CSI3660 – System Administration

Prof. Fredericks

Network Configuration

Outline

- Network configuration
  - Networks
  - Protocols
  - Access methods

  - TCP/IP configuration
  - NIC configuration

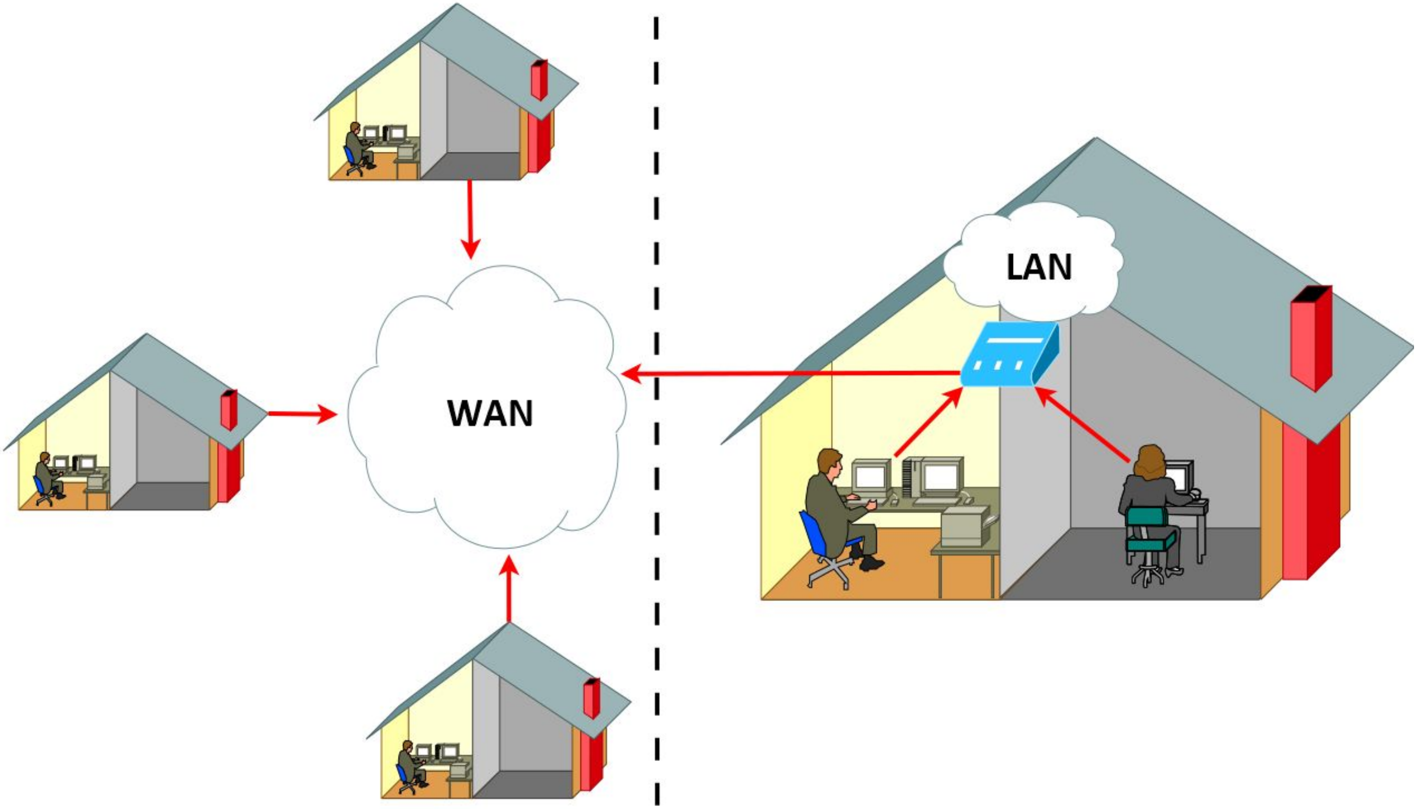
Objectives

- Describe the purpose of host names and how they are resolved to IP addresses
- Configure TCP/IP routing
- Identify common network services
- Use command-line utilities to perform remote administration

Networks

- Network
  - Two or more computers joined via media and able to exchange information
- Local area networks (LANs)
  - Networks that connect computers within close proximity
  - E.g., used to allow connection to shared resources
- Wide area networks (WANs)
  - Networks that connect computers separated by large distances
  - e.g., used to connect to Internet Service Provider
- Internet service provider (ISP)
  - Company providing Internet access

Networks

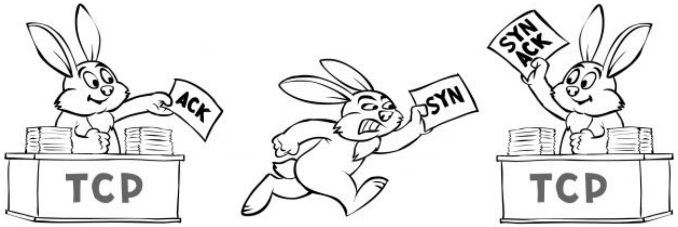


Networks

- Routers
  - Special computers/appliances capable of transferring information between networks
- Protocol
  - Set of rules for communication between networked computers
- Packets
  - Data messages formatted by a network protocol
  - Packets can be recognized by routers and other network devices

Networks

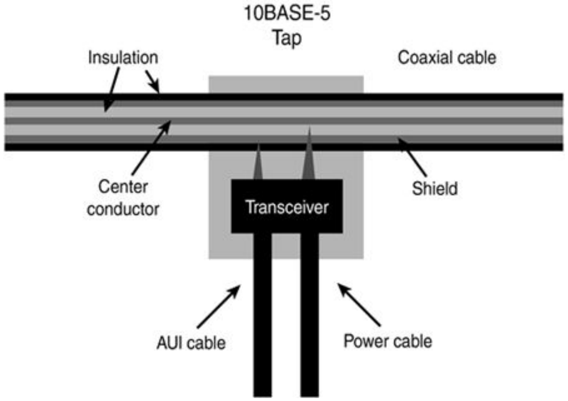
- Common network protocols:
  - TCP/IP (Transfer Control Protocol/Internet Protocol)
  - UDP/IP (User Datagram Protocol/Internet Protocol)
  - IPX/SPX (Internetwork Packet Exchange/Sequence Packet Exchange)
  - AppleTalk

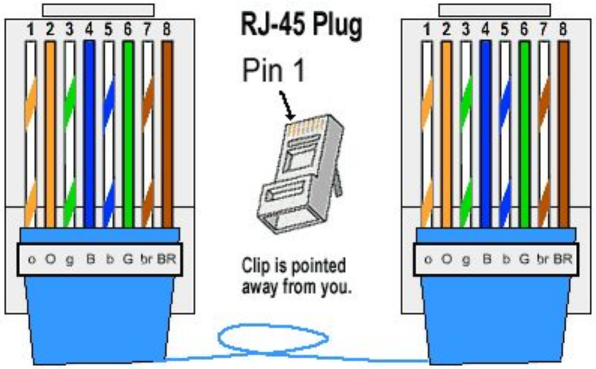


Networks

- Media access method
  - Set of rules that govern how devices on a network share the network media
  - Contained within the hardware on NIC or modem
- Ethernet
  - Most common network media access method
  - Ensures that packets are retransmitted onto the network if a network error occurs
- Token ring
  - Media access method
  - Controls which computer has the ability to transmit information by using a token

