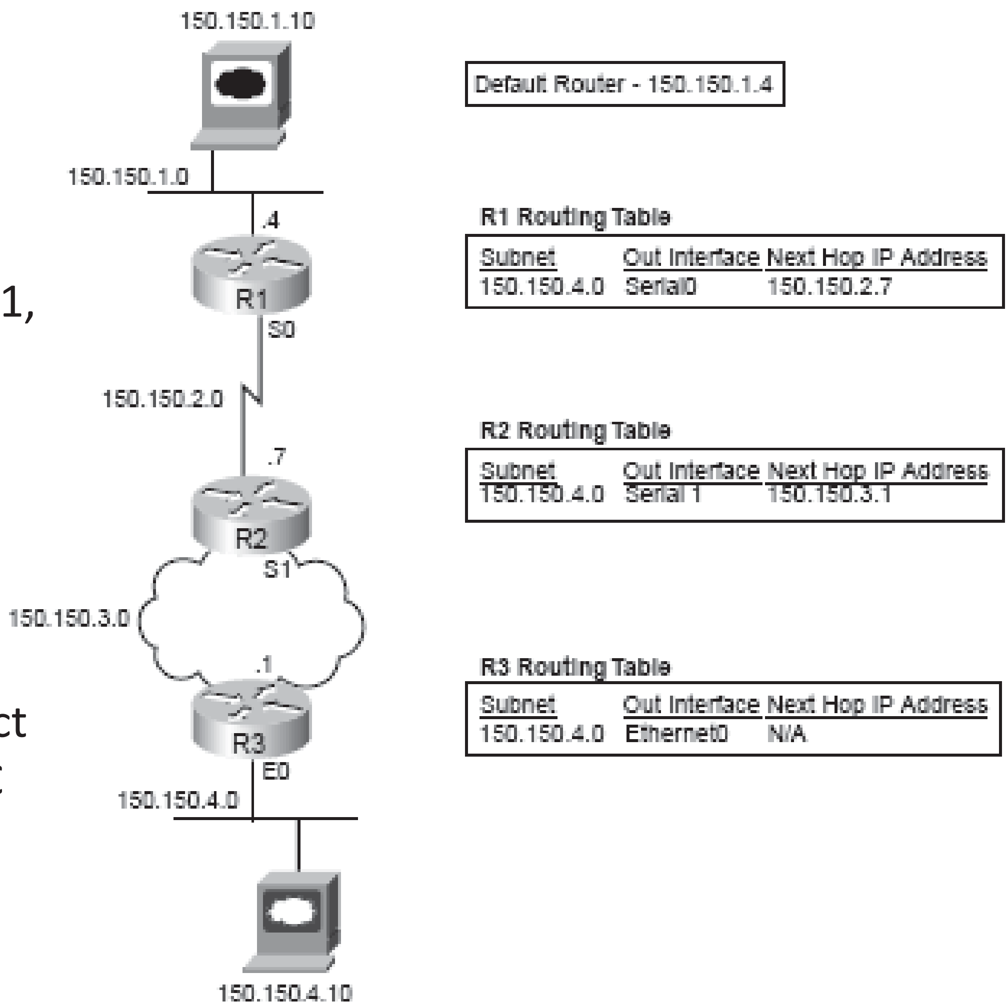
Step 3

- R2 processes the incoming frame and forwards the packet to R3—R2 repeats the same general process as R1 when it receives the HDLC frame.
- After stripping the HDLC header and trailer, R2 also needs to find the routing table entry that matches destination 150.150.4.10.



