CSE 3231
Computer Networks

Chapter 5
The Network Layer
part 2

William Allen, PhD

Spring 2022

Dotted Decimal Notation

- IP addresses are written in what is often called ***dotted decimal* notation**
  - Each byte is represented by a decimal number in the range [0..255]:

10000000	10001111	10001001	10010000
1st Byte	2nd Byte	3rd Byte	4th Byte
= 128	**= 143**	**= 137**	**= 144**

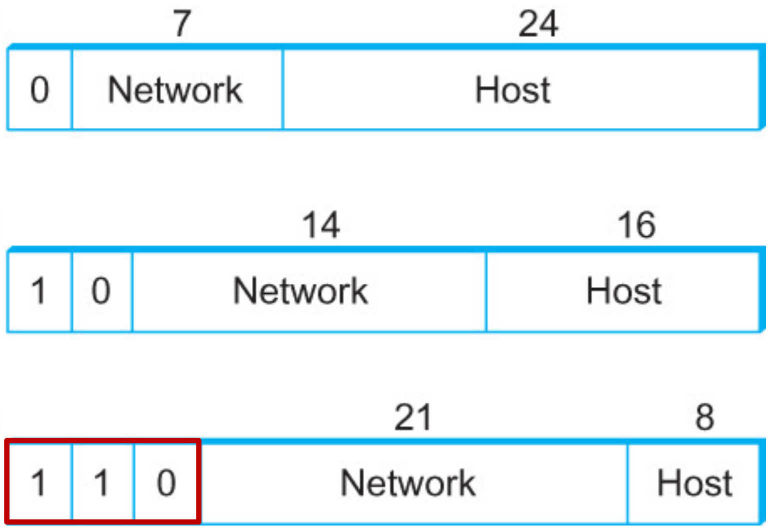
128.143.137.144

Internet Addresses

- There are two common ways to assign an IP address to a network or host
- Class-based addressing divides the IP address range to support large and small networks
  - fewer large networks, many smaller networks
- Classless Inter-Domain Routing (CIDR) is more flexible for allowing different network sizes
  - smaller networks can share a range of IP addresses to provide more efficient use of the limited range of possible addresses
  - Internet Service Providers (ISPs) can create networks more easily and allocate the size of network that a customer needs
  - however, routers must be updated regarding changes in the subdivisions

Class-based Addressing

- Properties:
  - globally unique: each host has its own IP address
  - hierarchical: address divided into network + host
  - 32-bit number = ~4 Billion IP addresses
- IP Address Classes:
  - Class A: leading bit = `0`
    - total of ~120 networks
    - 16,777,214 hosts per network
  - Class B: leading bits = `10`
    - total of ~16,000 networks
    - 65,534 hosts per network
  - Class C: leading bits = `110`
    - can be ~2 million networks
    - 254 hosts per network



Classless Inter-Domain Routing (CIDR)

- Why did we need Classless Addressing?
  - Two major concerns with the growth of the Internet
    - Routing tables in backbone routers grow quickly in size as more and more network address numbers must be added
      - backbone routers must store the addresses of all of the networks
    - The potential for exhaustion of the 32-bit address space because there are only ~4.2 billion possible addresses
  - Address assignment efficiency
    - Arises because of the IP address structure with class A, B, and C addresses forces us to hand out network address space in fixed-size chunks of three very different sizes
      - A network with two hosts needs a class C address
        - » Address assignment efficiency = 2/255 = 0.78%
      - A network with 256 hosts needs a class B address
        - » Address assignment efficiency = 256/65,535 = 0.39%

The Limited Number of IP Addresses

- One major problem with the limited range of IP address space is the allocation of poorly-used class B networks
  - many organizations need > 254 addresses, but not 10’s of thousands
- One Solution
  - Deny requests for a class B address unless the requester can show a need for something close to 64K addresses
  - Instead give them an appropriate number of class C addresses
    - e.g., if they need 10,000 addresses, give them ~40 class C addresses
  - If each of those networks is fully allocated before the next one is used, the unused addresses will be in only one class C network, an average of around 128 (50%) unused addresses
  - If class B addresses are allocated, the number of unused addresses will average around 32,000 addresses that would not be available for anyone else

Classless Addressing

- However, this can cause an excessive storage requirement at the routers
- If a single ISP (Internet Service Provider) has 16 class C network numbers assigned to it
  - *Every* Internet backbone router needs 16 entries in its routing tables for that same ISP
  - This is true, even if the path to every one of these networks is the same
- If we had assigned a class B address to the ISP
  - They store the same routing information in one entry
  - But, address efficiency = $16 \times 255 / 65{,}536 = 6.2\%$

Classless Addressing

- Classless Inter-Domain Routing (CIDR) tries to balance the desire to minimize the number of routes that a router needs to know against the need to hand out addresses efficiently.
- CIDR uses *aggregate routes*
  – Uses a single entry in the forwarding table to tell the router how to reach a number of related networks
  – Also, this breaks the rigid boundaries between address classes
    - network sizes are based on any powers of 2 within a range

Classless Addressing

- Consider an ISP with 16 class C network numbers.
- Instead of handing out 16 addresses at random, hand out a block of *contiguous* class C addresses
- Suppose we assign the class C network numbers from 192.4.16.x through 192.4.31.x
- Observe that the first 20 bits of *all the addresses* in this range are the same (`11000000 00000100 0001`)
  - We have created a 20-bit network number, which is in between class B network number (16 bit) and class C number (24 bit)
  - This allows the ISP to allocate a range of class C addresses with that same common prefix (the leading 20 bits are the same) and the Internet's backbone routers only have to store one address for the entire range of class C networks this ISP allocated

Address Range for 16 Class C Networks

All of these addresses are below 192.4.16.000

192.4.15.000 = 11000000 00000100 00001111 00000000
192.4.15.255 = 11000000 00000100 00001111 11111111

Someone else's network addresses

The ISP's addresses

This range includes 16 class C networks and all of the addresses in this range have the prefix 192.4.16.xxx

192.4.16.000 = 11000000 00000100 00010000 00000000
192.4.16.255 = 11000000 00000100 00010000 11111111
192.4.17.000 = 11000000 00000100 00010001 00000000
192.4.17.255 = 11000000 00000100 00010001 00000000

•••

192.4.30.000 = 11000000 00000100 00011110 00000000
192.4.30.255 = 11000000 00000100 00011110 00000000
192.4.31.000 = 11000000 00000100 00011111 00000000
192.4.31.255 = 11000000 00000100 00011111 11111111