- Early networking: vampire taps



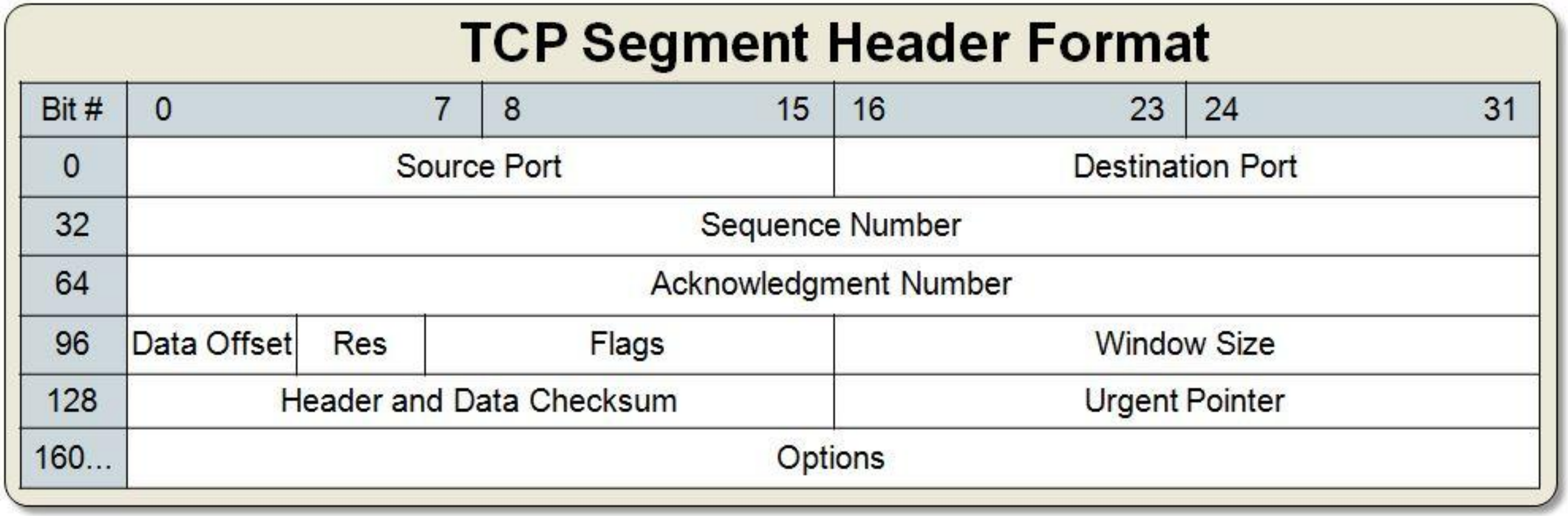


- Current: Ethernet (RJ45)

The TCP/IP Protocol

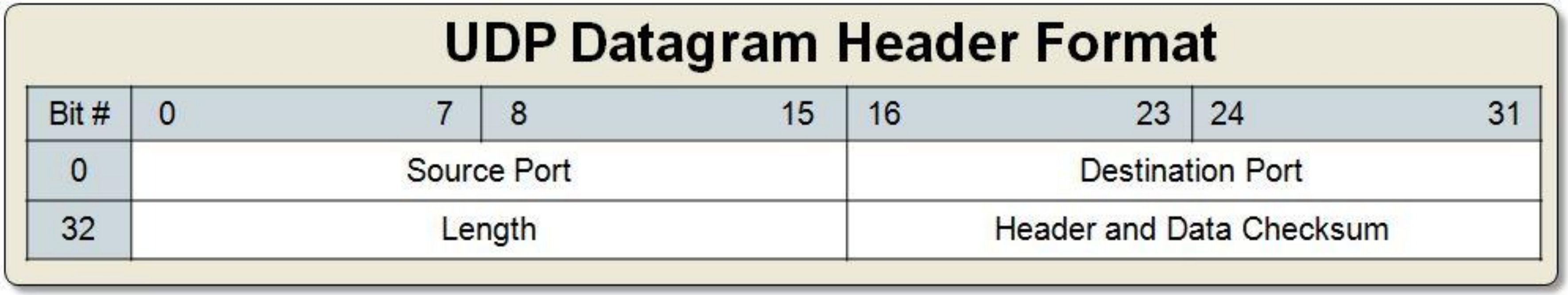
- Set of protocols with two core components
  - **TCP**: ensures that packets are assembled in the correct order, regardless of arrival order
  - **IP**: responsible for labeling each packet with destination address
- Together, TCP and IP ensure that information packets travel across the network as quickly as possible without getting lost

TCP/IP vs UDP



TCP Segment Header Format

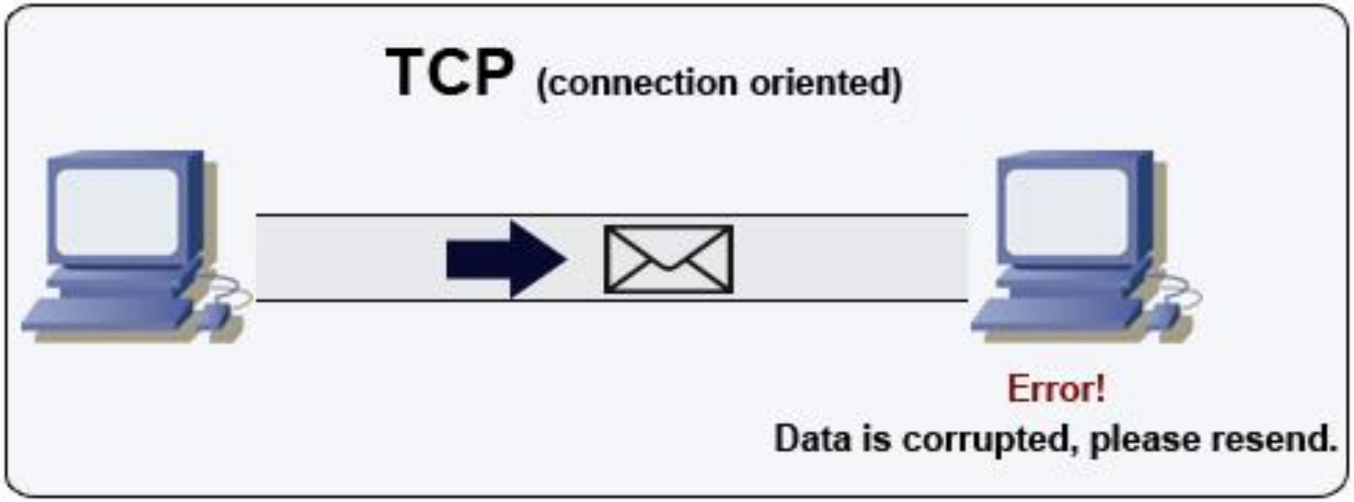
<table>
<tr><th>Bit #</th><th colspan="3">0 7</th><th>8 15</th><th>16 23</th><th>24 31</th></tr>
<tr><td>0</td><td colspan="4">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td>32</td><td colspan="6">Sequence Number</td></tr>
<tr><td>64</td><td colspan="6">Acknowledgment Number</td></tr>
<tr><td>96</td><td>Data Offset</td><td>Res</td><td colspan="2">Flags</td><td colspan="2">Window Size</td></tr>
<tr><td>128</td><td colspan="4">Header and Data Checksum</td><td colspan="2">Urgent Pointer</td></tr>
<tr><td>160...</td><td colspan="6">Options</td></tr>
</table>



