CSE 365: Communication Engineering

Chapter 11: Data Link Control (DLC)

Framing

- The data link layer needs to pack bits into frames, so that each frame is distinguishable from another.
- Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter.
- Frames can be of fixed or variable size.

Fixed Size Framing

- In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter.
- An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells.
- ATM: Asynchronous Transfer Mode is connection oriented, high-speed network technology that is used in both LAN and WAN over optical fiber and operates upto gigabit speed.

Variable Size Framing

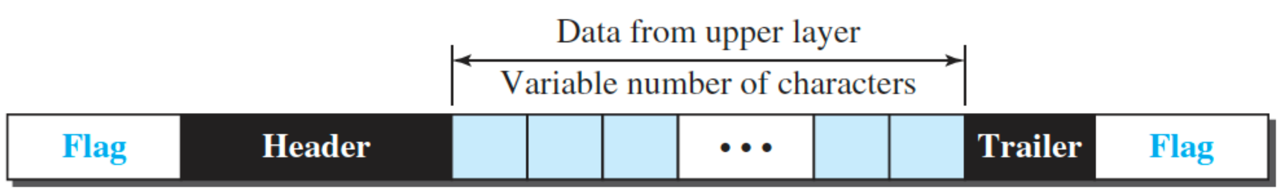
- In variable-size framing, we need a way to define the end of the frame and the beginning of the next.
- Two approaches were used for this purpose:
  - a character-oriented approach and
  - a bit-oriented approach.

Character-Oriented Framing

- In a character-oriented protocol, data to be carried are 8-bit characters from a coding system such as ASCII.
- The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits.
- To separate one frame from the next, an 8-bit (1-byte) *flag* is added at the beginning and the end of a frame.
- The flag, composed of protocol-dependent special characters, signals the start or end of a frame.

Character-Oriented Framing

- In a character-oriented protocol, data to be carried are 8-bit characters from a coding system such as ASCII.
- The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits.
- To separate one frame from the next, an 8-bit (1-byte) *flag* is added at the beginning and the end of a frame.
- The flag, composed of protocol-dependent special characters, signals the start or end of a frame.



