[StuCo 98008]
GNU/Linux for Beginners

Session 9

Networking

By the end of this lecture you will know

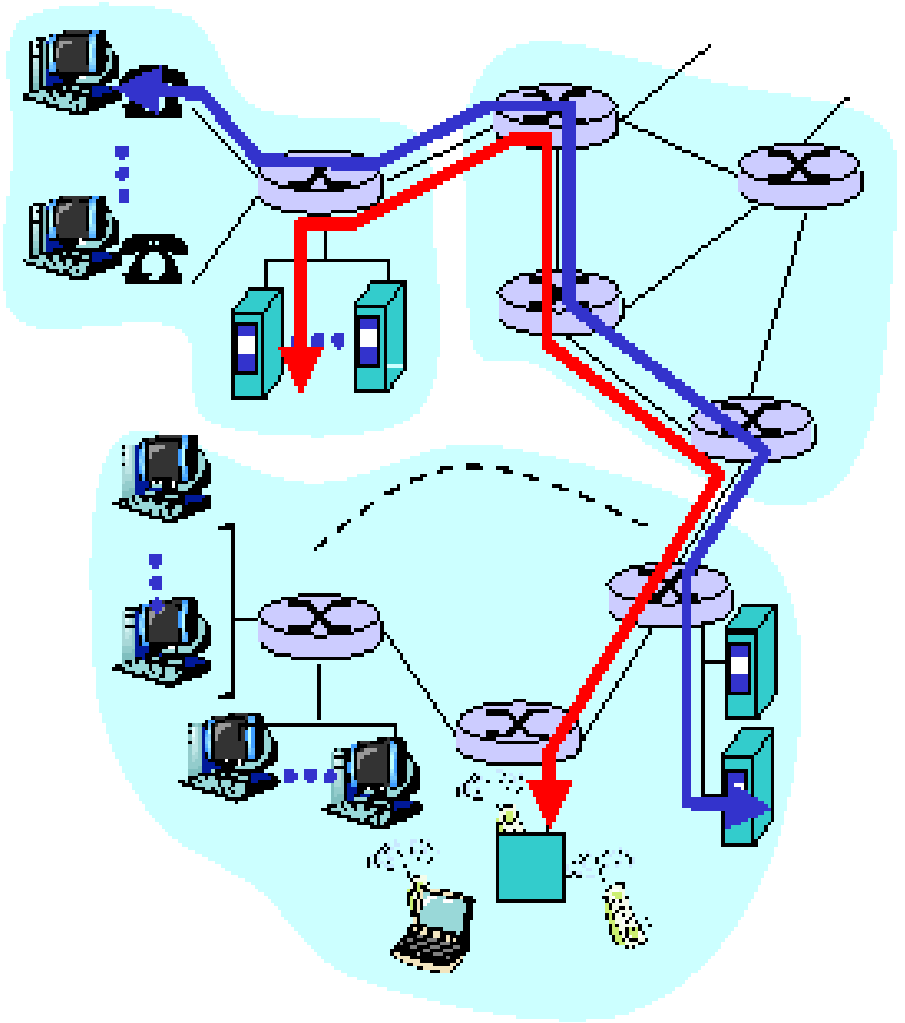
- Basic networking terminology
- The protocols used in modern networks
- Some common network applications and servers.

Networking Concepts

- **Node:** a logical data processing unit that is part of a network (e.g. a PC, a server, a router)
- **Client:** anything receiving data from the network
- **Server:** anything that provides data to the network
- **Protocol:** A clearly defined method of communication
- **Bandwidth:** The maximum amount of data that can be exchanged between two nodes in a unit of time (usually one second)