Simple Routing Example, with IP Subnets

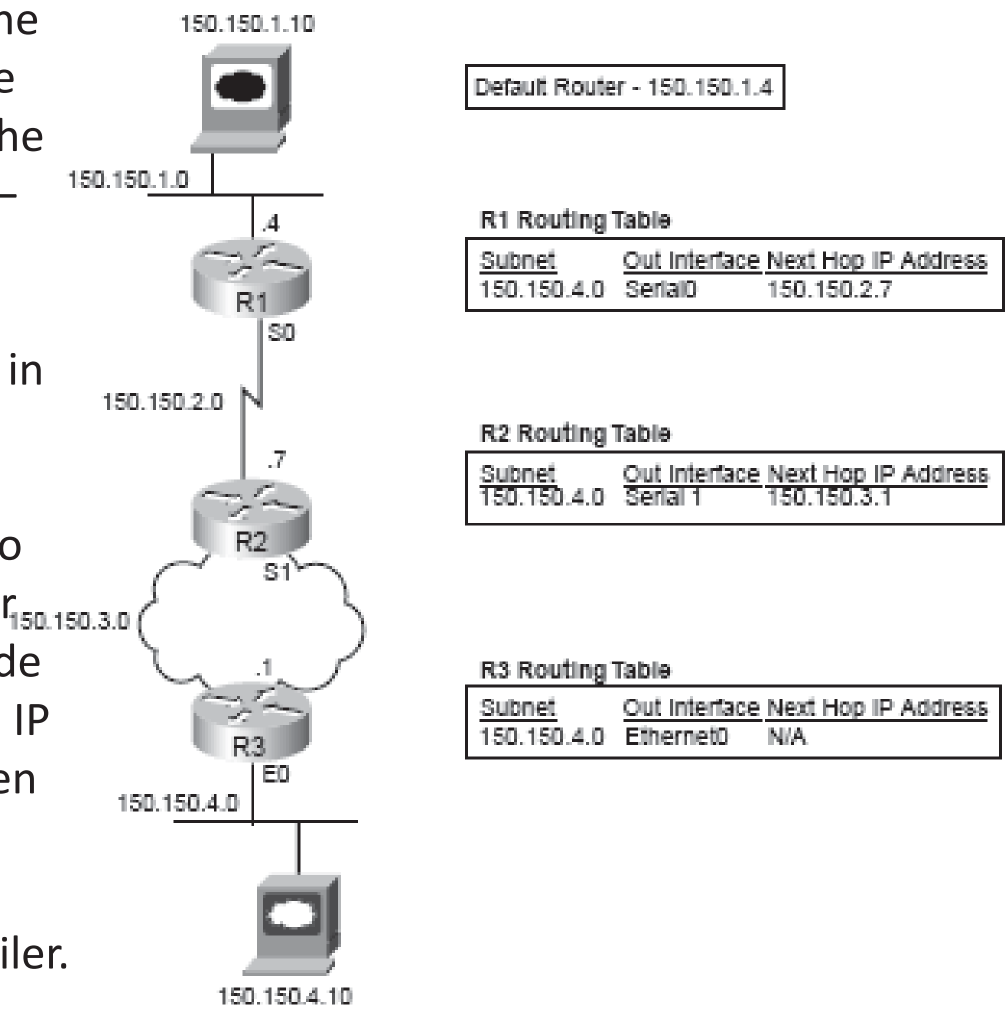
- R2's routing table has an entry for 150.150.4.0, outgoing interface serial1, to next-hop router 150.150.3.1, which is R3.
- Before R2 can complete the task, the correct DLCI for the VC to R3 must be decided.