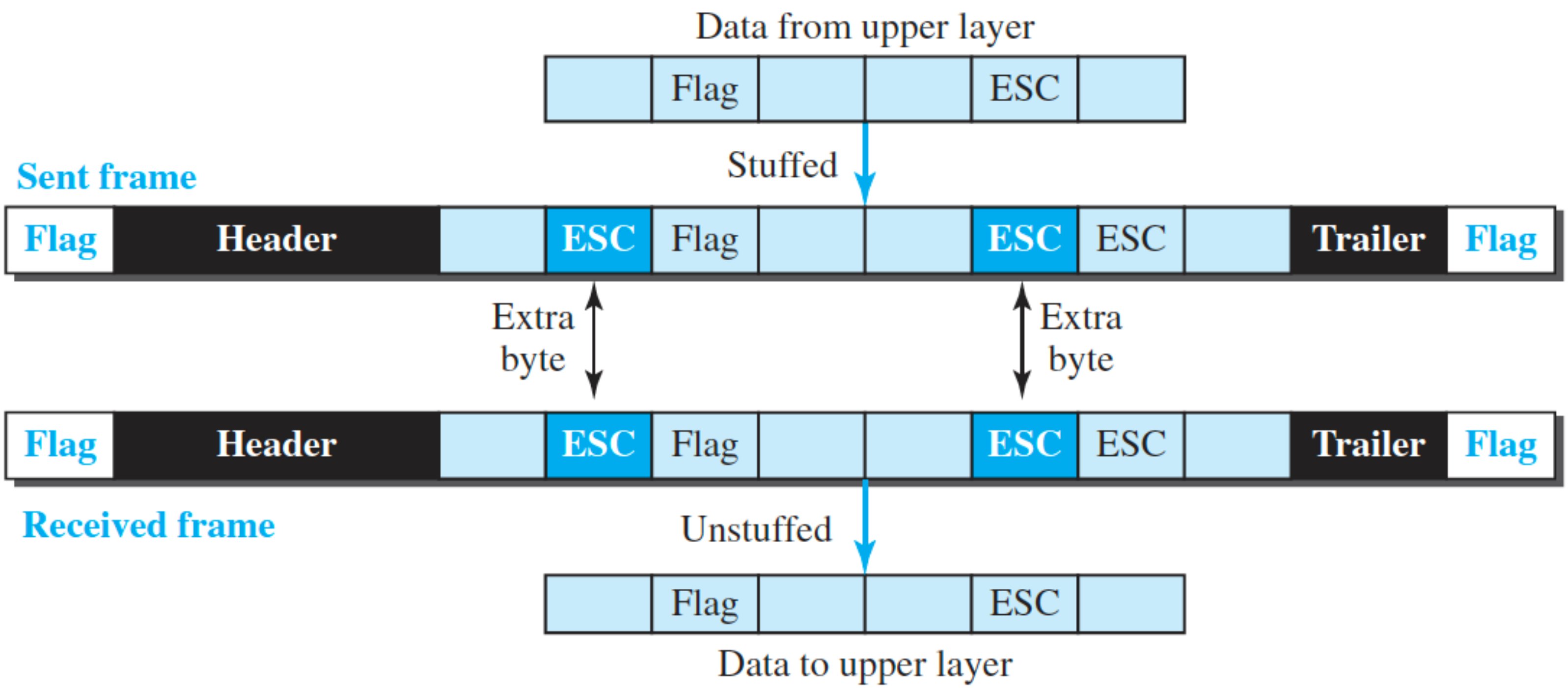
Character-Oriented Framing

- Character-oriented framing was popular when only text was exchanged by the data-link layers.
- The flag could be selected to be any character not used for text communication.
- Now, however, we send other types of information such as graphs, audio, and video; any character used for the flag could also be part of the information.
- If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame.
- To fix this problem, a byte-stuffing strategy was added to character-oriented framing.

Byte-Stuffing

- In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag.
- The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC) and has a predefined bit pattern.
- Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not as a delimiting flag.

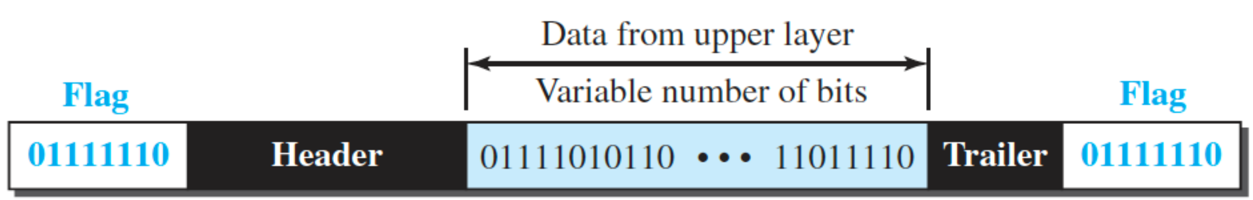
Byte-Stuffing and unstuffing



Byte stuffing is the process of adding one extra byte whenever there is a flag or escape character in the text.

Bit-Oriented Framing

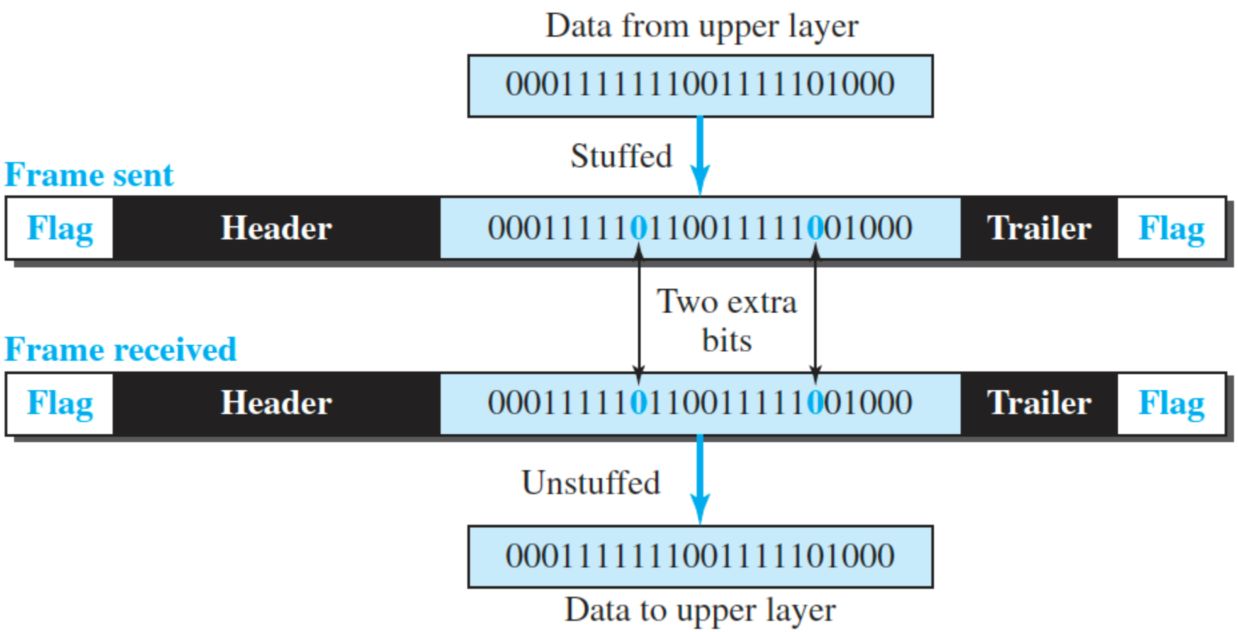
- In bit-oriented framing, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on.
- However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other.
- Most protocols use a special 8-bit pattern flag, 01111110, as the delimiter to define the beginning and the end of the frame.