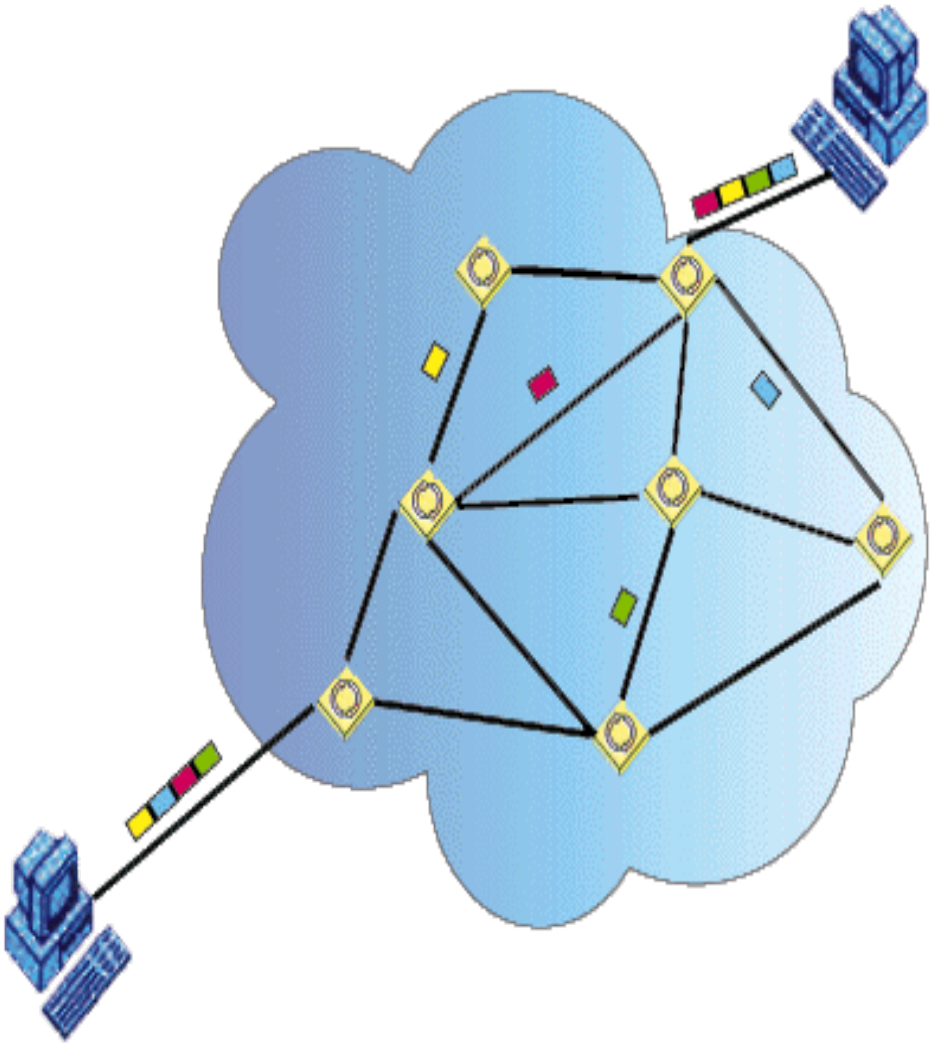
The old way of transferring data

- **Circuit-switched networks**
- No sharing of network resources
- Guaranteed bandwidth
- Static path, defined *before* the connection takes place



The new and improved way

- **Packet-switched networks**
- Information is broken down to network frames (“packets”, in IP)
- Resource sharing
- Unpredictable path, dynamically constructed



IP Addresses

- Each packet has a source and destination address
- Example: 192.168.15.48 (decimal notation)
- 11000000.101010000.00001111.00110000 (binary notation – four 8-bit parts – “octets”)
- Boundaries: 0-255 for each octet
- Try counting beyond 255 with 8 bits!

DNS

- A system to aid humans remember network addresses: **D**omain **N**ame **S**ystem
- DNS servers keep mappings of IP addresses and easy-to-remember names:
  - www.cmu.edu. CNAME WEB3.ANDREW.cmu.edu.
  - WEB3.ANDREW.cmu.edu A 128.2.11.43
- Also, aliases of easy-to-remember names for (drums...)EVEN MORE easy-to-remember names
- DNS is a hierarchical, (sort of) distributed system
  - If my DNS server doesn't know an address, it asks its *parent* DNS server
  - And so on, up to the *root DNS servers* (that have all the answers)