The ISP's addresses

Someone else's network addresses

All of these addresses are 192.4.32.000 or higher

192.4.32.000 = 11000000 00000100 00100000 00000000
192.4.32.255 = 11000000 00000100 00100000 11111111

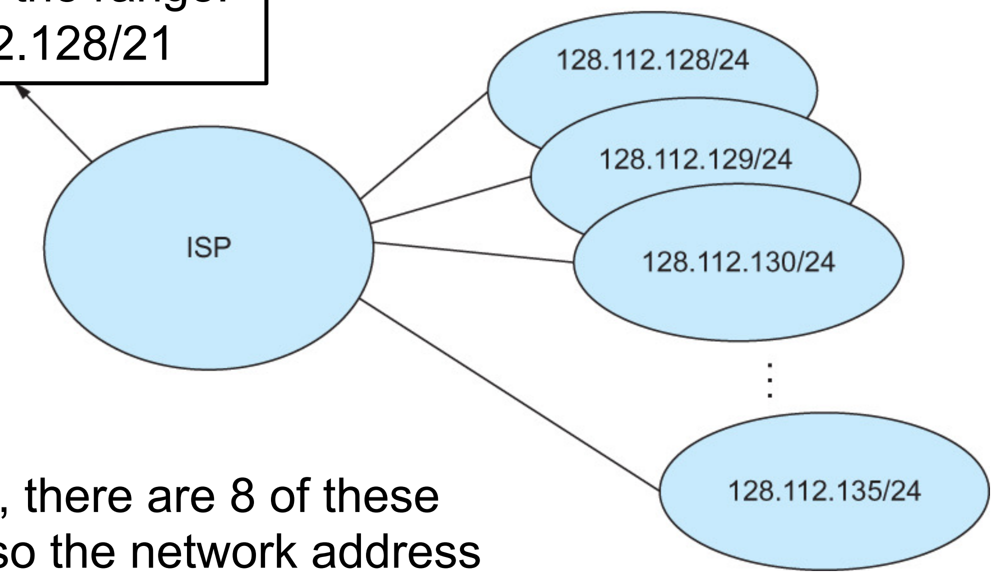
Classless Addressing

- CIDR requires we hand out blocks of class C network addresses that share a common prefix
  - How do routers "know" how many bits of this network address are in the prefix?
    - with class-based addressing the prefixes were a fixed size
  - The convention is to place a **`/X`** after the prefix where **`X`** is the *prefix length* in bits
    - for example, networks in the range `192.4.16` through `192.4.31` use a 20-bit prefix and are represented as `192.4.16/20`
  - By contrast, if we wanted to represent a single class C network address, which is 24 bits long, we would represent it with `192.4.16/24`

Route aggregation with CIDR

The ISP tells the backbone routers that these networks are all in the range:
128.112.128/21

Networks allocated by the ISP



In this example, there are 8 of these `/24` networks, so the network address that covers this range of networks must use 3 bits to separate them internally and the ISP reports this as a `/21` network

CIDR Addresses

More bits in the prefix

Fewer hosts in the network

Network Bits	Subnet Mask	Number of Subnets	Number of Hosts
/8	255.0.0.0	0	16777214
/9	255.128.0.0	2 (0)	8388606
/10	255.192.0.0	4 (2)	4194302
/11	255.224.0.0	8 (6)	2097150
/12	255.240.0.0	16 (14)	1048574
/13	255.248.0.0	32 (30)	524286
/14	255.252.0.0	64 (62)	262142
/15	255.254.0.0	128 (126)	131070
/16	255.255.0.0	256 (254)	65534
/17	255.255.128.0	512 (510)	32766
/18	255.255.192.0	1024 (1022)	16382
/19	255.255.224.0	2048 (2046)	8190
/20	255.255.240.0	4096 (4094)	4094
/21	255.255.248.0	8192 (8190)	2046
/22	255.255.252.0	16384 (16382)	1022
/23	255.255.254.0	32768 (32766)	510
/24	255.255.255.0	65536 (65534)	254
/25	255.255.255.128	131072 (131070)	126
/26	255.255.255.192	262144 (262142)	62
/27	255.255.255.224	524288 (524286)	30
/28	255.255.255.240	1048576 (1048574)	14
/29	255.255.255.248	2097152 (2097150)	6
/30	255.255.255.252	4194304 (4194302)	2

Smaller networks have a longer prefix (network portion) and a correspondingly shorter host address section

Mapping Class B into CIDR

Network Bits	Subnet Mask	Number of Subnets	Number of Hosts
/16	255.255.0.0	256 (254)	65534
/17	255.255.128.0	512 (510)	32766
/18	255.255.192.0	1024 (1022)	16382
/19	255.255.224.0	2048 (2046)	8190
/20	255.255.240.0	4096 (4094)	4094
/21	255.255.248.0	8192 (8190)	2046
/22	255.255.252.0	16384 (16382)	1022
/23	255.255.254.0	32768 (32766)	510
/24	255.255.255.0	65536 (65534)	254
/25	255.255.255.128	131072 (131070)	126
/26	255.255.255.192	262144 (262142)	62
/27	255.255.255.224	524288 (524286)	30
/28	255.255.255.240	1048576 (1048574)	14
/29	255.255.255.248	2097152 (2097150)	6
/30	255.255.255.252	4194304 (4194302)	2

Mapping Class C into CIDR

Network Bits	Subnet Mask	Number of Subnets	Number of Hosts
/24	255.255.255.0	65536 (65534)	254
/25	255.255.255.128	131072 (131070)	126
/26	255.255.255.192	262144 (262142)	62
/27	255.255.255.224	524288 (524286)	30
/28	255.255.255.240	1048576 (1048574)	14
/29	255.255.255.248	2097152 (2097150)	6
/30	255.255.255.252	4194304 (4194302)	2

IP Forwarding

- The IP forwarding mechanism assumes that it can find the network number in a packet and then look up that number in the forwarding table
- We need to change this assumption for Classless Inter-Domain Routing (CIDR)
  - the *prefix* is the network portion of the address
  - prefixes may be of any length, from 2 to 32 bits
- It is also possible to have prefixes in the forwarding tables that overlap
  - some addresses may match more than one prefix

IP Forwarding

- For example, forwarding table of a single router might contain both 171.69 (a 16 bit prefix) and 171.69.10 (a 24 bit prefix)
  - A packet sent to 171.69.10.5 matches both prefixes.
- We resolve this with a rule that is based on the principle of "*longest prefix match*"
  - i.e., use the longest prefix sequence that matches