UDP Datagram Header Format

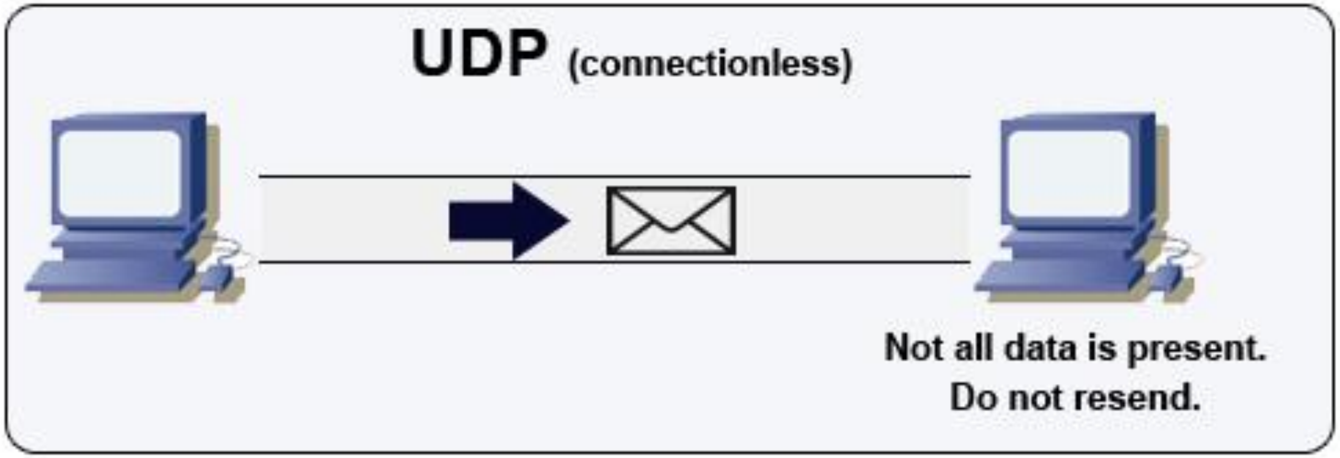
Bit #	0 7	8 15	16 23	24 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

TCP/IP vs UDP



TCP (connection oriented)

Error!
Data is corrupted, please resend.



UDP (connectionless)

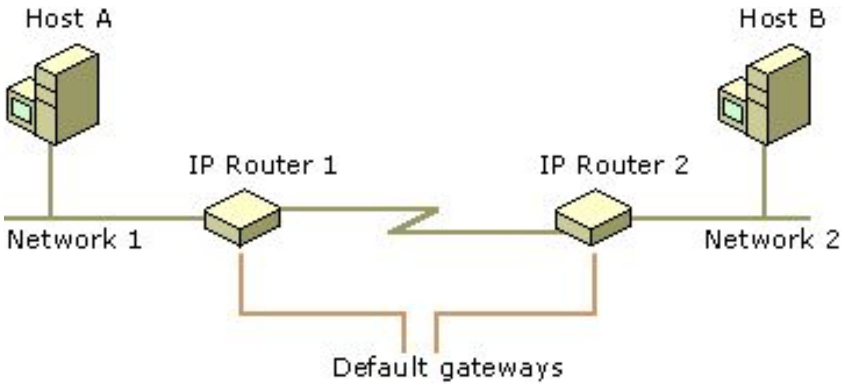
Not all data is present.
Do not resend.

The TCP/IP Protocol

- Each computer on a TCP/IP network must have a valid Internet Protocol (IP) address
  - Identifies itself to the IP protocol
- IP version 4 still very common (IPv4)
- Next-generation protocol IP version 6 (IPv6) available
  - Anybody know one big difference between 4 and 6?

The IPv4 Protocol

- To participate on an IPv4 network, your computer must have a valid IP address
  - As well as a subnet mask
  - To participate on a larger network (Internet) you can configure a default gateway



IPv4 Addresses

- IP address: unique number that identifies a networked computer
  - Octet: series of four 8-bit binary numbers
    - Common format of IPv4 addresses
    - 192.168.5.69 is an example

- Unicast
  - Directed TCP/IP communication from one computer to another single computer

IPv4 Addresses

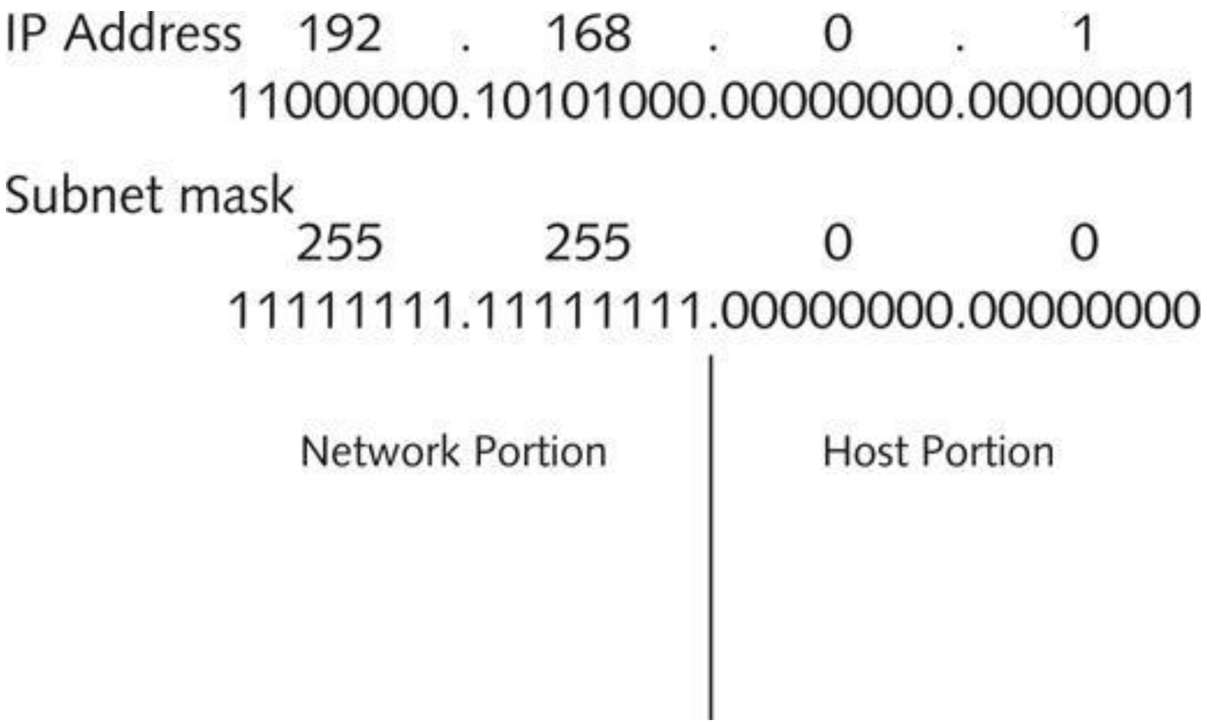
- IPv4 addresses are composed of two parts
  - Network ID
    - Network on which a computer is located
  - Host ID
    - Single computer on that network
    - Two computers with different network IDs can have the same host ID
- Only computers with the same network ID can communicate without a router
  - Allows administrators to logically separate computers on a network