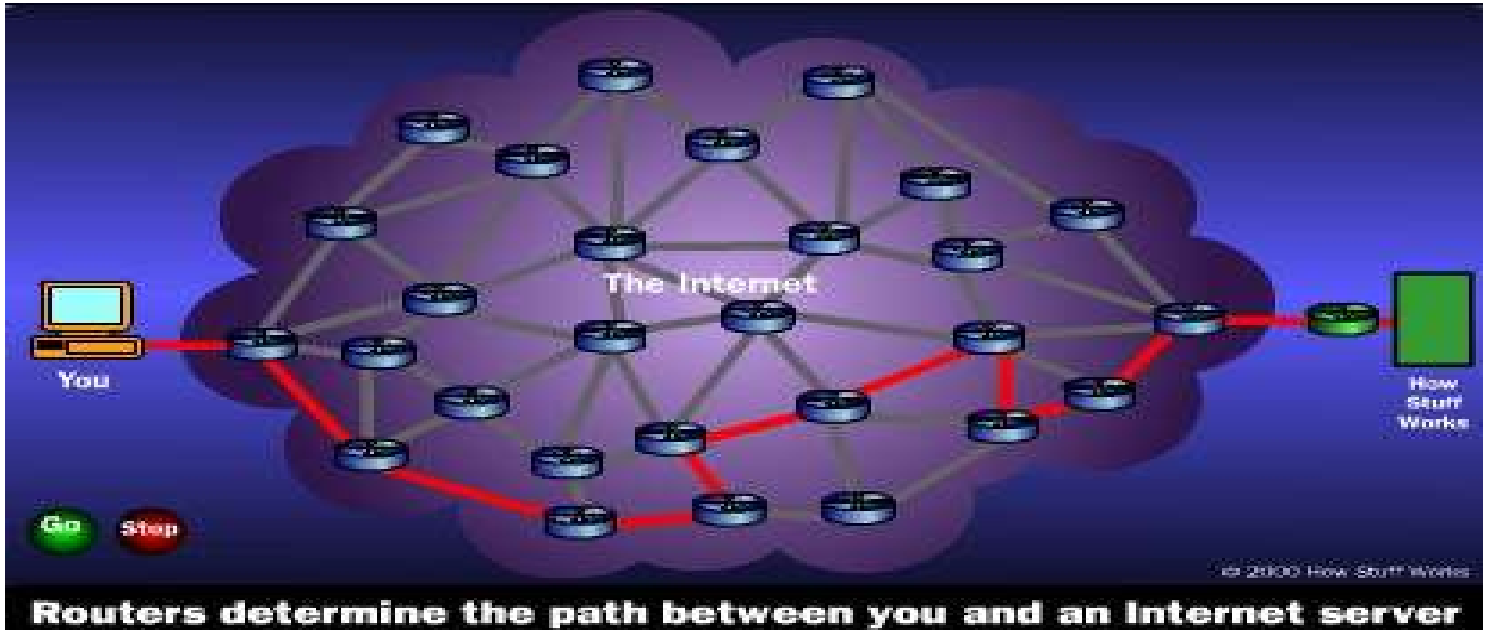
Network/Host Addressing

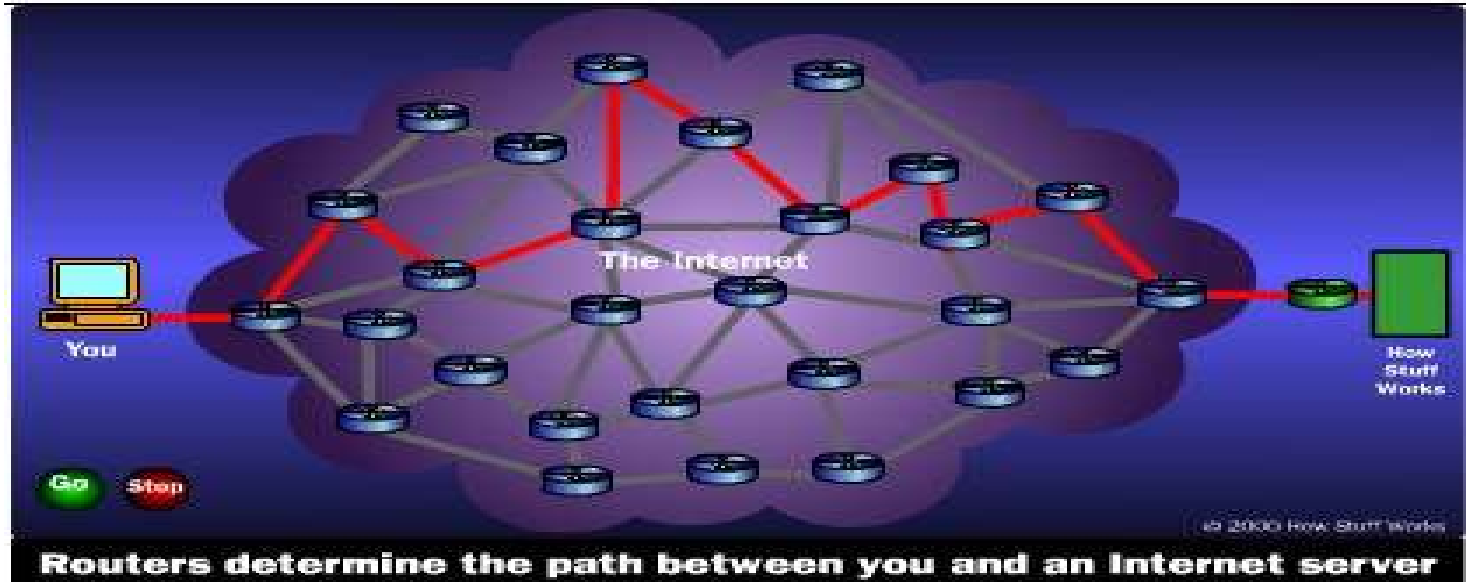
- Each address encapsulates **two** pieces of information
  - Network ID (e.g. CMU, Scaife Hall)
  - Host ID (a specific machine)
- 192.168.15.48/**24**
  - The first **24** bits are the network ID
  - The remaining **8** bits (32-24=8) are the host ID.

Routing

- **Routers** are dedicated nodes that decide how the packets will be delivered to their destination
  - Shortest path
  - Fastest path
  - Route around network failures
- Routing tables: where should I forward a packet with an Australian destination address?
- Routers have limited knowledge of their environment (impossible to have complete knowledge of the entire Internet's state)

