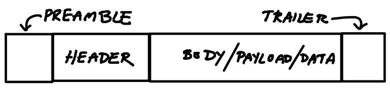
Framing

After this lecture you should be able to: determine if a data communication system requires framing or not, and choose between bit- and byte-oriented framing. You should be able to insert and remove escape sequences and bit stuffing from byte- and bit sequences respectively. For each of the framing techniques described in this lecture (line coding violations, byte escape sequences, and HDLC flags) you should be able to: write out a properly-framed bit- or byte-sequence, and extract the data sequence from a bit- or byte-sequence that contains framing information. You should be able to add and remove padding bits.

Introduction

Data communication systems often need to group bits or bytes into logically related groups called "frames." Each frame contains information that is grouped and treated together. All of the data within a frame typically comes from the same data source and is destined for the same sink. The data within a frame typically is treated as a whole for the purposes of error checking and for routing through a network.

A frame typically consists of a header, data and a trailer:



- the header contains information about the frame. This can include destination and source addresses, frame length, priority, and many other fields. The header sometimes includes a "preamble" for physical-layer synchronization.
- the data is typically a variable-length field and contains the "payload" data that is being communicated.
- the trailer often includes a checksum that allows for detection of errors and may include other fields.

Bit- and Byte-Oriented Framing

Some protocols transmit arbitrary-length sequences of bits while others transmit data as complete bytes (multiples of 8 bits). The type of framing must be consistent with the underlying structure of the data.

Bit-oriented protocols are often used when maximum efficiency is desired or the data is in the form of discrete bits rather than bytes. The length of each portion of data within a frame can be set to the number of bits required. An example is the bits used to control hardware such as the values read from status registers or written to control registers.

Byte-oriented protocols are typically used when the underlying data source consists of bytes. Most modern data networks use byte-oriented framing.

Physical-Layer Framing Information

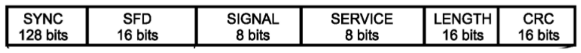
One way to mark the start or end of frame is to use a unique physical-layer waveform. One example is to violate some rule of the line code. For example, it could involve transmitting the same polarity mark twice when using an AMI line code.

Exercise 1: Draw the waveform for an AMI-NRZ encoded sequence of bits '011011' assuming the previous mark was transmitted as a positive pulse. Draw the waveform assuming the second '1' indicates the start of a frame.

The impact on physical layer characteristics such as the DC content of the signal is minimized because this violation only happens at the start, and possibly the end, of the frame.

Another common PHY-layer framing techniques is to use a *preamble*. A preamble is a special type of PHY-layer "frame before the frame." The preamble is encoded in a way that can be easily decoded and carries information about the rest of the frame. Typically the preamble will help the receiver synchronize to the received signal and may contain additional information required to receive the frame such as the data rate to be used in the rest of frame and the frame duration.

The following figure, from the IEEE 802.11 specification, shows the preamble used by an 802.11 wireless LAN frame:



SYNC is a waveform used by the receiver for synchronization. SFD (start of frame delimiter) marks the end of the SYNC field. The SIGNAL field describes the modulation format and data rate and the LENGTH field is the duration of the frame in microseconds. The CRC field is used for error detection.

Exercise 2: Preambles such as this allow multiple transmission standards to co-exist on the same channel. What might be some advantages of this? What might be some disadvantages of using such a preamble?

Escape Sequences

An escape sequence consists of a special character that is used as a prefix to indicate that the following character should not be treated as data but as a command to perform some action. For example, if the ASCII ESC (0x1B) character is used as the escape character, then ESC followed by another character is interpreted as a special command. For example, ESC followed by the character 'A' (ESC-A) might mean to move the cursor up one line.

To allow the escape character itself to be transmitted, an escape sequence can be defined that means "insert one escape character." Often this sequence is the escape character repeated twice.

Exercise 3: By how much does the use of escape characters slow down a link transmitting a continuous stream of escape characters?

Byte-Oriented Framing

Escape sequences can also be used to mark the start and/or end of a frame. For example, we can insert one escape sequence to mark the start of a frame and another to mark the end of a frame.

Another alternative is to use specific characters to indicate the start or end of a frame. Of course, if these characters are to be allowed within the frame some sort of escape sequence must be defined to "escape" them.

For example, the PPP protocol is often used to frame IP (Internet Protocol) packets when they are to be sent over channels that do not support other means of separating frames such as asynchronous ("RS-232") serial interfaces.

PPP uses the PPP 'flag' character (0x7e) to indicate the start and end of a frame and the PPP escape character (0x7d, not to be confused with the ASCII ESC character, 0x1b) to escape both flag and escape characters within the frame.

Exercise 4: What sequence of bytes would be sent to transmit a PPP-encapsulated frame containing the bytes 0xff 0x03 0x7d 0x1b 0x7e?

The use of specific characters for framing is the origin for the names of many of the ASCII control characters such as STX (start of text), ETX (end of text), and ESC/DLE (escape or "data link escape"). These were used in protocols, such as "Bisync", that are now obsolete.

Bit Stuffing

Bit stuffing is a technique used both for framing and to avoid long runs of all 1's or all 0's. The transmitter and receiver both count the number of consecutive 1's (and/or 0's) seen and if this count reaches some upper limit the transmitter inserts a bit of the opposite value and the receiver removes this extra bit.

The disadvantage of bit stuffing is that the data rate is no longer predictable because it depends on how often bit stuffing happens which in turn depends on the data values being transmitted.

Exercise 5: You receive the sequence of bits 10001101 and are told that bit stuffing was used to limit runs of 0 to three or fewer. What was the original data sequence?

Bit-Oriented Framing

Bit-oriented framing uses a special bit sequence to indicate the start and end of each frame. The most common bit-oriented framing protocol is that used by the HDLC (High-Level Data Link Control) and many similar protocols.

This framing technique uses a flag sequence of consisting of six '1' bits preceded and followed by one zero bit (01111110). This "flag" sequence is added before and after each frame.

When this sequence occurs in the data within the frame, bit stuffing (stuff a 0 after five 1's) is used to make sure this sequence does not occur within data.

Exercise 6: Write out the complete sequence of 1's and 0's required to transmit the 12 bits 0110 1111 1100. Include the start and end flag sequences and any necessary bit stuffing.

Exercise 7: An HDLC receiver sees the sequence 1000 0111 1110 1111 1001 0111 1110 0110. What data bits were contained within the frame?

Padding

The physical layer transmits symbols that may contain more than one bit. Thus the number of symbols required to transmit a frame may not be an integer. In this case additional "padding" bits must be added to increase the number of bits to the next-largest multiple of the number of bits per symbol. These extra padding bits are ignored by the receiver.

Exercise 8: A physical layer transmits 3 bits per symbol. A frame of 128 bytes is being transmitted. How many padding bits will have to be added to the frame?