Subnet Masks

- Define which part of an IP address is the network ID and which part is the host ID
  - Series of four octets
  - Octet in subnet mask containing 255 is part of network ID
  - Octet in subnet mask containing 0 is part of host ID

- ANDing: calculate network and host IDs from an IP address and subnet mask
  - Compare binary digits and gives a result of 1 or 0

Subnet Masks



IP Address 192 . 168 . 0 . 1

11000000.10101000.00000000.00000001

Subnet mask

255 255 0 0

11111111.11111111.00000000.00000000

Network Portion

Host Portion

Figure 12-1: A sample IP address and subnet mask

IPv4 and Subnetting

```
192.168.123.132 - IP address
255.255.255.0 - subnet mask

11000000.10101000.01111011.10000100
         -- IP address (192.168.123.132)
11111111.11111111.11111111.00000000
         -- Subnet mask (255.255.255.0)

11000000.10101000.01111011.00000000
         -- Network address (192.168.123.0)
00000000.00000000.00000000.10000100
         -- Host address (000.000.000.132)
```

Subnet Masks

- IP addresses that cannot be assigned to a host computer:
  - 0.0.0.0 = all networks (sometimes localhost)
  - 255.255.255.255 = all computers on all networks
- 255 in an IP address can specify many hosts
  - Broadcast addresses
- Example: 192.168.255.255 refers to all hosts on the 192.168.0.0 network
- 127.0.0.1
  - Loopback – computer can talk to itself

Private Addresses

- Cannot be assigned to hosts on Internet
  - Available to be used on internal networks (behind router)

- All IPs in 10.0.0.0 network
- 172.16 – 172.31 networks
- 192.168 network

Subnetting

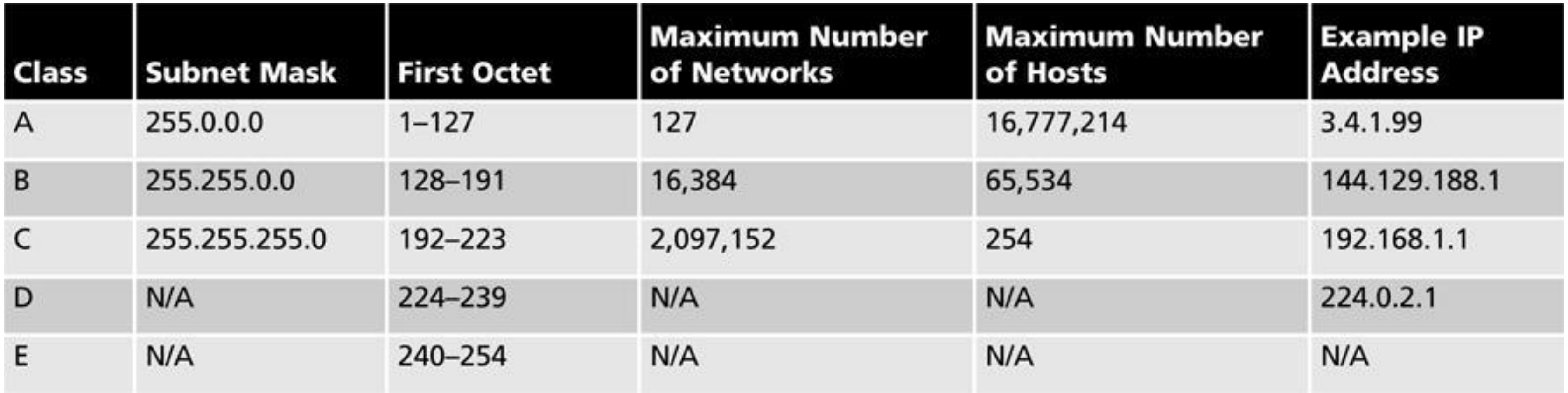
- Why subnet?
  - Ethernet can’t have more than 1024 hosts on single domain
  - Performance issues – switches can handle only so many

  - Departments need to compartmentalize their hosts
    - HR may need secure hosts but developers don’t
    - Allocate Class C networks to different departments
      - 10.0.0.0 network
        - 10.1.1.0 – HR
        - 10.1.2.0 – Engineering
        - 10.1.3.0 – Upper management
        - etc.

IPv4 Classes and Subnetting

- IPv4 address class defines default subnet mask of associated device
  - All IP address classes can be identified by first octet
  - Class A
    - **8** bits for network ID, **24** bits for host ID
    - Assigned to very large companies
  - Class B
    - **16** bits for network ID, **16** bits for host ID
    - Assigned to larger organizations with several thousand users
  - Class C
    - **24** bits for network ID, **8** bits for host ID
    - Used for small and home networks

IPv4 Classes and Subnetting



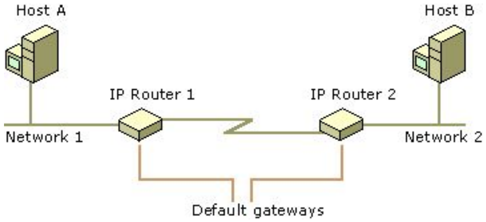
Class	Subnet Mask	First Octet	Maximum Number of Networks	Maximum Number of Hosts	Example IP Address
A	255.0.0.0	1–127	127	16,777,214	3.4.1.99
B	255.255.0.0	128–191	16,384	65,534	144.129.188.1
C	255.255.255.0	192–223	2,097,152	254	192.168.1.1
D	N/A	224–239	N/A	N/A	224.0.2.1
E	N/A	240–254	N/A	N/A	N/A

Table 12-1: IP address classes

IPv4 Classes and Subnetting

- Multicast: TCP/IP communication destined for a certain group of computers
  - Class D addresses
    - 224.0.0.0 – 239.255.255.255
  - Relatively new … older computers may not support multicast

