CS 336 Lecture 6– 2/18/98

Announcements

- Exercise 2 is due today.

The Data Link Layer

1. Both the physical and data link layer are not really about "networks" but rather about communication between simple pairs of "directly" connected machines.

2. In my introduction to the course, I suggested that the data link layer concentrated on the software issues of a machine to machine connection while the physcial layer worried about the hardware. Like all definitions of the roles of the layers, this isn't entirely accurate. Data link layer issues are sometimes address in hardware.

3. Another way to distinguish these two layer is the way they view the data being transmitted.

   - The physical layer simply transmits a stream of bits from a source to a destination.
   - The data link layer transmits a stream of messages or *packets* (each of which is just a sequence of bits).

4. While a stream of packets of bits is in some sense just a stream of bits, this difference in viewpoint has a very practical implication.

5. While it is not practical to associate control information with each bit in a message stream, it is practical to associate control information with groups of bits viewed as packets. Such control information can support several enhancements to the basic functionality provided by the physical layer:

   - Error Detection and Correction:

     The detection of errors requires that redundant information be included in the data stream that can be used to detect errors. To use this redundant information, the receiving end of the connection must know what collection of bits the redundant information is associated with. Packets provide natural units to associate with error checks.

   - Time Division Multiplexing:

     If a channel is to be shared by taking turns, groups of bits have to be packaged in such a way that some form of "address label" can be associated with the groups so that the receiver will know to which process they should be delivered.

   - Multiaccess networks:

     In some cases the "direct" connection between two machines is not really very direct. For example, in a network based on radio transmission, each message broadcast can he heard by many machines. It is still appropriate to view such a connection as direct because there is no need to perform any routing to find a path through one or more intermediaries from the sender to the receiver. All communications is "direct".

     To make such communications possible, data bits must be grouped into messages and an address field indicating the destination machine must be included. (Obviously, this is similar to the TDM issue).

6. While the notion of messages or packets is quite central to modern networking techniques, the data link layer actually concentrates primarily on three issues in the handling of packets:

   Framing Including information in the bit stream to identify the beginning of and end of each packet.

   Error detection and correction The data link layer usually includes mechanisms for including error detecting codes and arranging for the retransmission of damaged packets.

   Flow Control Arranging for enough communication between the sender and receiver to ensure that the receiver does not get overloaded by a transmitter that is capable of sending message faster than the receiver can process them.

Framing

1. Breaking a stream of bits up into messages may seem like a simple task, but it is complicated by the presence of errors.

   - One simple (but insufficient) approach would be to begin each message with a field that indicate how many bits (or bytes) were contained in the entire message.
   - Unfortunately, if any of the bits in the message length field get damaged, the receiver will have no way of getting back into synchronization so that it can correctly identify the length fields of subsequent packages.

2. A more popular technique is to begin and end each packet with a special sequence of bits that is used to indicate the begin (or end) of a packet.

   - If one of these sequences gets damaged, the receiver might miss one packet, but would get back in synch on the following packet.

3. Since much data sent on the network can contain odd segments of binary data, there is a risk that one of these special start/end sequences might appear in the midst of a message as data.
   - This possibility can be addressed by techniques known as byte stuffing. For example:
     - The ASCII character set includes two special codes associated with characters names STX (start transmission) and ETX (end transmission). These are intended to be used to indicate the start and end of a message.
     - ASCII also contains another character named DLE (data link escape).
     - If an STX or ETX appears in the data of a message, the receiver can simply add a DLE before the STX or ETX to tell the receive not to treat the following character as a marker for a packets boundary.
     - Of course, if a DLE appears in the data, the sender will have to send a pair of DLE's in its place.
   - In messages that do not have any notion of a character set like ASCII associated with them (i.e. message need not contain multiples of 8 bits), a similar technique called bit stuffing may be applied.
   - For a very nice discussion of byte stuffing techniques, see the paper by Cheshire and Baker included in the readings for this topic.

4. In some cases, the framing problem is solved at the physical layer using signals that could not correspond to any sequence of bits can be used to indicate message boundaries.
   - As an example, in a system that uses manchester encoding, a signal that did not have the appropriate number of voltage transitions could be used as a frame marker.

5. Beyond marking the boundaries of a frame, it is typical to map out some skeletal structure for the frame.

   For example, the layout of an ethernet frame is shown below:

Preamble	Ethernet Address of Destination	Ethernet Address of Sender	Packet Length	Message Data	Error Checking Information
64 bits	48 bits	48 bits	16 bits	368 - 12 K bits	32 bits

Error Detection and Correction

1. A significant role of the Data Link layer is to convert the potentially unreliable physical link between two machines into an apparently very reliable link.

2. This is done by including redundant information in each transmitted frame. Depending on the nature of the link and the data one can either:
   - include just enough redundancy to make it possible to detect errors and then arrange for the retransmission of damaged frames, or
   - include enough redundancy to enable the receiver to correct any errors produced during transmission.

   Most current networks take the former approach.

3. Codes for detecting (and even for correcting) errors are largely based on generalizing the simple notion of a parity bit.
   - Given a set of bits, a parity bit is an extra binary digit added to the group so that the total number of one's in the group is even (or odd if you prefer).
   - For example, a few "random" bytes of binary data are shown below with appropriate digits added as parity bits in a ninth column.

0	1	0	1	0	1	0	1	0
0	1	1	1	1	0	0	1	1
0	0	0	0	0	0	0	0	1

4. Used in this simple way, parity bits are a rather weak mechanism for detecting errors.
   - A single parity bit can not detect any error in which an even number of bits are changed.
   - Errors in real communication lines tend to effect many consecutive bits rather than one isolated bit (these are called *burst errors*.

5. By associating several parity bits with a message so that each parity bit is determined by a different subset of the message bits and being clever about the placement of the parity bits one can do much better.
   - Simply adding two bits per byte with one bit for the odd positions and one for the evens enables one to recognize any error of up to three bits.
   - If one organized data into a table, adds a column of parity bits as above and then sends the data by columns from left to right, one can detect any error whose length is less than two columns.