    171.69.10.5 = 10101011 01000101 00001010 00000101
  - 171.69.10 has a *longer* prefix match for 171.69.10.5 than 171.69, thus the packet is routed to 171.69.10

$$171.69.00.0 = \overbrace{10101011\ 01000101}^{\text{shorter prefix}}\ 00000000\ 00000000$$

$$171.69.10.0 = \underbrace{10101011\ 01000101\ 00001010}_{\text{longer prefix}}\ 00000000$$

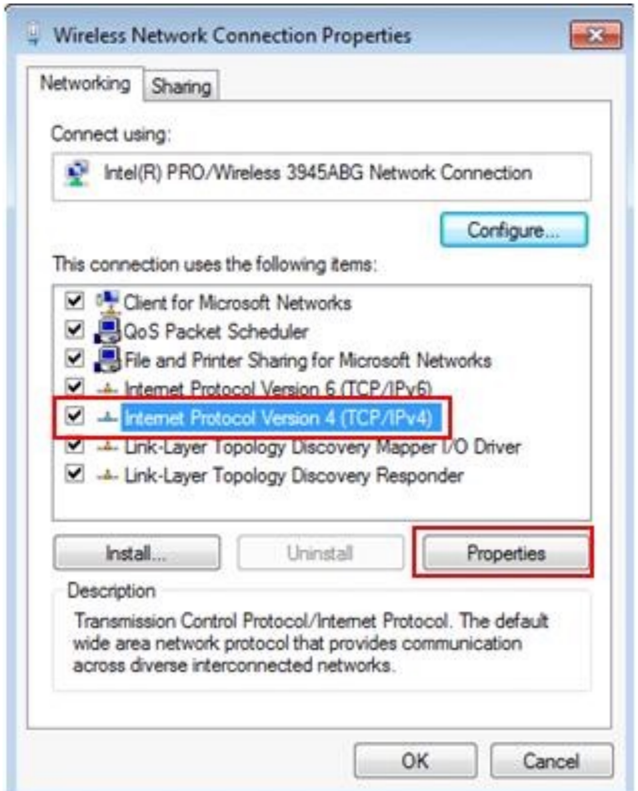
How are IP Addresses Assigned?

- There are two ways for a host to be assigned an IP address in a network
  1. network administrators can assign IP addresses manually, entering the address into the host’s network configuration files
  2. an automated system can have a pool of available addresses and assign one to a host on request
     - this requires a standard protocol so that all hosts “know” how to request an address and how to process the reply

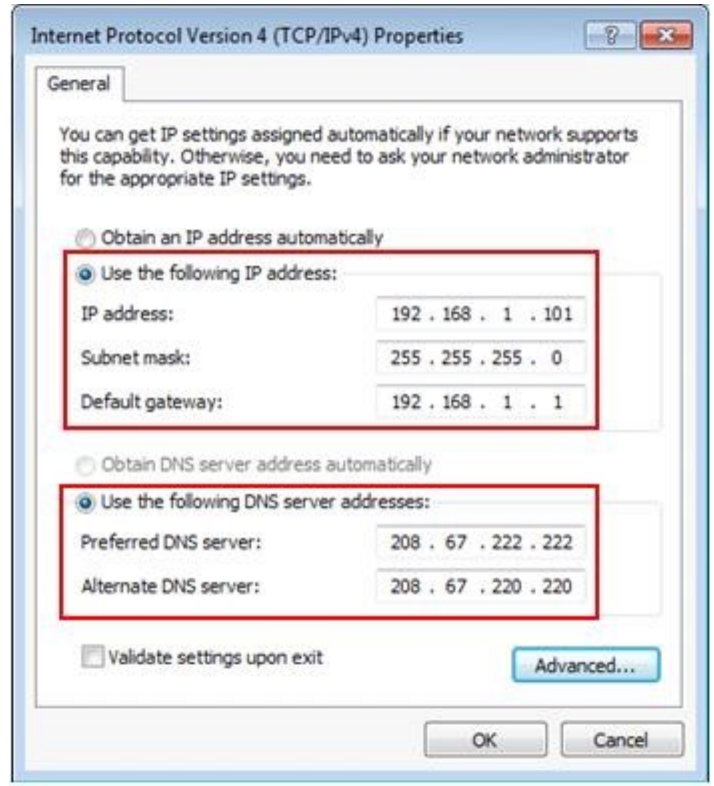
IP Configuration For a Host

- Operating Systems provide a way to manually configure the IP address for the host
  - modern systems will provide both a GUI configuration tool and a command-line method
  - command-line configuration may require editing configuration files or using specialized software
  - the following slides show examples from Windows and Linux
- Drawbacks of manual configuration:
  - large networks can take considerable time to configure and address conflicts can occur

Windows GUI Network Configuration

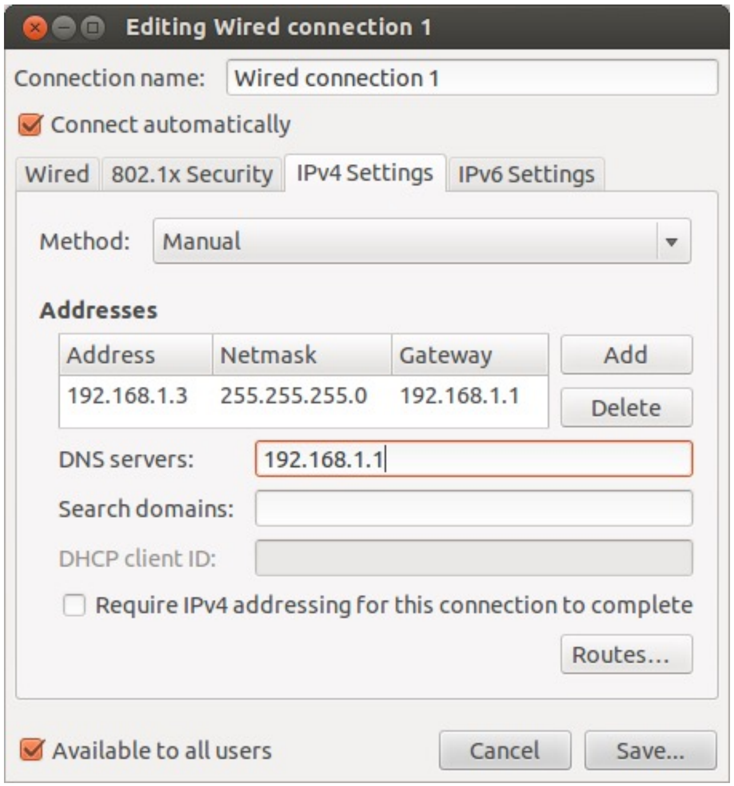


Select the protocol (IPv4 or IPv6)