Step 4

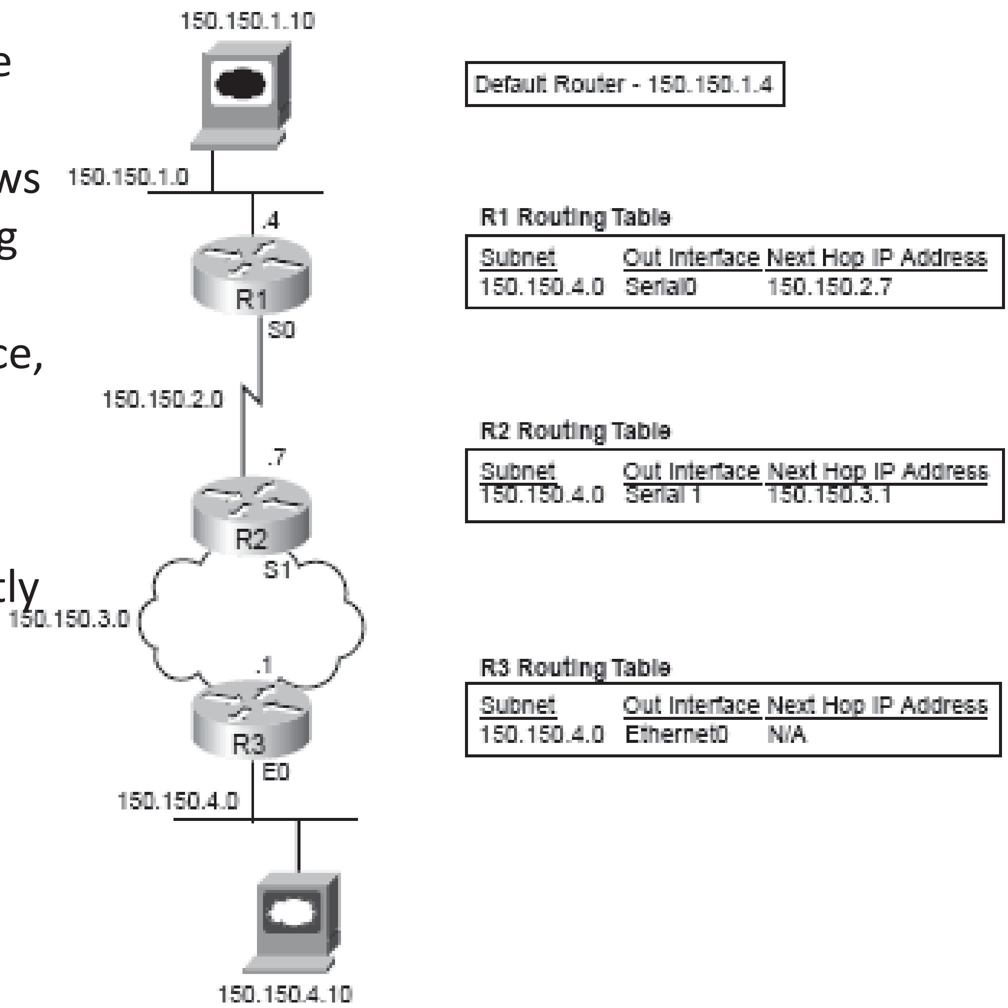
Simple Routing Example, with IP Subnets

- R3 processes the incoming frame and forwards the packet to PC2—Like R1 and R2 before it, R3 checks the FCS in the data-link trailer, looks at the type field to decide whether the packet inside the frame is an IP packet, and then discards the Frame Relay header and trailer.