- Subnetting
  - Process of dividing a large network into smaller networks
  - Used to control traffic flow
  - Take bits from host ID; give them to network ID



Default Gateway

- IP address of network interface on a router
  - Send packets destined for a different network
  - May not necessarily know destination

- Routers can distinguish between different networks
  - Move packets between
  - Have assigned IP addresses on each attached network interface

- Determine if sending packets to local host or remote host

Troubleshooting

- Issues typically caused by one of these main topics:
- **Incorrect subnet mask**
  - Non-default subnet mask used, but client uses default mask
  - Communication will fail to nearby networks but not remote
- **Incorrect IP address**
  - Computers assigned IP addresses that should be assigned for different subnets
  - Issues communicating with nearby networks using same address
- **Incorrect default gateway**
  - Wrong router configured as default gateway
  - E.g., 1 router for internet, 1 for internal network
  - Communication issues

The IPv6 Protocol

- Number of IP addresses using IPv4 is unsuitable for Internet growth
- IPv6 protocol: uses 128 bits to identify computers
- IPv6 IP addresses are written using 8 colon-delimited 16-bit hexadecimal numbers
  - 2001:0db8:3c4d:0015:0000:0000:adb6:ef12
  - 0000 can be omitted in most notation
  - Above address could also be written:
    - 2001:0db8:3c4d:0015:::adb6:ef12

The IPv6 Protocol

- IPv6 address contains two portions
  - First half assigned by ISP and identifies network
  - Last half is link local portion: used to uniquely identify computers in a LAN

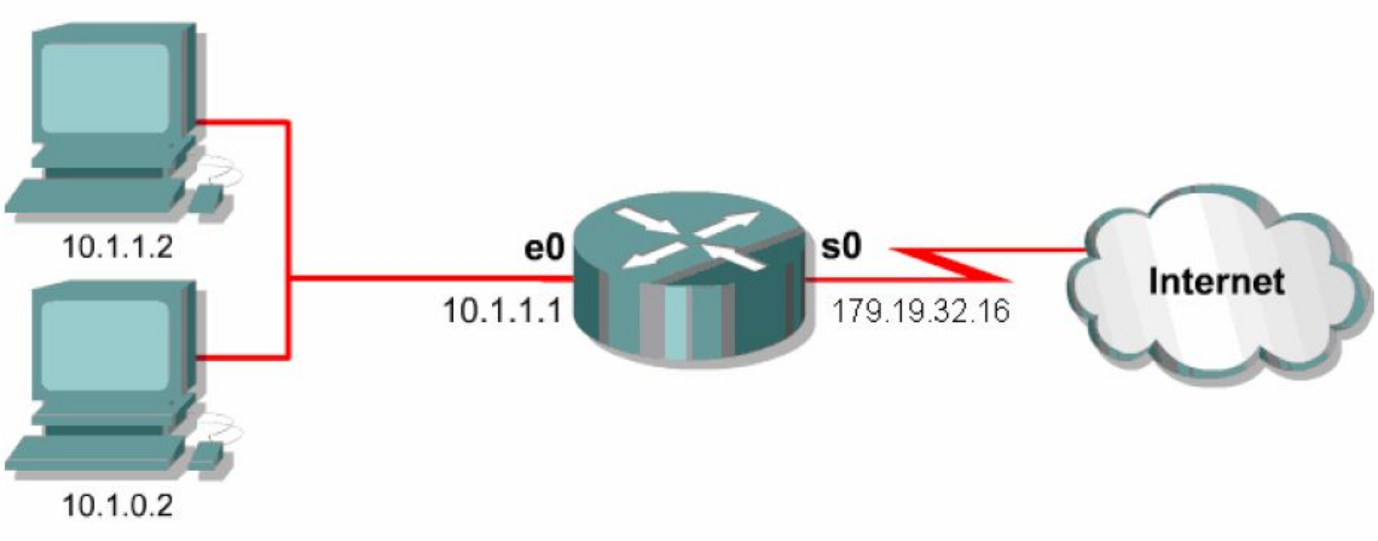
- Most operating systems today support IPv6
  - Few networks and computers on the Internet have adopted IPv6

The IPv6 Protocol

- Proxy servers and NAT routers
  - Computers or hardware devices that have an IP address and access to a network
    - Translate one IP address to another
  - Used by other computers to obtain network resources on their behalf
  - Allows computers behind different NAT routers or proxy servers to have the same IPv4 address
- Due to usage of proxy servers and NAT
  - Available IPv4 addresses has remained high and slowed the adoption of IPv6

The IPv6 Protocol

- Proxy servers and NAT routers



ress and

s on their behalf

xy servers to

- Due to usage of proxy servers and NAT
  - Available IPv4 addresses has remained high and slowed the adoption of IPv6

tcpdump

- Reading/writing traffic dumpfiles

  -
    ```
    [user]# tcpdump -w /path/to/dumpfile -i ens192 (write)
    ```

  -
    ```
    [user]# tcpdump -r /path/to/dumpfile -n icmp (read)
    ```

    -
      ```
      -vvv for more verbosity
      ```

- Captures first 65,535 bytes of packet
- If you need more or less, you can change this value

  -
    ```
    [user]$ tcpdump -w /path/to/dumpfile -i ens192 -s 1500
        (full 1500 byte packet)
    ```

tcpdump

- Packet tracing can impact performance
  - Minimize with good filter
  - -w option writes raw packet to disk for later decoding

- Don’t capture your own traffic
  - Login to network then tcpdump
  - Capture your own session packets
    - Printing to screen…can generate new packets…which then get captured again…which then get printed to screen….

  - Skip port 22 (SSH)
    - `[user]$ tcpdump not tcp port 22`
  - Watch 22 but skip your own IP address
    - `[user]$ tcpdump “not (host 192.168.1.8 and tcp port 22)”`

tshark (Wireshark)

- Getting data with tshark (needs Wireshark)

```
$ sudo yum install wireshark

$ sudo tshark -i eth0 -w dumpfile.pcap  ### doesn’t work in Cent

$ sudo tshark -i eth0 -w - > dumpfile.pcap

      https://bugzilla.redhat.com/show_bug.cgi?id=850768

$ sudo chown <user> dumpfile.pcap

   .. get rid of those pesky permissions issues

$ tshark -r dumpfile.pcap

   .. summarize

$ tshark -nr dumpfile.pcap -z conv,ip
```

tshark

.. prettier summary

```
$ tshark -q -nr dumpfile.pcap -t ad -z
io,stat,1,”AVG(frame.len)frame.len”
```

what fields can we use?