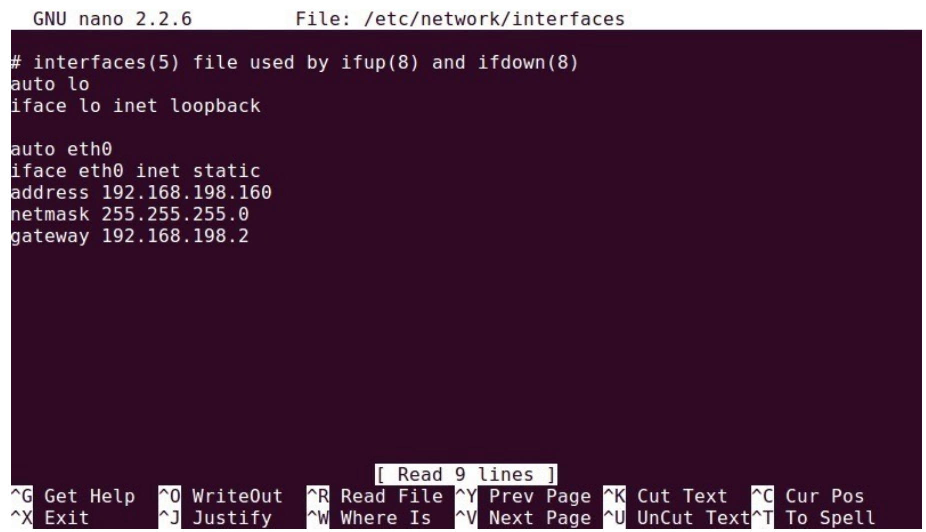


Enter the IP address (and other info)

Linux Network Configuration



GUI tool in Ubuntu Desktop



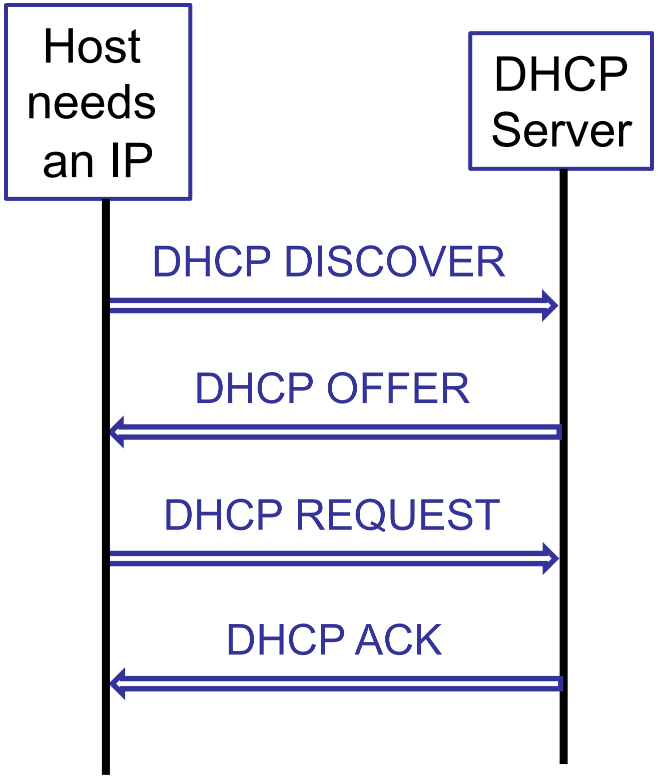
Manually editing the network configuration files in Ubuntu 16

Dynamic Host Configuration Protocol (DHCP)

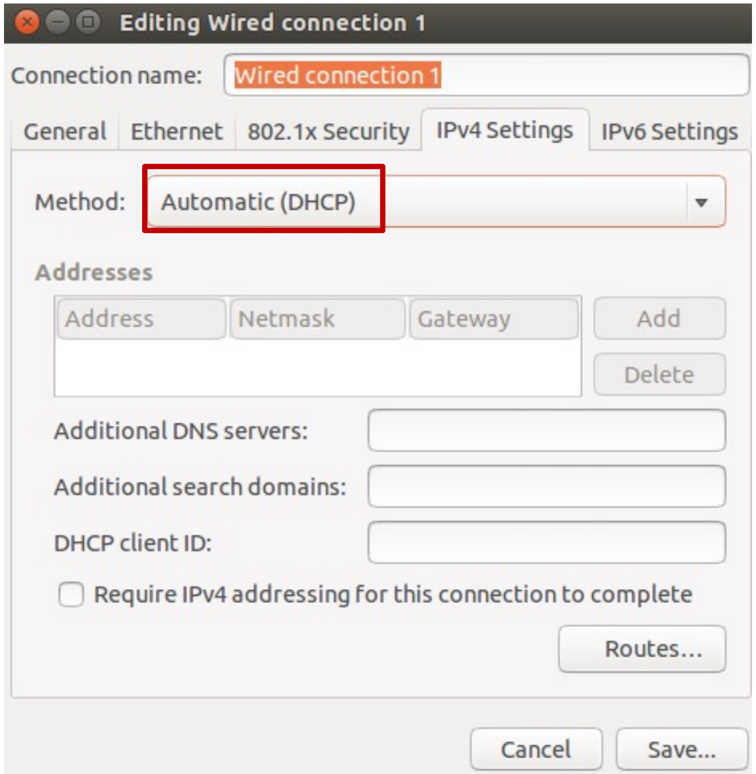
- A DHCP server is responsible for automatically providing configuration information to hosts
- There is normally at least one DHCP server for an administrative domain (IP network)
- The DHCP server maintains a pool of available IP addresses
  - When a host requests an address, one is taken from the pool, reserved for that host for a period of time and then released back into the pool when no longer needed, a timeout period recovers unused addresses