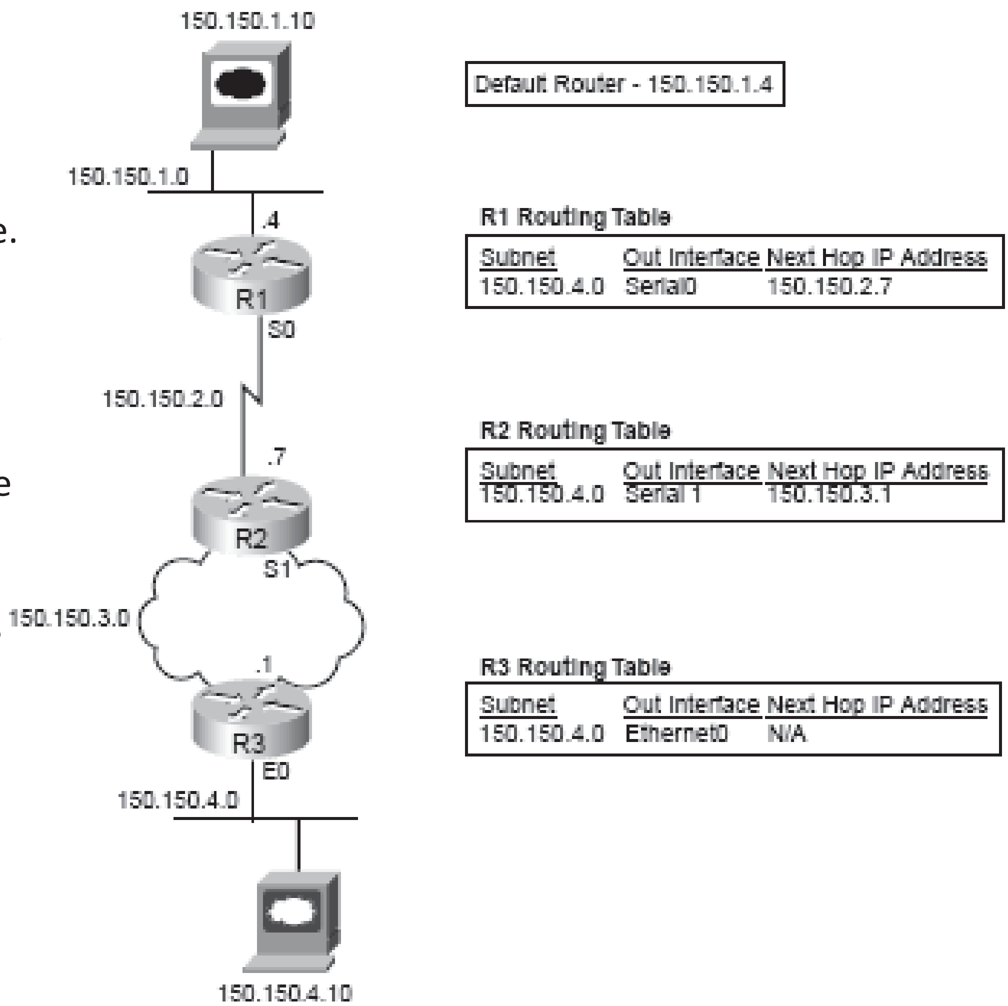
Simple Routing Example, with IP Subnets

- The routing table entry for 150.150.4.0 shows that the outgoing interface is R3's Ethernet interface, but there is no next-hop router because R3 is connected directly to subnet 150.150.4.0.



