CS162, Spring 2002
Section 12
Steve Martin

Announcements

- Design Reviews are ongoing
    - Haven't signed up yet? DO SO ASAP.
- Code due Thursday, 4/22 (week from today)

Today's Menu:

- ACL's vs. Capabilities
- Distributed Systems
- Intro to Networking: Ethernet and IP

ACL's and Capabilities

- The take-home from lecture about these:
- ACL's:
    - For each File:
        - Keep track of user A's permissions. . .
        - Keep track of group B's permissions. . .
        - Etc. . . (form a matrix)
- Capability Lists:
    - For each User:
        - Keep track of permissions for File A
        - Keep track of permissions for File B
        - Etc. . . (list of objects)

Distributed Systems

QUESTION: What is a distributed system?
ANSWER: Physically separate computers working together.

- Some examples?
    - Clusters (EmuLab, Earth Lab, CERN, Stanford's supercomputer, the new one in Virginia)
    - Flash-mob supercomputing! (recently in SF!)
    - OceanStore (here)
- Why do we want distributed systems as opposed to huge mainframes?
    - Cheap to build lots of simple computers
    - Easy to add power incrementally
    - Better reliability
        - One node fails? Another can take its place.

        - Data stored in multiple locations.
- What's hard about distributed systems?
    - Components can fail independently
    - Network link between components can drop/reorder/corrupt/delay messages
    - Network link between two components of a system can go down
    - All nodes may not be trusted: may have nodes that are actively trying to corrupt and confuse the rest of the system
    - Can't synchronize events in time (General's Paradox)
        - How does one keep a consistent time across a distributed system?
    - Difficult to maintain consistent view of shared state across all nodes

<u>Intro to Networking</u>

- We'll be talking about this for a while
    - So if today seems fast, don't worry.
    - Very much a 'how things really work' section—ask questions!
- Networking layers: Each part of a network communicates a certain level of abstraction
    - For example, Web browser uses the HTTP protocol using. . .
    - A packet router that manipulates lower-level IP packets.
- One method of abstraction: the ISO/OSI Seven Layers of Networking
    - Physical Layer
        - Describes the physical properties of the various communications media, as well as the electrical properties and interpretation of the exchanged signals.
            - Ex: this layer defines the size of Ethernet coaxial cable, the type of BNC connector used, and the termination method.
    - Data Link Layer
        - Describes the logical organization of data bits transmitted on a particular medium.
            - Ex: this layer defines the framing, addressing and checksumming of Ethernet packets.
    - Network Layer
        - Describes how a series of exchanges over various data links can deliver data between any two nodes in a network.
            - Ex: this layer defines the addressing and routing structure of the Internet.
    - Transport Layer
        - Describes the quality and nature of the data delivery.
            - Ex: this layer defines if and how retransmissions will be used to ensure data delivery.
    - Session Layer
        - Describes the organization of data sequences larger than the packets handled by lower layers.

            - Ex: this layer describes how request and reply packets are paired in a remote procedure call.
    - Presentation Layer
        - Describes the syntax of data being transferred.
            - Ex: this layer describes how floating point numbers can be exchanged between hosts with different math formats.
    - Application Layer
        - Describes how real work actually gets done.
            - Ex: this layer would interact with users, i.e. a web browser.
- A few more common examples for each layer.
    - Application: Telnet, FTP, HTTP, etc.
    - Transport: TCP, UDP, etc.
        - These will be important
    - Network: IP, ICMP, IGMP, etc.
    - Link: Ethernet, PPP, etc.

- **EXAMPLE**: HTTP is encapsulated within TCP
    - which is in turn encapsulated within IP
    - which is encapsulated over whatever physical network the messages happen to be traveling over.
    - So, if we had the following HTTP message which contains the data for a Web page:

<html><body> ... </html></body>
which is called [PAYLOAD] below.