DHCP

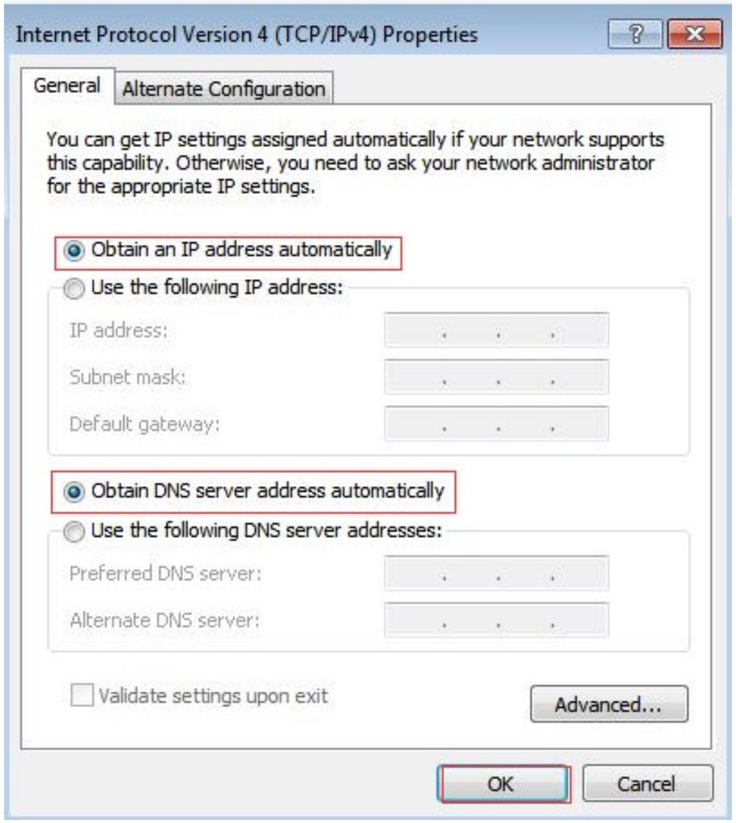
- A host broadcasts a *DHCP DISCOVER* message using the broadcast IP address (255.255.255.255) to find a DHCP server
- A DHCP server replies that is willing to deliver an IP address with a *DHCP OFFER*
- The host then sends a *DHCP REQUEST* to get an IP address
  - if more than one offer arrives, the host will only send a request to one of them
- The server sends the IP address to the host in a *DHCP ACK* and records the IP to MAC mapping along with an expiration time for the address



DHCP Configuration



Requesting DHCP in Ubuntu Desktop



Let DHCP get an IP address in Windows

Which Address Does a Host Use, IP or MAC?

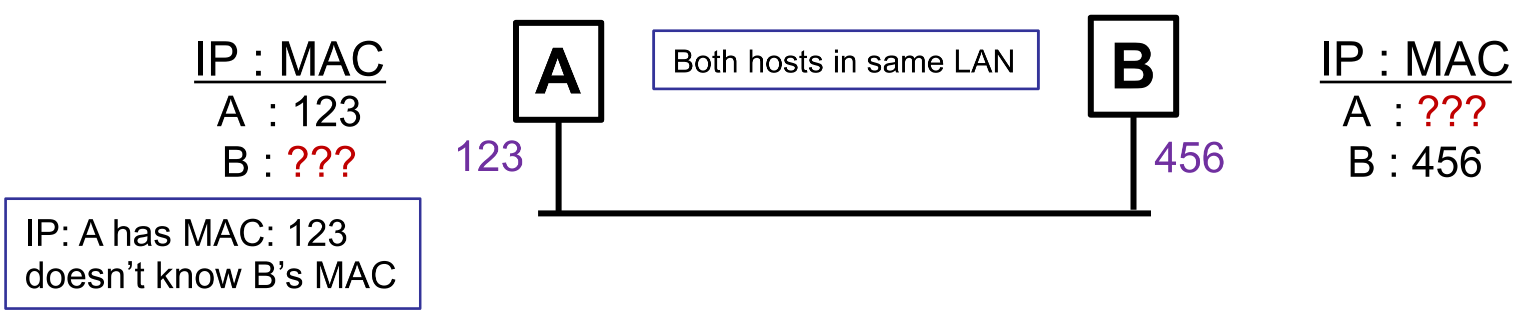
- If a host needs to get an IP address, it sends its MAC address to a DHCP server so it can get an IP address assigned to it
  - since it doesn't have an IP address yet, it can only use its MAC address in the DHCP request frame
- If a host is sending a packet to another host in the same LAN, it can use the MAC address, but it may not know that the destination is in the LAN
  - If the sender only has the IP address of the destination, how does it get the MAC address?

How Do We Map IP Addresses to MAC Addresses and Vice Versa?

- We use features from the Address Resolution Protocols, which work at the Data Link Layer
  - ARP (Address Resolution Protocol) requests the MAC (Ethernet) address for a specific IP address
  - RARP (Reverse ARP) requests the IP address for a specific MAC address
    - RARP is often replaced by DHCP or other protocols
  - Proxy ARP causes a LAN switch or bridge to pass an ARP request from one network segment to another
    - IP routers don't pass MAC broadcasts from one LAN segment to another, so the request has to be relayed by the Layer 2 device (LAN switch or bridge) between them

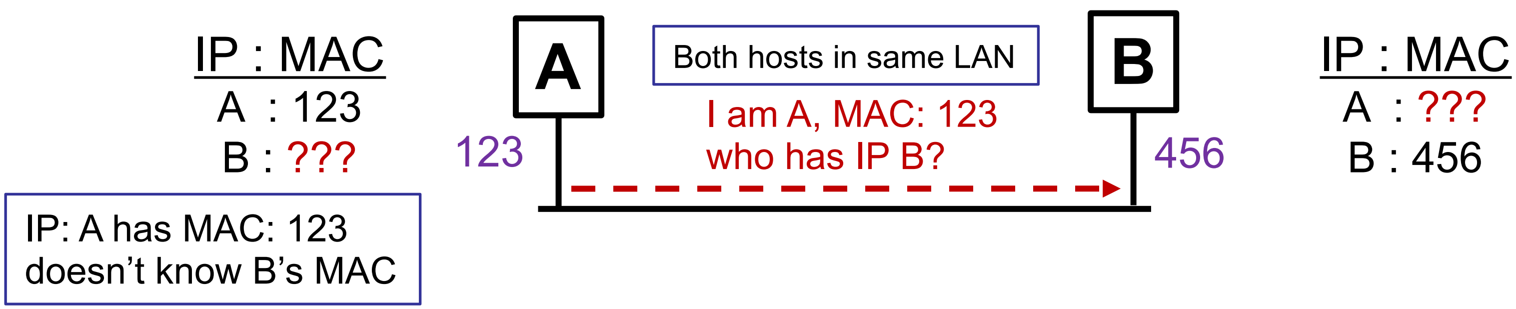
How Do We Map IP Addresses to MAC Addresses?