Routes Change All The Time



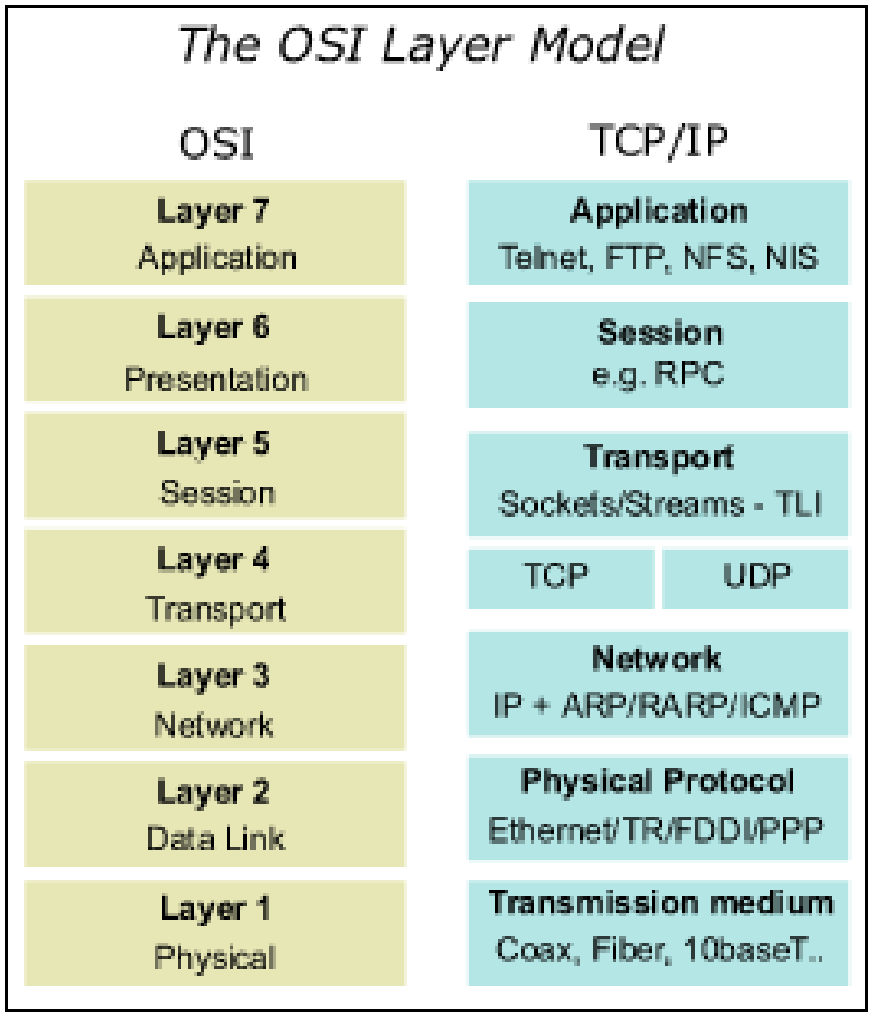


Source: http://computer.howstuffworks.com/router2.html

Protocols For Packet Transfer

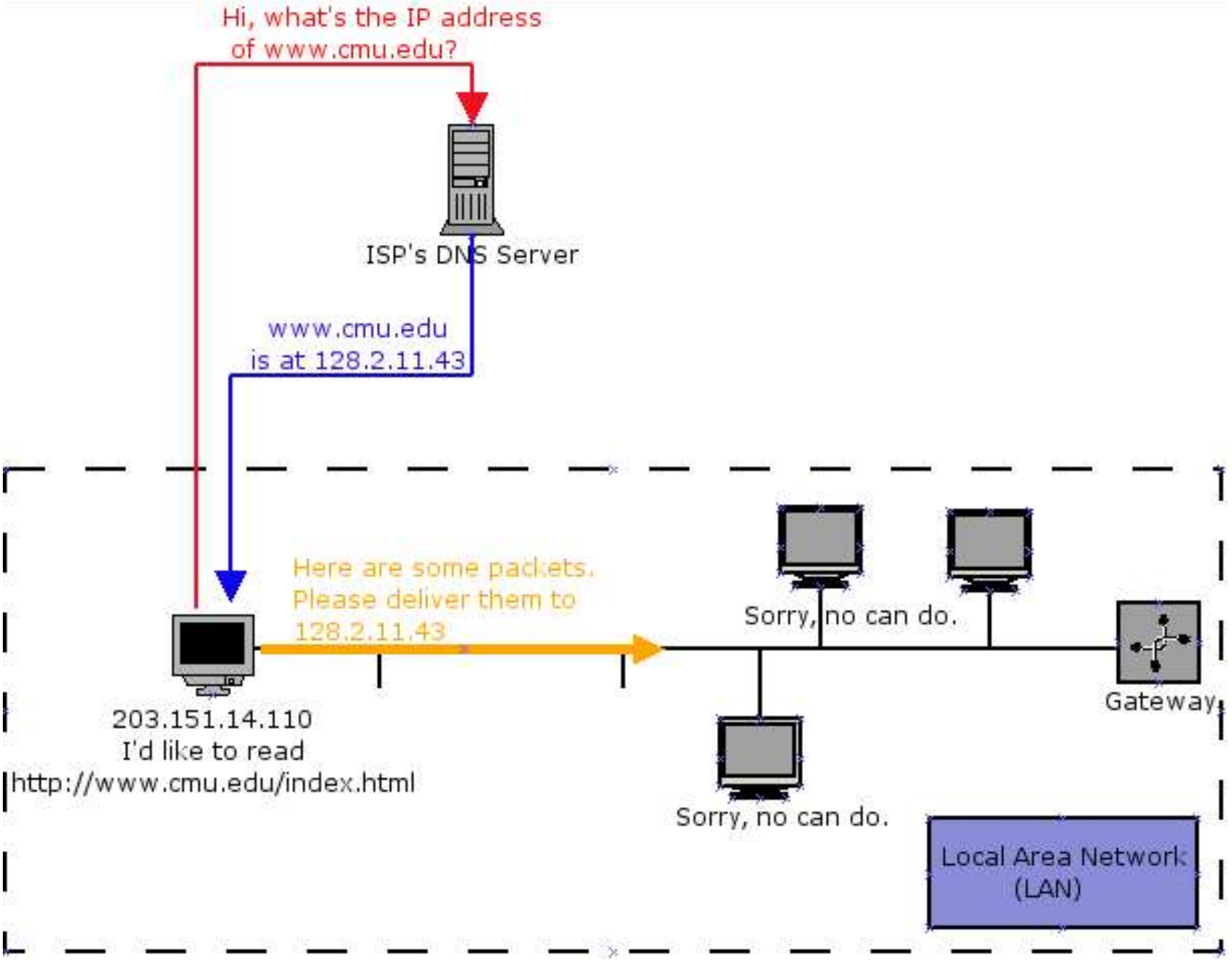
- Connectionless
  - Packets of arbitrary size are dispatched as soon as application provides data.
  - Nobody knows if they arrived or not.
  - Example: **U**ser **D**atagram **P**rotocol
- Connection-Oriented
  - A connection is first established between source and destination.
  - Checks to make sure that everything is transferred properly (sequence numbers, retransmission, timeouts)
  - Example: **T**ransmission **C**ontrol **P**rotocol