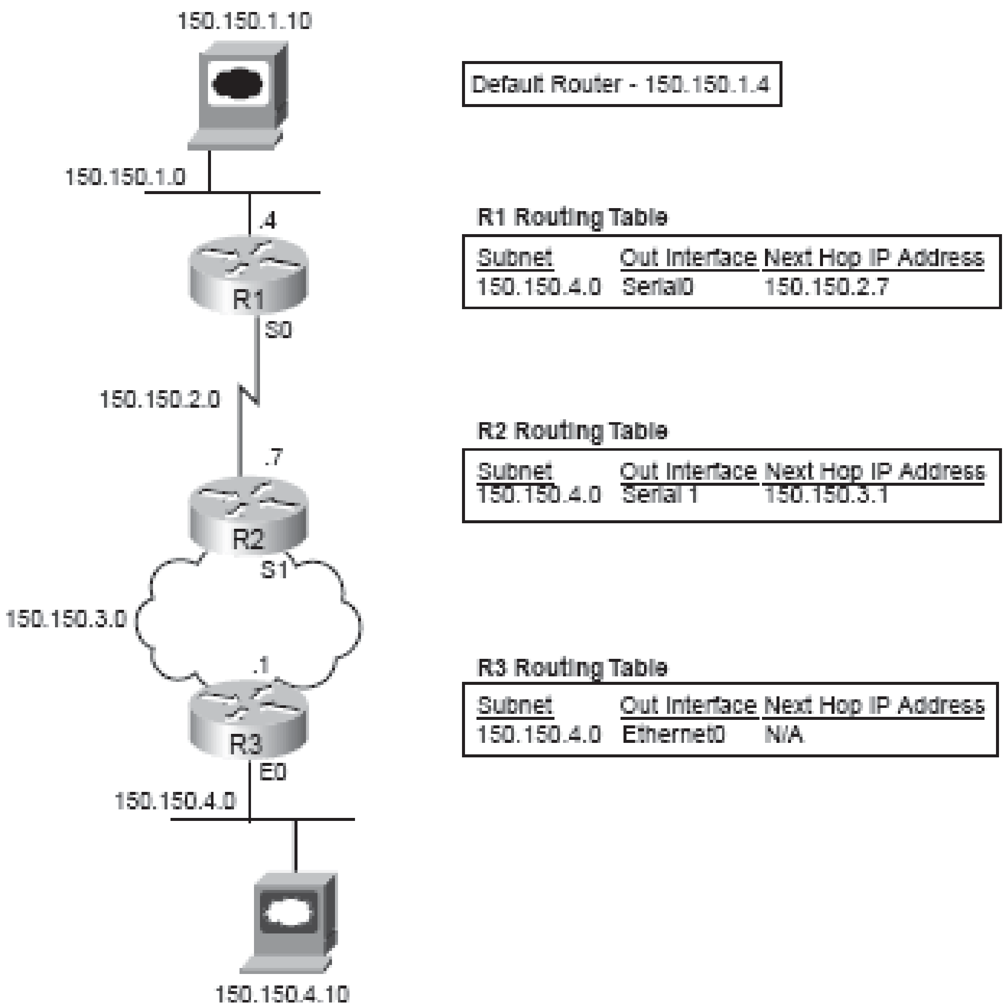
Simple Routing Example, with IP Subnets

- All R3 has to do is encapsulate the packet inside a Ethernet header and trailer, and forward the frame.
- Before R3 can finish building the Ethernet header, an IP ARP broadcast must be used to find PC2's MAC address (assuming that R3 doesn't already have that information in its IP ARP cache).



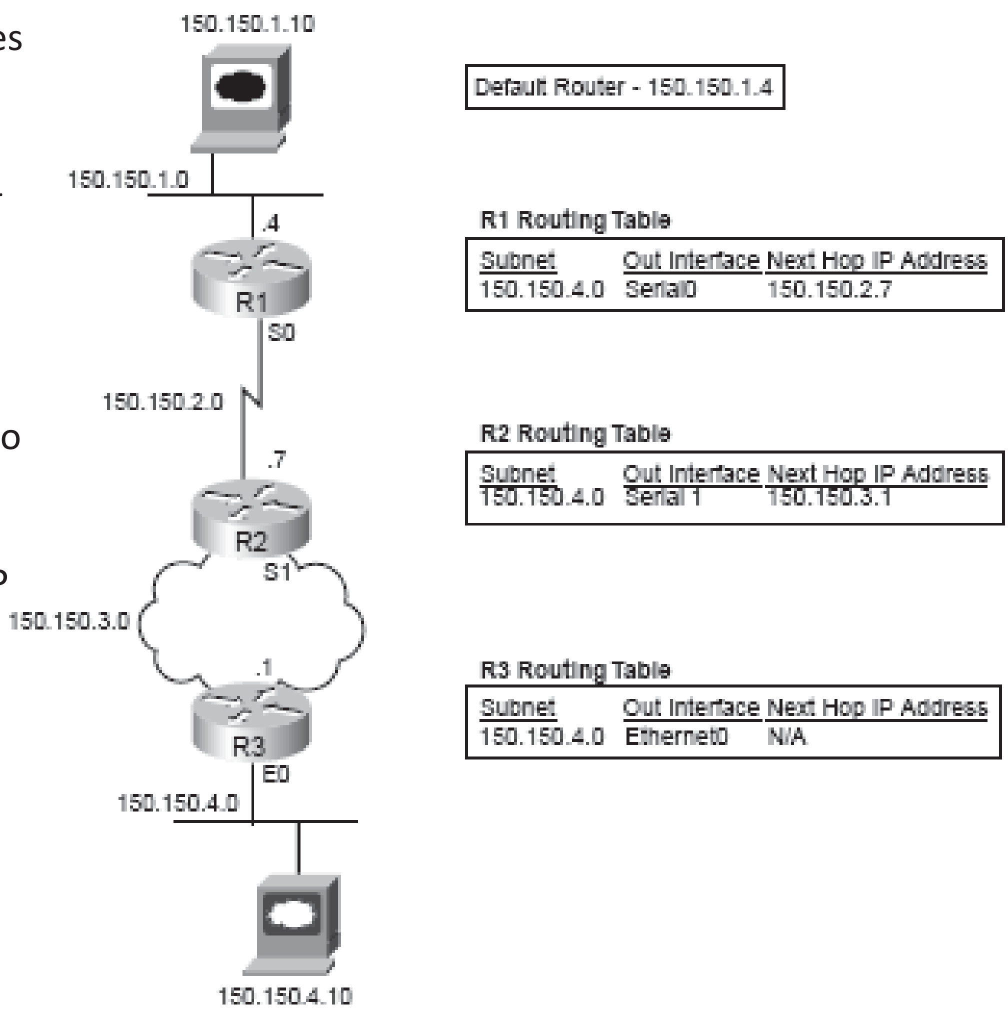
Simple Routing Example, with IP Subnets

- The routing process relies on the rules relating to IP addressing. For instance, why did 150.150.1.10 (PC1) assume that 150.150.4.10 (PC2) was not on the same Ethernet?
- Well, because 150.150.4.0, PC2's subnet, is different than 150.150.1.0, which is PC1's subnet.



