LECTURE 3.5
NETWORKS

COP4600

Dr. Matthew Gerber

4/20/2015

REVIEW: MESSAGE PASSING AND SOCKETS

Message Passing

- All message passing involves *sending* and *receiving*.
- There are three types of addressing…

- Symmetric

```
send(P, message)        message = receive(P)
```

- Asymmetric

```
send(P, message)        receive(var message, var sender)
```

- Mailbox *(but how do the rules work?)*

```
send(M, message)        message = receive(M)
```

More About Message Passing (3.4.2)

- Sending and receiving can be either *blocking* or *non-blocking* – including several queued variations
  - *Blocking sends* force a process to wait until there is space in the send queue
  - *Non-blocking sends* fail (not the same thing as producing an error) if there isn't space in the send queue
  - *Blocking receives* force a process to wait until data are available
  - *Non-blocking receives* fail if no data are available

Pipes (3.6.1)

- The *pipe* model implements message passing using the file metaphor
  - Varying degrees of complexity and flexibility
  - *Ordinary* pipes can only communicate between parent and child processes
  - *Named* pipes, once created, can communicate between any number of processes
  - They sit in the file system, and can be opened, closed, read and written by any process with permissions to do so

Sockets (3.6.3)

- *Sockets* are metaphors used for pipes that go between machines on a network
  - Properly, the socket is the *endpoint* of the pipe
  - Each active network connection therefore has *two* sockets
- Rather than having a filename, a socket has a network address and port
- Sockets are the fundamental metaphor used by TCP/IP networking
  - …and are based on pipes
  - …which are based on message passing

THE NETWORK STACK

Theory

- The *Open Systems Interconnection* model conceptually defines every layer of networking – *including* the data to be exchanged
- The *session* defines the *exchange* of the data
- The *presentation* defines the *format* of the data
- The *application* determines the *semantics* and *usage* of the data

Network Layers
7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data Link
1. Physical
Medium

Reality

- Everybody actually uses the *Internet Protocol Suite*, which doesn't enforce any of those boundaries
- Those boundaries exist to *some* extent application by application, but they are defined only by individual applications
- The IP suite doesn't define anything below the link layer…
- …but in actual use, layers 1 through 4 of OSI really have caught on

Network Layers
Application
4. Transport
3. Network
2. [Data] Link
1. Physical
Medium

Medium

- The *medium* is what packets *travel over*
  - For practical purposes, media include cables and electromagnetic waves
  - Obviously, only certain bands of the electromagnetic spectrum are useful
    - Most typically you see wireless networking done using radio waves
    - Infrared and other light-based links *have* been used
  - All other things being equal, a good cable is always more reliable than any wireless link…
  - …but is a lot harder to set up, and a lot less flexible

Physical

- The *physical layer* is the *transmitters* and *receivers* that, in turn:
  - Render *symbols* (for our purposes, bits) into physical *signals*
  - Commit the signals to the medium
  - Retrieve the signals from the medium
  - Convert the signals back into symbols
- The *radios* on your wireless networking adapters are physical-layer devices
- So are the *1000BASE-T* transceivers that transmit *Gigabit Ethernet* packets over twisted pair cable
  - (*Specifically* twisted-pair cable – over fiber, it's 1000BASE-X)

Data Link

- The *data link layer* is the protocol layer that transmits and receives data between *nodes* in a single *network*
- At the levels most of you have worked with, data link layers are completely dominated by Ethernet and its variants
- Its *Media Access Control (MAC)* functionality:
  - Addresses *frames* between devices on the same network
  - Manages collisions between multiple devices trying to use the network at once
- Network adapters, switches and hubs are data link-layer devices
- Routers are *not*

Network

- Provides transmission of data packets between points on an *arbitrarily large* network *of networks*
- The ***Internet Protocol***:
  - Allows individual networks to be interconnected, and packets to be routed between these sub-networks
  - Addresses packets unambiguously from sender to receiver, given an *IP address* and a *port*
  - Routes packets from the sender to the receiver, following the path of nodes between them
  - Locates a path for those packets arrive at their final destination, regardless (within reasonable limits) of how many nodes must be on the path

Transport

- Provides *reliable* transmission of data packets between points on an arbitrarily large network
- The ***Transmission Control Protocol***:
  - Creates the “virtual circuits” associated with sockets
  - Retries messages multiple times in case of failure
  - Ensures that messages arrive with a low probability of error
  - Throttles network traffic to a pace that all devices in between the two points can handle
  - Ensures that messages arrive in order or not at all
  - Ensures that messages arrive in the same format that the sender sent them
  - Provides assurance to the sender that messages have been received
- **When you open a socket, this is what you are using**

Application

- ***Applications*** are what provide network services to the *user*
  - The classic Internet application is the World Wide Web
  - Other applications include E-mail, Secure Shell, online games…
- Applications use the transport layer…
  - …which uses the network layer…
  - …which uses the data link layer…
  - …which uses the physical layer…
  - …which uses the medium…
  - …and all the way back up again

NETWORKING AND OPERATING SYSTEMS

Networking Organization

- The application provides user-level functionality
- The operating system provides everything else…
- …that the hardware doesn't
- Classically:
  - Transport and network are exclusively handled in the OS
  - Physical is exclusively handled in hardware
  - Data link is the interface between the two…
    - …and is the device that has a device driver in the classic sense

Filtering the Network

- Network *filters* give the operating system the capability to allow, disallow, drop or modify network packets on a given computer
- Most of the time you are creating a software-based “firewall”, it is really more accurate to say that you are using network filtering
- Linux implements a general-purpose network filtering mechanism called ***Netfilter***, which allows kernel modules to register themselves as providing filtering and modification rules
- If you are interested in network filter programming, you already know how to write kernel modules now!

NEXT TIME: EXAM REVIEW