The OSI Model

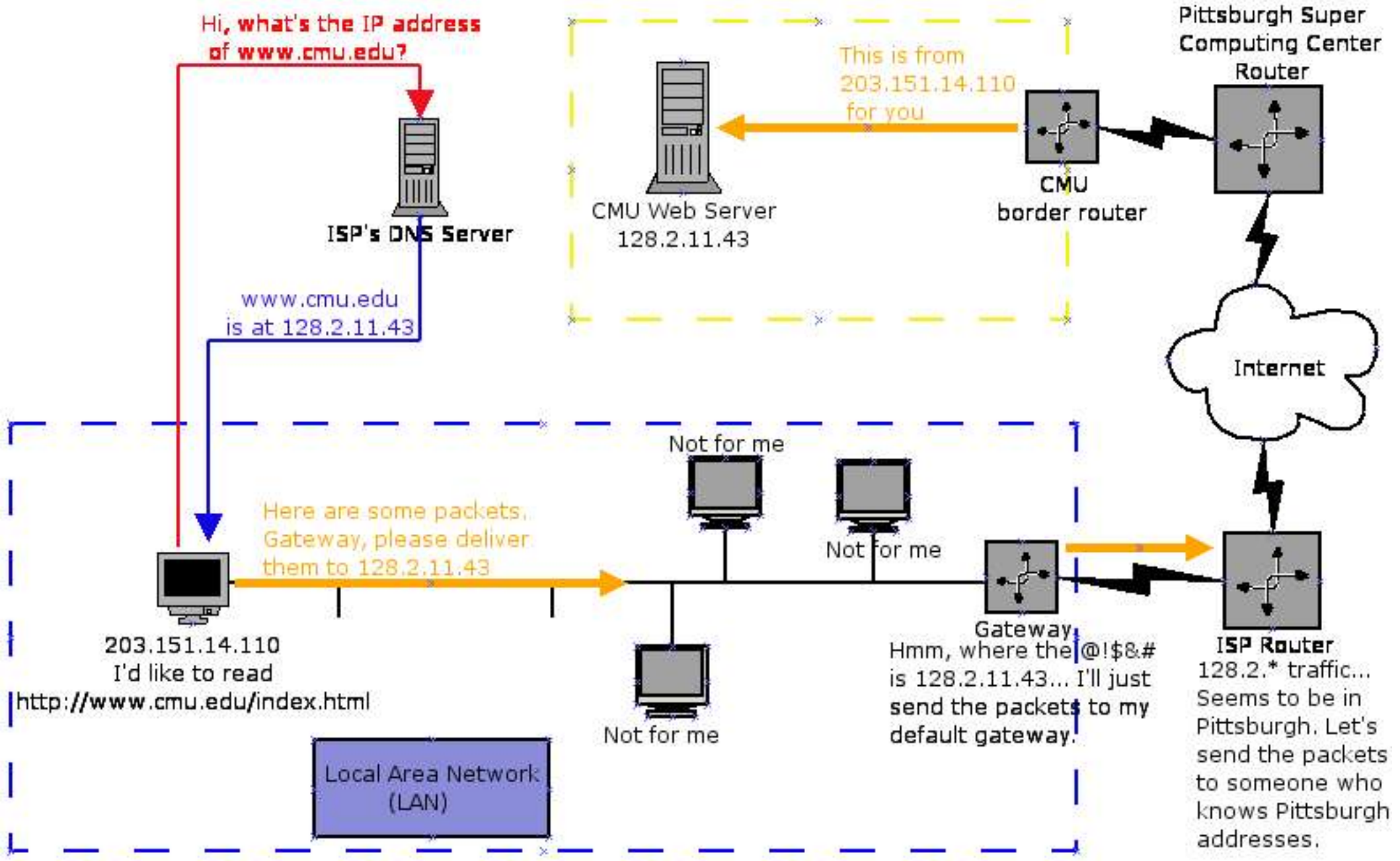


- I instruct my browser to fetch a web page. It sends an HTTP request down the stack.
- TCP breaks it up in segments of optimal size, assigns sequence numbers, and passes the segments down to IP.
- IP creates packets, assigns source and destination addresses, and passes them down the stack.
- Ethernet frames are created to carry my data to the next node.
- My network card translates these frames to electricity and puts them on the wire.

Getting A Web Page



Getting A Web Page (2)



IP Ports

- Problem: When a host receives data from the network, it has no way of knowing what application that data is trying to reach.