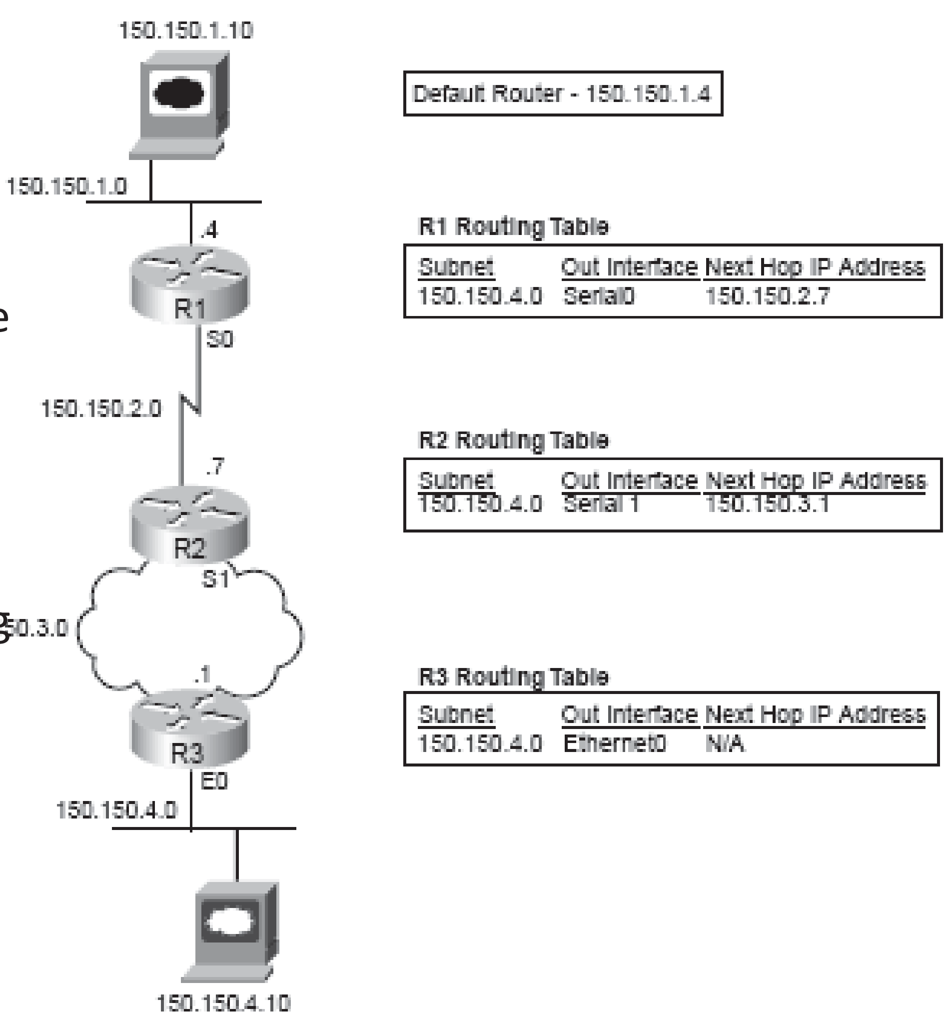
Simple Routing Example, with IP Subnets

- Because IP addresses in different subnets must be separated by some router, PC1 needed to send the packet to some router—and it did.
- Similarly, all three routers list a route to subnet 150.150.4.0, which, in this example, includes IP addresses 150.150.4.1 to 150.150.4.254.



Simple Routing Example, with IP Subnets

- What if someone tried to put PC2 somewhere else in the network, but still using 150.150.4.10? The routers then would forward packets to the wrong place.
- So, Layer 3 routing relies on the structure of Layer 3 addressing to route more efficiently.