```
$ tshark -G fields
```

Graphing Throughput

Graphing (the CLI way) (FIX LATER)

- Capture SYN/ACK packets sent from webserver (synchronize/acknowledge)
  - Plotting ISN (initial sequence number) – helps prevent spoofing
- Pipe to Perl script

```
[user]$ tcpdump -i eth0 -l -c 5 -n -t “tcp[13] ==
18” | perl -ane ‘($s,$j)=split(/,/,$F[7]); print
“$s\n”;’ > graphme
```

- Output file (graphme) is string of numbers

Graphing (the CLI way)

- Use **gnuplot** to graph

```
[user]$ gnuplot

gnuplot>set terminal png

gnuplot> set output ‘syns.png’

gnuplot> plot ‘graphme’

gnuplot> quit
```

Day 2

Reddit> My college classes are like a high-level Dora the Explorer episode. Person up front asks a question, stares at you blankly for a few seconds, and then answers their own question.

Favorite comment:

My Intro to System Administration professor asked us a question on the first or second day and nobody said anything for a while. Then he said, "Don't worry, I'm really good at this 'awkward silence' thing." And proceeded to just stare back at us until someone guessed. He does that every class.

Configuring a Network Interface

- If a NIC was detected during installation
  - Linux automatically configures appropriate driver
- `insmod` and `modprobe` commands
  - Load kernel objects into the Linux kernel
  - Can be used to load NIC drivers
- `lsmod`
  - Displays a list of currently loaded modules
- `rmmod`
  - Removes module from kernel
- Older Linux kernels loaded from entries within the /etc/modprobe.conf or /etc/modules.conf file

Network Drivers

- All network drivers found here:

```
[user]$ cd /lib/modules/`uname -r`/kernel/drivers/net

[user]$ ls



[user]$ modinfo tulip | grep -i description
```

Create a Dummy NIC

- Don’t want to affect eth0 while messing around…

```
$ sudo lsmod | grep dummy
$ sudo modprobe dummy
$ sudo lsmod | grep dummy

$ sudo ip link set name eth10 dev dummy0

$ ip link show eth10

Remove later:
$ sudo ip link delete eth10 type dummy
$ sudo rmmod dummy
```

Configuring a Network Interface

- `ifconfig`
  - Used to configure TCP/IP on a NIC
  - Also used without any arguments to view configuration of all network interfaces in computer
  - `[user]$ ifconfig eth10 <new IP address>`
    - (Don't recommend changing your IP address on eth0 through SSH – all kinds of things go wonky)

- `dhclient` command
  - Receive TCP/IP configuration from DHCP or Boot Protocol (BOOTP) server

Configuring a Network Interface

- If your network has IPv6-configured routers
  - An IPv6 address is automatically assigned to each NIC

- NICs use Internet Control Message Protocol version 6 (ICMPv6) router discovery messages to probe the network for IPv6 configuration information

Configuring a Network Interface

- /etc/sysconfig/network-scripts/ifcfg-*interface* file
  - Stores NIC configurations
  - Allows the system to activate and configure TCP/IP information at each boot time

- `ifdown` command
  - Unconfigures a NIC

- `ifup` command
  - Configures NIC using /etc/sysconfig/network-scripts/ifcfg-*interface* file

- `ping` (Packet Internet Groper)
  - Check TCP/IP connectivity on a network

    `-c` option: limit the number of `ping` packets sent

More Configuration Options

- ifconfig <NIC> down
- ifconfig <NIC> up
  - Bring NIC up and down, respectively
  - Doesn’t look to /etc/sysconfig/network-scripts/…

Name Resolution

- Host name
  - User-friendly computer name
- Fully qualified domain name (FQDN)
  - Host name following the DNS (Domain Name Space) convention
- DNS
  - Hierarchical namespace for host names
- `whois` (need to yum install whois)
  - Used to obtain registration information about a domain within a name space
  - `whois google.com`
- `hostname`
  - View or set a computer's host name

Name Resolution

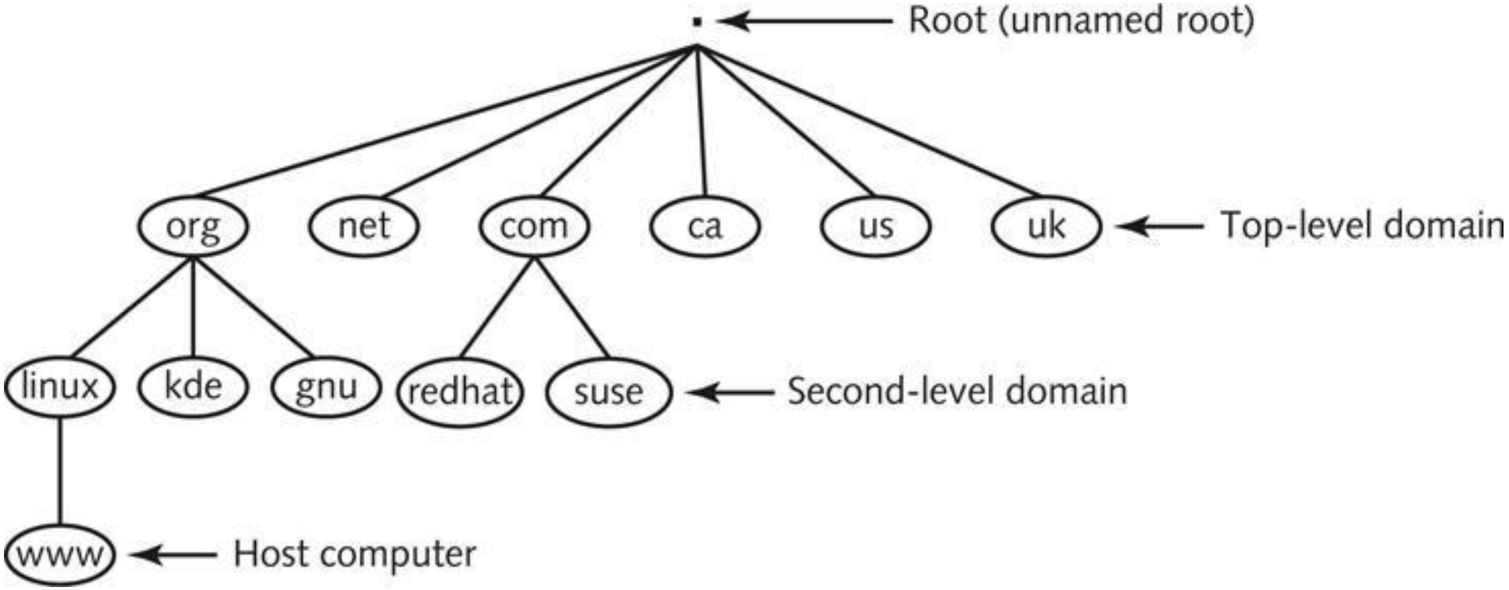


Figure 12-6: The domain name space

www.oakland.edu is a FQDN

Name Resolution

1) Find correct server
   - Somewhat difficult (distributed via hierarchy of servers)
     - Starts at root (dot) then goes downward
       - Servers responsible for one zone of that namespace

2) Map server name to IP address
   - Lowest level (or second lowest) will generally have the address

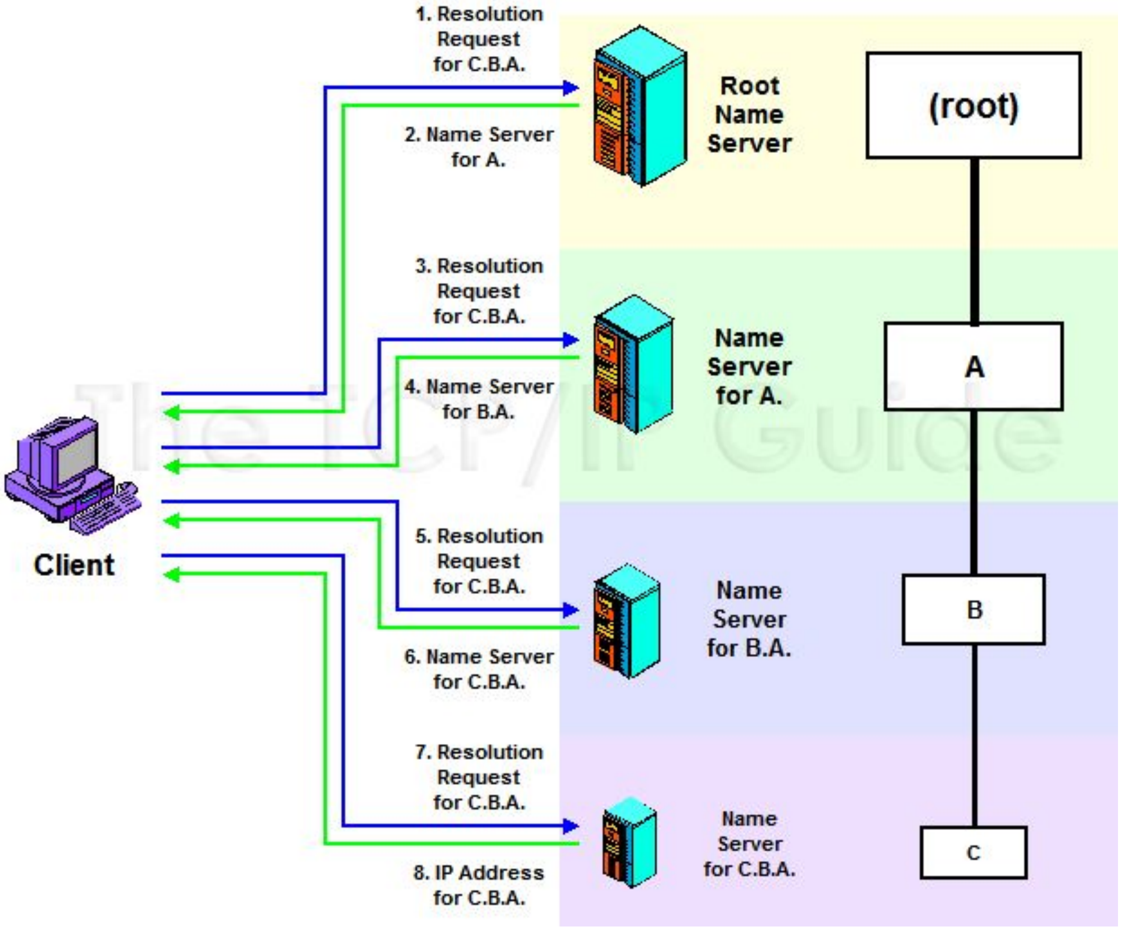
Name Resolution

- Assume our fully-qualified domain name is **www.oakland.edu**

- Top-level domain (.) will have the information to resolve **edu**
- **edu** domain will have the information to resolve **oakland**
- **oakland** will have the info to resolve **www**

- Possible that **edu** may know IP address already, but not necessarily likely
  - But, knows server for resolving **www.oakland**, so therefore edu is considered authoritative for **www.oakland**

Name Resolution (for c.b.a domain)



Name Resolution

- TCP/IP cannot identify computers via hostnames
  - Must map host names to IP addresses
    - Can be done by placing entries in the /etc/hosts file