- We use the Address Resolution Protocol (ARP)
  - nodes have a temporary table of IP to MAC mappings
  - ARP can be used when a new node is added to a LAN or two nodes haven't communicated recently
  - assume A knows B's IP address, but needs B's MAC address to send frames directly to B



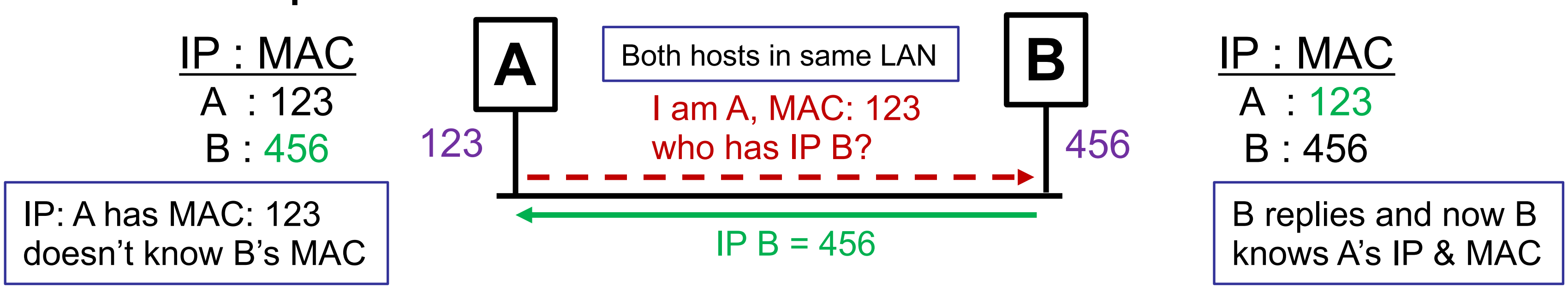
How Do We Map IP Addresses to MAC Addresses?

- We use the Address Resolution Protocol (ARP)
  - nodes have a temporary table of IP to MAC mappings
  - ARP can be used when a new node is added to a LAN or two nodes haven't communicated recently
  - assume A knows B's IP address, but needs B's MAC address to send frames directly to B
  - A broadcasts a request



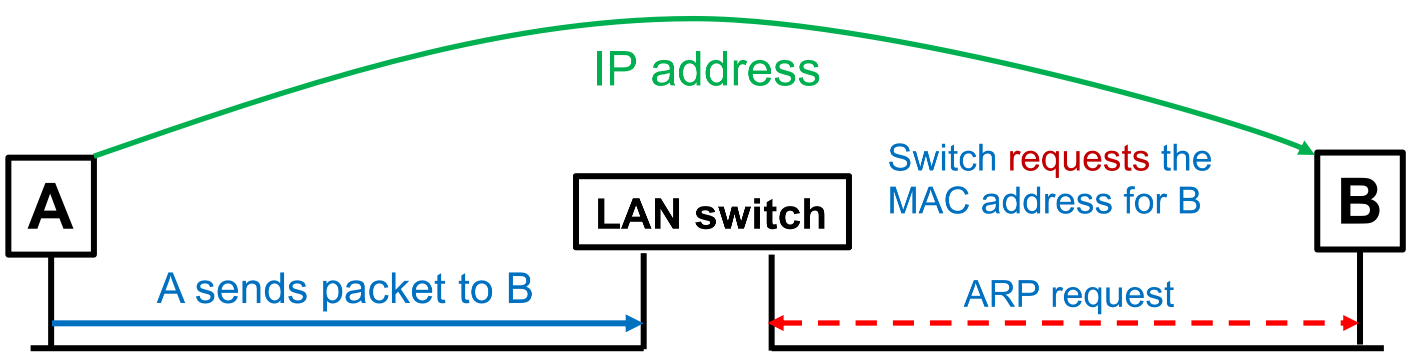
How Do We Map IP Addresses to MAC Addresses?

- We use the Address Resolution Protocol (ARP)
  - nodes have a temporary table of IP to MAC mappings
  - ARP can be used when a new node is added to a LAN or two nodes haven't communicated recently
  - assume A knows B's IP address, but needs B's MAC address to send frames directly to B
  - A broadcasts a request
  - B responds to A with its MAC address



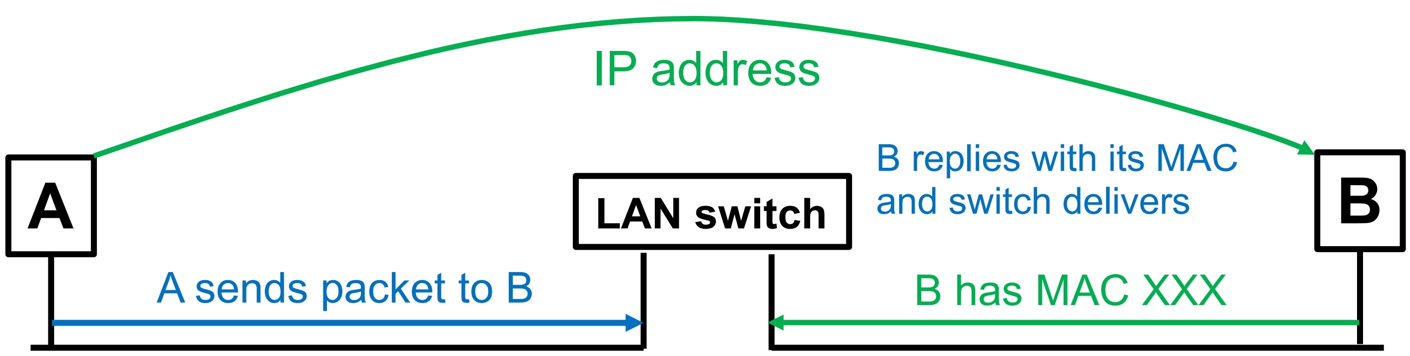
Proxy Address Resolution

- Sometimes we need to get the MAC address of a node in another segment of the same LAN
  - example: A wants to send a packet to B, but the LAN switch doesn't know B's MAC address
  - when A's packet arrives, the LAN switch sends an ARP request to get the MAC address for B



Proxy Address Resolution

- Sometimes we need to get the MAC address of a node in another segment of the same LAN
  - example: A wants to send a packet to B, but the LAN switch doesn't know B's MAC address
  - when A's packet arrives, the LAN switch sends an ARP request to get the MAC address for B
  - B replies with its MAC, the packet can be delivered