IP Routing Protocols

- IP routing protocols fill the IP routing table with valid, (hopefully) loop-free routes.
- Each route includes a subnet number, the interface out which to forward packets so that they are delivered to that subnet, and the IP address of the next router that should receive packets destined for that subnet (if needed).

the goals of a routing protocol

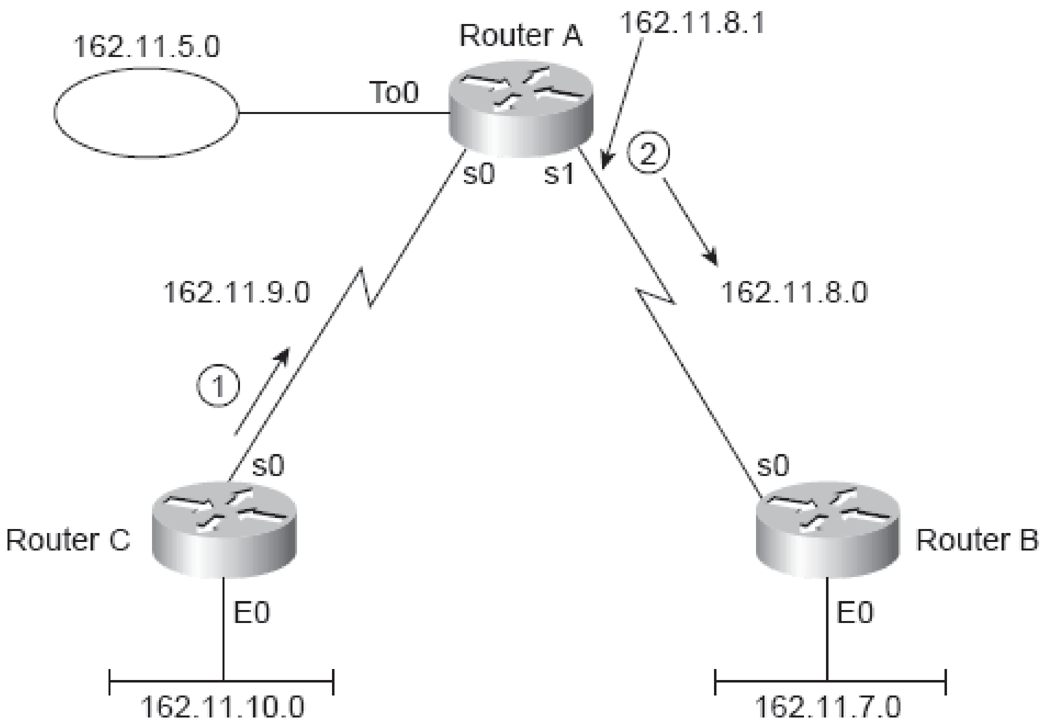
- To dynamically learn and fill the routing table with a route to all subnets in the network.
- If more than one route to a subnet is available, to place the best route in the routing table.
- To notice when routes in the table are no longer valid, and to remove those routes from the routing table.
- If a route is removed from the routing table and another route through another neighboring router is available, to add the route to the routing table.

- To add new routes, or to replace lost routes, with the best currently available route as quickly as possible.
- The time between losing the route and finding a working replacement route is called ***convergence time.***
- To prevent routing loops.

The logic

- basic logic that they use is relatively simple.
- Routing protocols take the routes in a routing table and send a message to their neighbors telling them about the routes.
- After a while, everyone has heard about all the routes.

- Figure shows a sample network, with routing updates shown.



IP Routing and Routing Protocols (20)