```
$ lynx csci3660.com
```

- ISPs list FQDNs in DNS servers on Internet
  - Applications request IP addresses associated with a specific FQDN
  - Configure by specifying the IP address of the DNS server in /etc/resolv.conf file

Routing

- Route table
  - List of TCP/IP networks stored in system memory
- `route`
  - Displays the route table
- Multihomed hosts
  - Computers with multiple network interfaces
- IP forwarding
  - Forwarding packets from one interface to another
  - Also known as routing

Routing

- To enable routing:
  - Place number 1 in:
    - /proc/sys/net/ipv4/ip_forward for IPv4
    - /proc/sys/net/ipv6/conf/all/forwarding for IPv6

- To enable routing at every boot:
  - Edit the /etc/sysctl.conf file to include:
    - "net.ipv4.ip_forward = 1" for IPv4
    - "net.ipv6.conf.default.forwarding = 1" for IPv6

Why have a workstation setup as a router?

Routing

- Large networks may have several routers
  - Packet may travel through several routers
  - May require adding entries in the router table
  - Redirect packets to appropriate destination
- `route add <route>`
  - Add entries to route table
- `route del <route>`
  - Remove entries from route table
- `ip`
  - Can be used to manipulate the route table

Routing

- Set default gateway of 192.168.1.1

```
[user]$ route add -net default gw 192.168.1.1 dev
eth0

default: destination network

gw: specify as gateway
```

- Add route

```
[user]$ sudo ip route add 192.168.2.6 dev eth10
```

Routing

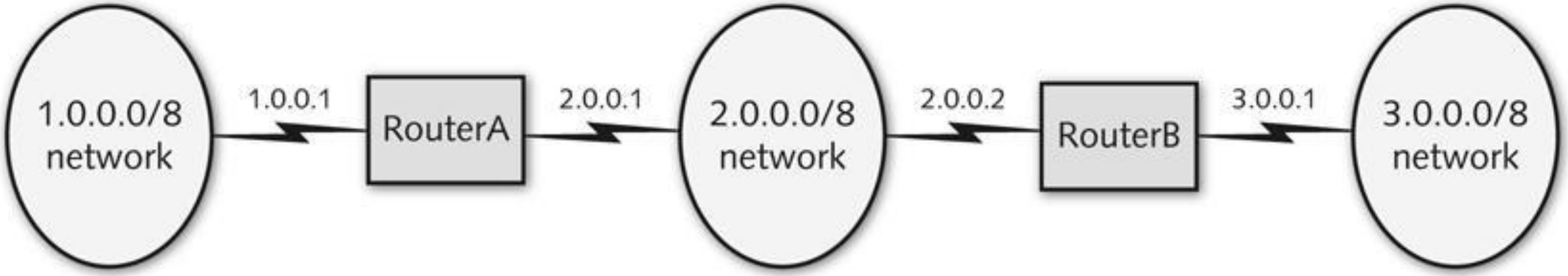


Figure 12-7: A sample routed network

1.0.0.0/**8** ??
8 ones -- subnet mask is 255.0.0.0

Routing

- Most routers are configured with a default gateway
  - For packets addressed to destinations not in route table
- `traceroute`
  - Used to troubleshoot routing
  - Displays all routers between the current and a remote computer
  - To trace an IPv6 route, use the `traceroute6` or `tracepath6` commands