    - This is encapsulated within TCP, which places a header on the front of the message:

[TCP HEADER][PAYLOAD]

    - This in turn is encapsulated within IP:

[IP HEADER][TCP HEADER][PAYLOAD]

    - And finally this is transmitted over a local network, such as Ethernet (which places a header and a trailer on the message):

[ETHERNET HEADER][IP HEADER][TCP HEADER][PAYLOAD][ETHERNET TRAILER]
Size: (14 bytes) (20 bytes) (20 bytes) (4 bytes)

- What are the components of Networking Performance?
    - Overhead - CPU time to put packet on wire.
    - Latency - How long to send one byte packet
    - Throughput - maximum bytes per second

- **QUESTION**: How long would it take to transfer 1 MB across country from Berkeley to Boston where the latency is 15ms and the bandwidth is 10Mbit/s?
- **ANSWER**:
    - 10Mbit/s * 1 Byte/8 bit = 1.25MB/s
    - 0.015s + 1MB/1.25MB/s = 0.815s

- Ethernet
    - Original LAN from DEC, Intel, and Xerox (first standard published 1982).
        - Originally 10 Mbit/sec, then "Fast Ethernet" (100 Mbit/sec)
        - Now "Gigabit Ethernet" (1 GBit/sec).
        - Amazingly the standard has changed very little since the original 10 Mbit/sec Ethernet in 1982!
    - Each packet contains (DRAW):
        - 6-byte destination address
        - 6-byte source address
        - 2-byte type (0x0800 for IP)
        - 46-1500 byte payload
        - 4-byte CRC - a checksum on the entire packet
    - The source and dest addresses are 48 bits and are known as 'MAC addresses'.
        - Only known on a local Ethernet segment -- in order to talk to machines on other Ethernet segments, a hub or router must be used.
        - The addresses look like 00:00:c0:14:db:bf
        - Usually hard-wired into the Ethernet interface of the host.
            - You can identify the vendor that makes the Ethernet interface by looking at the first few bytes of the Ethernet address!
    - Ethernet is a "CSMA/CD" network
        - Carrier Sense Multiple Access with Collision Detection
            - Nodes can detect when another node is speaking on the network
            - Packet collisions are detected.
    - When packets collide, each node retransmits after waiting for a small amount of time.
        - Nodes perform EXPONENTIAL BACKOFF:
            - If there are many collisions, they wait for longer and longer periods of time before retransmitting.
    - Nodes discover the Ethernet address of the destination using ARP (Address Resolution Protocol).
        - A node broadcasts a small message (on Ethernet) with an IP address and requests that someone tell that node what the corresponding Ethernet address is.
        - Someone on the network responds with an ARP reply.
        - ARP lookups are cached and aged out.

- Is this a secure way of sending data?
  - With this setup it is possible for every node on the Ethernet to listen to every packet on that segment
  - These days most use “switched” Ethernet
  - Each node has a (private) line to a local Ethernet switch
  - In most cases the Ethernet is also*full duplex*, meaning that a host can send and receive data at the same time
  - For this reason there is never any collision on such a network.
  - With switched Ethernet, sniffing is no longer a problem.

- The Internet Protocol: IP
  - Ethernet is all fine and good, but is really designed for a local area network.
    - A packet sent to someone not on your network might need to traverse numerous different networks to get there.
    - This is where IP comes into play.
  - IP provides an *unreliable connectionless datagram service*.
    - IP may drop packets (due to network errors, lack of buffer space in routers, etc.) and it is the higher-level protocol's responsibility to detect this and deal with it.

QUESTION: What does ‘Connectionless’ mean?
ANSWER: IP does not "remember" any state about successive messages: it just sends individual messages with no notion of a "connection" between two hosts.

  - This means that messages can be delivered out of order
    - In fact, two back-to-back messages between the same hosts might take totally different paths through the network!

  - **EXAMPLE:** Think of IP as the US Postal Service:
    - All they do is deliver letters between addresses, with no idea how those different letters might be related.
    - They might drop letters at any time or deliver letters out of order.
    - If IP is the US Postal Service, then Ethernet is like the mail truck that delivers messages between your home and the local post office.
  - Each IP packet looks like this:

[12 bytes of various header fields]
32-bit source IP address (e.g., 216.102.32.12)
32-bit destination IP address (e.g., 128.32.131.6)
[options (if any)]
data

  - Note that these are numerical addresses only

        - If you want to use an alphanumeric name like "www.slashdot.org" you have to translate this to a numerical address
            - This is typically done using a separate protocol called DNS, Domain Name Service
    - We will cover how IP packets are routed next time

- TCP: Transmission Control Protocol
    - Reliable, Byte-stream oriented protocol!
    - Just about always runs over IP—you do NOT see TCP running just by itself.
    - We will go over this next time, but consider how you could provide reliability by layering over IP.