ARP Packet Format

<table>
<tr><td>0</td><td>8</td><td>16</td><td>31</td></tr>
<tr><td colspan="2">Hardware type = 1</td><td colspan="2">ProtocolType = 0x0800</td></tr>
<tr><td>HLen = 48</td><td>PLen = 32</td><td colspan="2">Operation</td></tr>
<tr><td colspan="4">SourceHardwareAddr (bytes 0–3)</td></tr>
<tr><td colspan="2">SourceHardwareAddr (bytes 4–5)</td><td colspan="2">SourceProtocolAddr (bytes 0–1)</td></tr>
<tr><td colspan="2">SourceProtocolAddr (bytes 2–3)</td><td colspan="2">TargetHardwareAddr (bytes 0–1)</td></tr>
<tr><td colspan="4">TargetHardwareAddr (bytes 2–5)</td></tr>
<tr><td colspan="4">TargetProtocolAddr (bytes 0–3)</td></tr>
</table>

- When an ARP request goes out, the IP (Protocol) address is present, the MAC (Hardware) address is the broadcast value
- The ARP reply replaces the broadcast MAC address with the MAC address of the node that matches the IP that was requested

HardwareType: type of physical network (e.g., Ethernet)

ProtocolType: type of higher layer protocol (e.g., IP)

HLEN & PLEN: length of physical & protocol addresses

Operation: request or response

Source/Target Physical (MAC) / Protocol (IP) addresses

Internet Control Message Protocol (ICMP)

- A network-layer protocol that provides a variety of error messages that are sent back to the source host whenever a router or host is unable to process an IP datagram successfully
  - Destination host unreachable due to link /node failure
  - Fragment reassembly process failed
  - TTL had reached 0 (so datagrams don't cycle forever)
  - IP header checksum failed
- ICMP-Redirect
  - Sent from router to a source host
  - Updates routing information

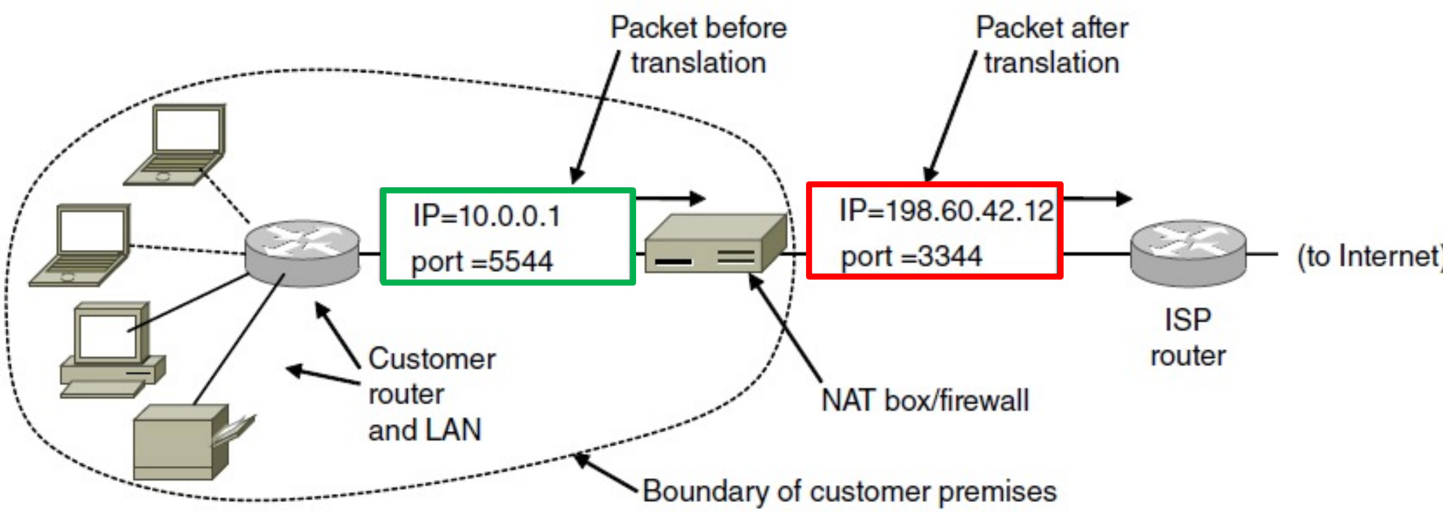
ICMP Header and Codes

Byte 0	Byte 1	Byte 2	Byte 3
Type	*Code*	*Checksum*	
Rest of Header (varies with type)			

Type	Code	Description
0	0	echo reply (ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	3	destination port unreachable
8	0	echo request (ping)
10	0	route discovery
11	0	TTL expired

Network Address Translation

- Network address translation (NAT)
  - Routers translate internal LAN-specific addresses into a globally routable IP address
  - Helps deal with the limited IPv4 address range
    - Most home networks use NAT to share one external IP
  - Security: prevents unsolicited direct inbound access



Private IP Addresses

- Internet *RFC 6761* describes special-use IP addresses, commonly used by NAT
  - Home routers typically use the `172.16.X.X` range or the `192.168.X.X` range

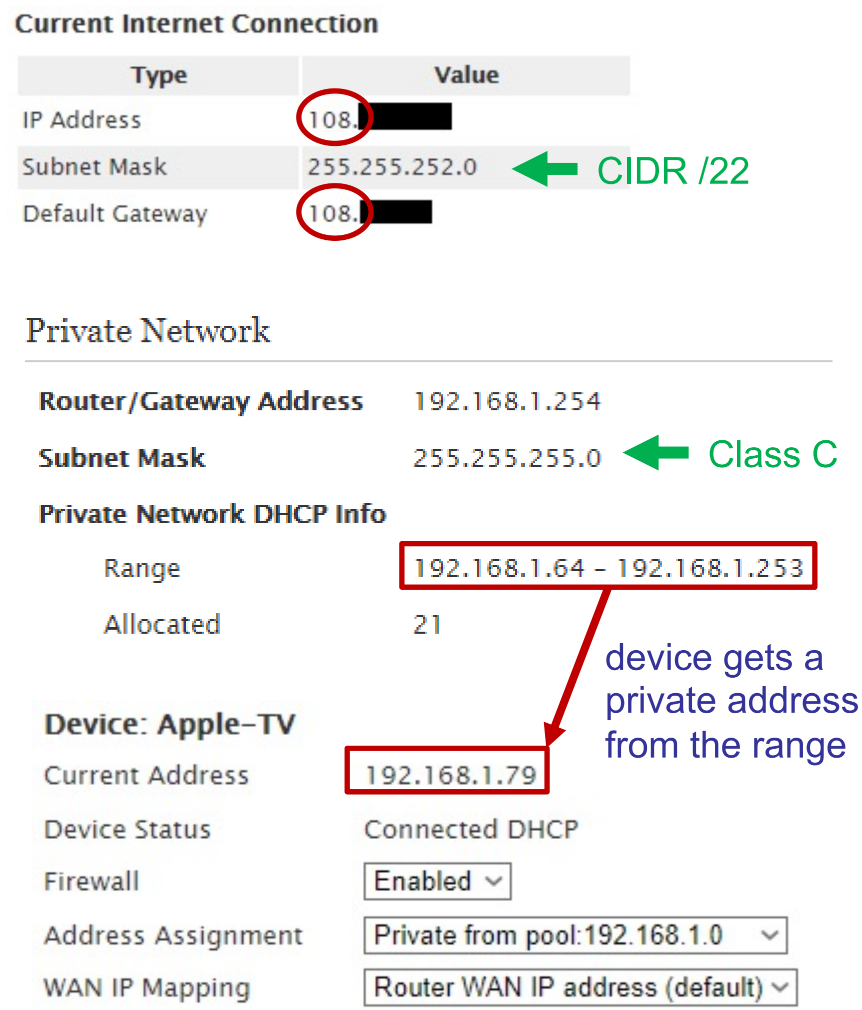
10.0.XX	172.21.X.X	172.27.X.X
172.16.X.X	172.22.X.X	172.28.X.X
172.17.X.X	172.23.X.X	172.29.X.X
172.18.X.X	172.24.X.X	172.30.X.X
172.19.X.X	172.25.X.X	172.30.X.X
172.20.X.X	172.26.X.X	192.168.X.X