```
$ traceroute www.google.com
```

Network Services

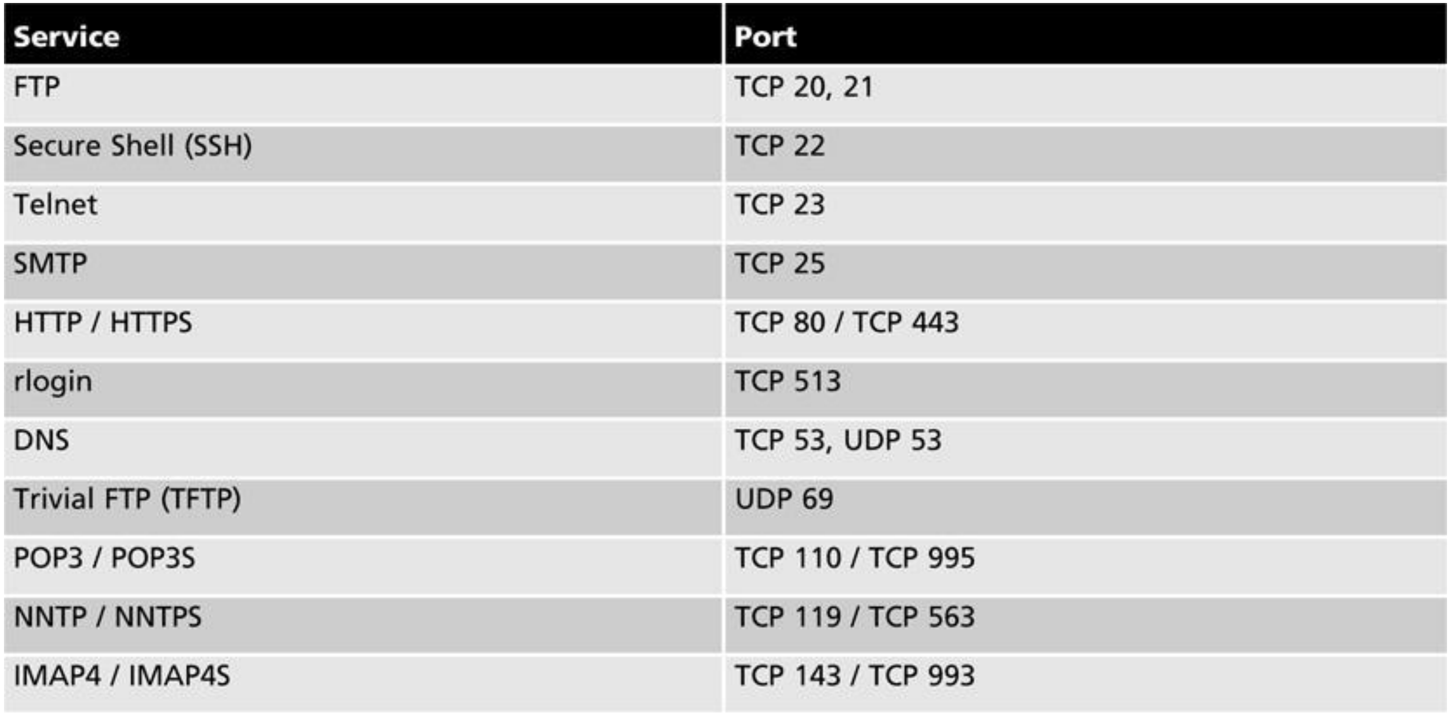
- Must identify types and features of network services before they can be configured

- Network services
  - Processes that provide some type of valuable service for client computers on network
  - Often represented by daemon processes that listen to certain requests
    - Daemons identify packets to which they should respond using a port number
      - Ex: CUPS web admin

Network Services

- Port
  - Number uniquely identifying a network service
  - Ensure that packets are delivered to the proper service
  - Range from 0 to 65534
- /etc/services file
  - Lists ports and associated protocols
- Well-known port: ports from 0 to 1023
  - Represent commonly used services

Network Services



Service	Port
FTP	TCP 20, 21
Secure Shell (SSH)	TCP 22
Telnet	TCP 23
SMTP	TCP 25
HTTP / HTTPS	TCP 80 / TCP 443
rlogin	TCP 513
DNS	TCP 53, UDP 53
Trivial FTP (TFTP)	UDP 69
POP3 / POP3S	TCP 110 / TCP 995
NNTP / NNTPS	TCP 119 / TCP 563
IMAP4 / IMAP4S	TCP 143 / TCP 993

Table 12-2: Common well-known ports

Remote Administration

- There are several ways to perform command-line and graphical administration of remote Linux servers:
  - Telnet
  - Secure Shell (SSH)
  - Virtual Network Computing (VNC)

Telnet

- `telnet`
  - Traditionally used to obtain a command-line shell on remote server
  - Receives host name or IP address of remote computer as argument
  - Easiest way to perform remote administration
- Telnet is not installed by default on most modern Linux distributions
  - Can be installed from a software repository
- Use regular commands and `exit` to kill remote BASH shell

Secure Shell (SSH)

- Secure Shell (SSH)
  - Encrypts information passing between computers
  - Secure replacement for telnet

- ssh
  - Connects to a remote computer running ssh daemon (sshd)
  - Receives host name or IP address of target computer as argument
  - Accept RSA encryption fingerprint for target computer
  - Can be used to transfer files between computers

Secure Shell (SSH)

- SSH is used to perform command-line administration of remote systems
  - The `–X` option to the `ssh` command can be used to tunnel X Windows information through the SSH connection if you are using the `ssh` command within a GUI environment

- By default, sshd uses a secure challenge-response authentication method that ensures that the password is not transmitted on the network
  - Can be changed to Kerberos authentication

Secure Shell (SSH)

- Main types of encryption supported by ssh daemon:
  - Symmetric
    - Triple Data Encryption Standard (3DES)
    - Advanced Encryption Standard (AES)
    - Blowfish
    - Carlisle Adams Stafford Tavares (CAST)
    - ARCfour
  - Asymmetric
    - Public/private key

X Forwarding (Windows)

- Need Windows X server and fonts
  - VcXsrv: chttps://sourceforge.net/projects/vcxsrv/

- Install on server (with yum):
  - xorg-x11-xauth
  - xorg-x11-fonts-*
  - xorg-x11-utils
- Enable the following in the /etc/ssh/sshd_config file
  - X11Forwarding yes

X Forwarding

- Start Xming
- PuTTY
  - Hostname: <username>@IP address
  - Connection → SSH → X Forwarding
    - Enable X forwarding
    - Add **localhost:0** to box for location (not necessary)
- OSX (untested)
  - Need to enable X forwarding on server (prev. slide)
  - ssh –Y <username>@IP
    - -Y is trusted X forwarding (-X is standard)