- Table shows Router B's routing table before receiving the routing updates,

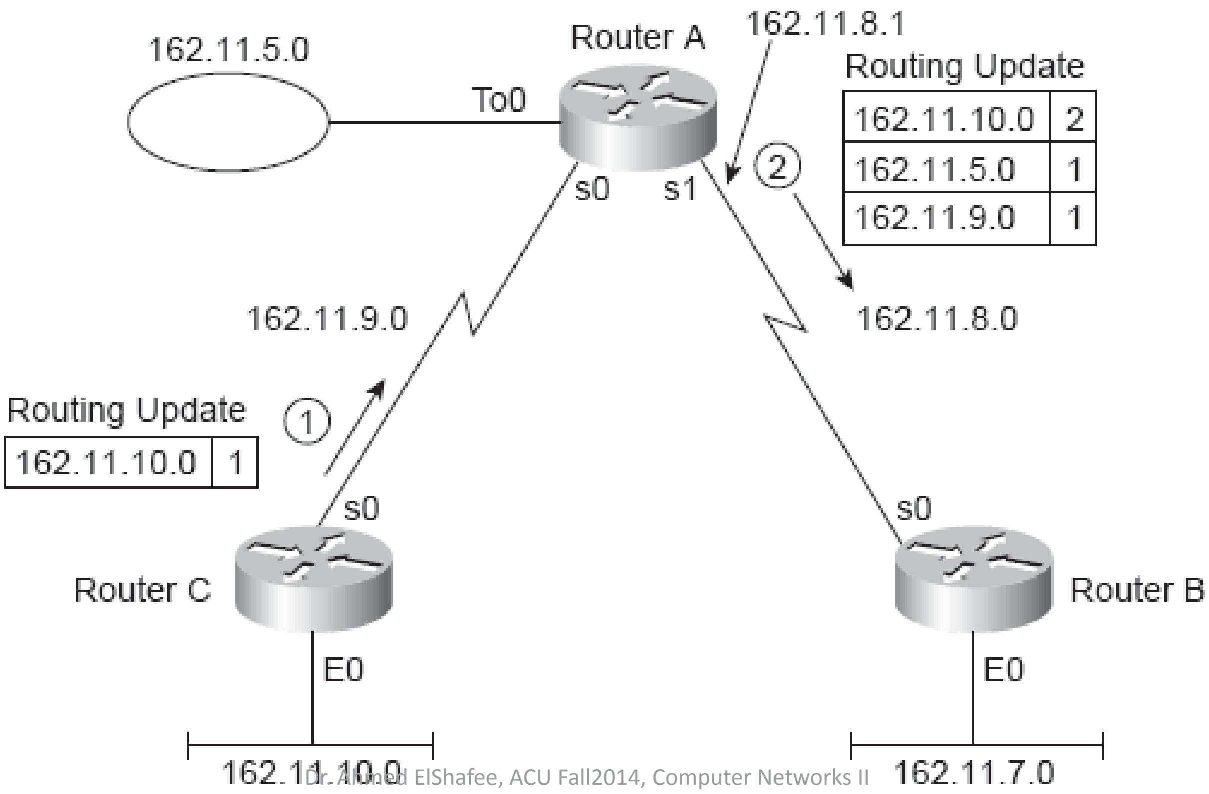
Group	Outgoing Interface	Next-Hop Router	Metric	Comments
162.11.7.0	E0	—	0	This is a directly connected route.
162.11.8.0	S0	—	0	This is a directly connected route.

- Router B adds routes for directly connected subnets when the interfaces first initialize.
- So, before Router B receives any routing updates, it knows about only two routes—the two connected routes—as listed

IP Routing and Routing Protocols (21)

- After receiving the update from Router A, Router B has learned three more routes.
- Because Router B learned those routes from Router A, all three of B's routes point back to Router A as the next hop router.

Group	Outgoing Interface	Next-Hop Router	Metric	Comments
162.11.5.0	S0	162.11.8.1	1	Learned from Router A, so next-hop is Router A.
162.11.7.0	E0	—	0	This is a directly connected route.
162.11.8.0	S0	—	0	This is a directly connected route.
162.11.9.0	S0	162.11.8.1	1	Learned from Router A, so next-hop is Router A.
162.11.10.0	S0	162.11.8.1	2	This one was learned from Router A, which learned it from Router C.



- Router A learned about subnets 162.11.5.0 and 162.11.9.0 because A is connected directly to those subnets.
- Router A, in turn, learned about subnet 162.11.10.0, the subnet off Router C's Ethernet, from routing updates sent by Router C.

Group	Outgoing Interface	Next-Hop Router	Metric	Comments
162.11.5.0	S0	162.11.8.1	1	Learned from Router A, so next-hop is Router A.
162.11.7.0	E0	—	0	This is a directly connected route.
162.11.8.0	S0	—	0	This is a directly connected route.
162.11.9.0	S0	162.11.8.1	1	Learned from Router A, so next-hop is Router A.
162.11.10.0	S0	162.11.8.1	2	This one was learned from Router A, which learned it from Router C.


Thanks,..
See you next week (ISA),…