- Solution: Use IP ports and bind applications to "sockets" (IP:port combinations)
  - E.g. any data sent to 128.2.11.43**:80** will be handled by CMU's web server.
  - Data send to 128.2.11.43**:22** will be handled by that machine's SSH daemon
- There are 65535 ports.
  - 0-1024 are considered "privileged" - only root can bind applications to them

Network Interfaces

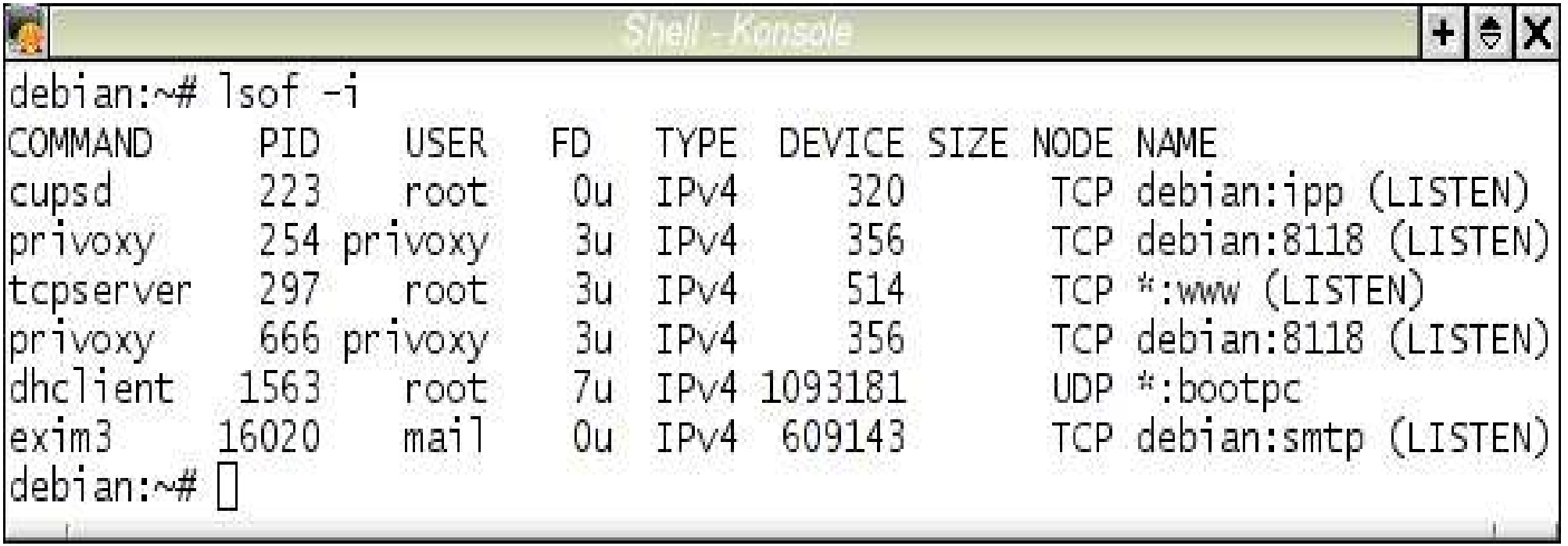
- Ethernet (802.3) card
  - eth0, eth1, etc...
- Wireless (802.11) card
  - wlan0, wlan1, etc...
- Modem (PPP, ADSL)
  - ppp0, ppp1, etc...
- Virtual Interfaces!
  - eth0:1, eth0:2 etc...