NAT types

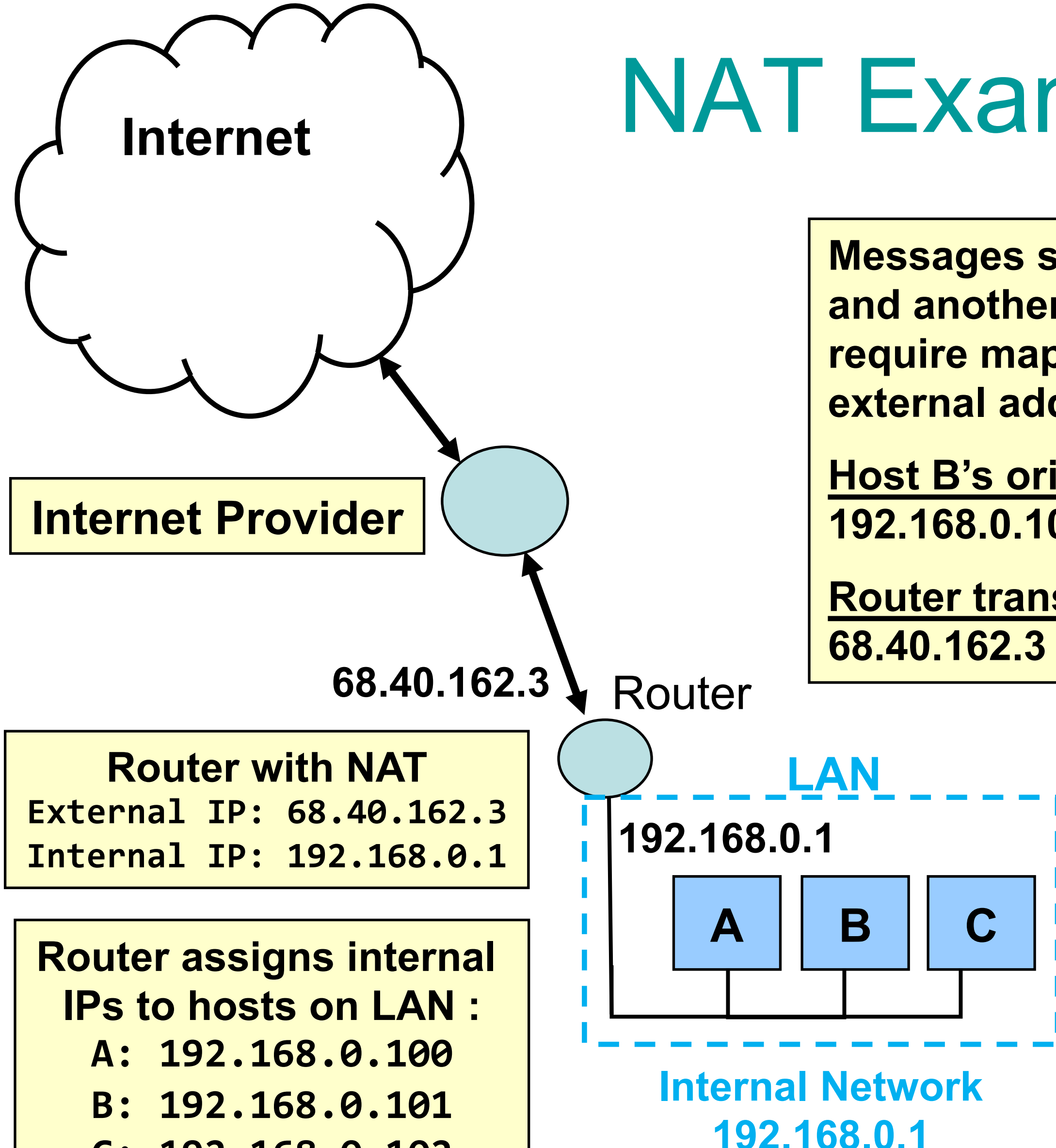
- Traditional NAT
  - only hosts that need direct Internet access get an IP address that is not in the private range
  - all others use the private range inside the LAN
  - this does limit the number of hosts on the Internet
- Address-and-port translation (NAPT)
  - most common form today, still referred to as NAT
  - one external (global) IP address, many internal
  - uses IP+port numbers to map to internal hosts

NAT Example

- Router's external address is a normal Internet address
- Router allocates a range of private IP addresses for hosts in the local network
- Devices are allocated addresses as needed



NAT Example



Messages sent between host B and another host on the Internet require mapping internal and external addresses and ports

<u>Host B's original socket:</u>
192.168.0.101 port 1341

<u>Router translates Host B's socket:</u>
68.40.162.3 port 5280

Router with NAT
`External IP: 68.40.162.3`
`Internal IP: 192.168.0.1`

Router assigns internal IPs to hosts on LAN :
```
A: 192.168.0.100
B: 192.168.0.101
C: 192.168.0.102
```

Messages between hosts inside the LAN use the internal IP addresses