InterFace **CONFIG**uration

- /sbin/ifconfig
- Mortals simply see information, cannot change stuff
- $ /sbin/ifconfig -a *lists all interfaces*
- # ifconfig eth0 192.168.0.1 up *activate the first ethernet interface and assign it the given address*
- $ man ifconfig for more!

Listening Ports

- When an application is bound to a port and is **listening** for incoming connections **from any address**, your computer is acting as a server



The Routing Table

- # route
  - # route -n (for numerical addresses only)
  - # route add default gw 10.0.0.1
  - # route add -net 192.168.0.2 dev eth1

Automatic Interface Configuration

- Depends on your distribution
  - RedHat, Mandrake have graphical tools
  - Most configuration files under /etc/sysconfig/
  - Debian, Slackware: command line
- DHCP addresses (server-assigned dynamic addresses)
  - # dhclient eth0 (*/etc/dhclient.conf*)
  - # ifup eth0 (*configuration files in /etc/network*)
- DIY interface handling (!)
  - Just put the commands you need in a script

Network Services

- Controlled by scripts under /etc/init.d/
- Your runlevel dictates what services are active
  - # grep initdefault /etc/inittab *(this is your default runlevel)*
  - # runlevel *(this is your current runlevel)*
  - # ls -l /etc/rc2.d/ *(this is what will happen on entering runlevel 2)*
  - start/stop/restart parameters for all scripts
  - **S** scripts **start** services (/etc/rc2.d/S20snort)
  - **K** scripts **kill** services (/etc/rc2.d/K11anacron)
  - The numbers denote priority in ascending order

Some Common Services

- /etc/init.d/httpd (in Red Hat systems) - Apache
- /etc/init.d/wu-ftpd – FTP daemon
- /etc/init.d/sshd – OpenSSH server
- /etc/init.d/xinetd – The Internet Super-Server
  - Controls lots of small servers
    - fingerd
    - talkd
    - etc...
  - Configuration file is (usually) /etc/xinetd.conf

Network Troubleshooting Recipe

1. **$ ping 127.0.0.1** – If it fails:
   - **# ifconfig lo 127.0.0.1 up** – still no good?
   - Your local networking, or firewall, is not set up properly.
2. **$ /sbin/ifconfig**
   - If it shows the loopback interface only, the driver for your NIC is not loaded. Look into /etc/modules.conf and the output of **# lsmod**
   - If you see your interface (e.g. eth0), but no IP assigned, ask for one: **# dhclient eth0** – or assign one statically with **# ifconfig eth0 10.0.0.57 up**
   - You see the interface, it has an IP, but still you can't talk to the network? **$ ping <my_IP_addr> -** if it fails, your NIC is bad.

Recipe continued

3. **$ ping <gateway_IP> -** Usually **$ /sbin/route -n** will tell you what your gateway is. If it fails, your cable is bad.

4. **$ host www.cmu.edu –** If this fails, your DNS resolution doesn't work. Check /etc/resolv.conf – it should have at least one nameserver's IP listed there, and you should be able to ping that IP.

5. **$ ping www.cmu.edu –** If this fails, your gateway is not routing your traffic to the Internet. Talk to your network administrator, your local configuration seems to be fine.

What we covered

- Types of networks
  - Circuit-switched (telephone)
  - Packet-switched (Internet)
- Types of protocols
  - Connectionless (UDP)
  - Connection-oriented (TCP)
- How hosts are located across the network
- Basic networking utilities, services, troubleshooting