Bit-Oriented Framing

- This flag can create the same type of problem we saw in the character-oriented protocols. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame.
- We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing.
- In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added.

Bit stuffing and unstuffing

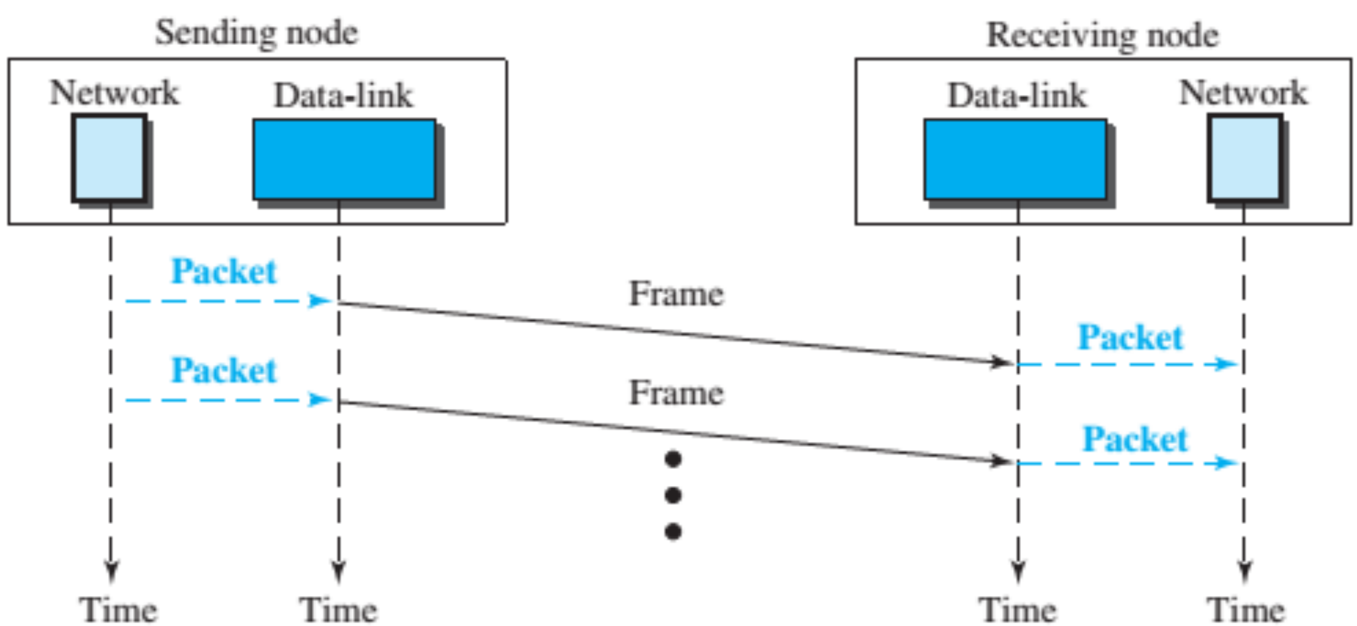


Data Link Layer Protocols

- To deal with flow and error control:
  - 1) Simple
  - 2) Stop-and-Wait

Simple Protocol

- The simple protocol has neither flow nor error control.
- We assume that the receiver can immediately handle any frame it receives. In other words, the receiver can never be overwhelmed with incoming frames.



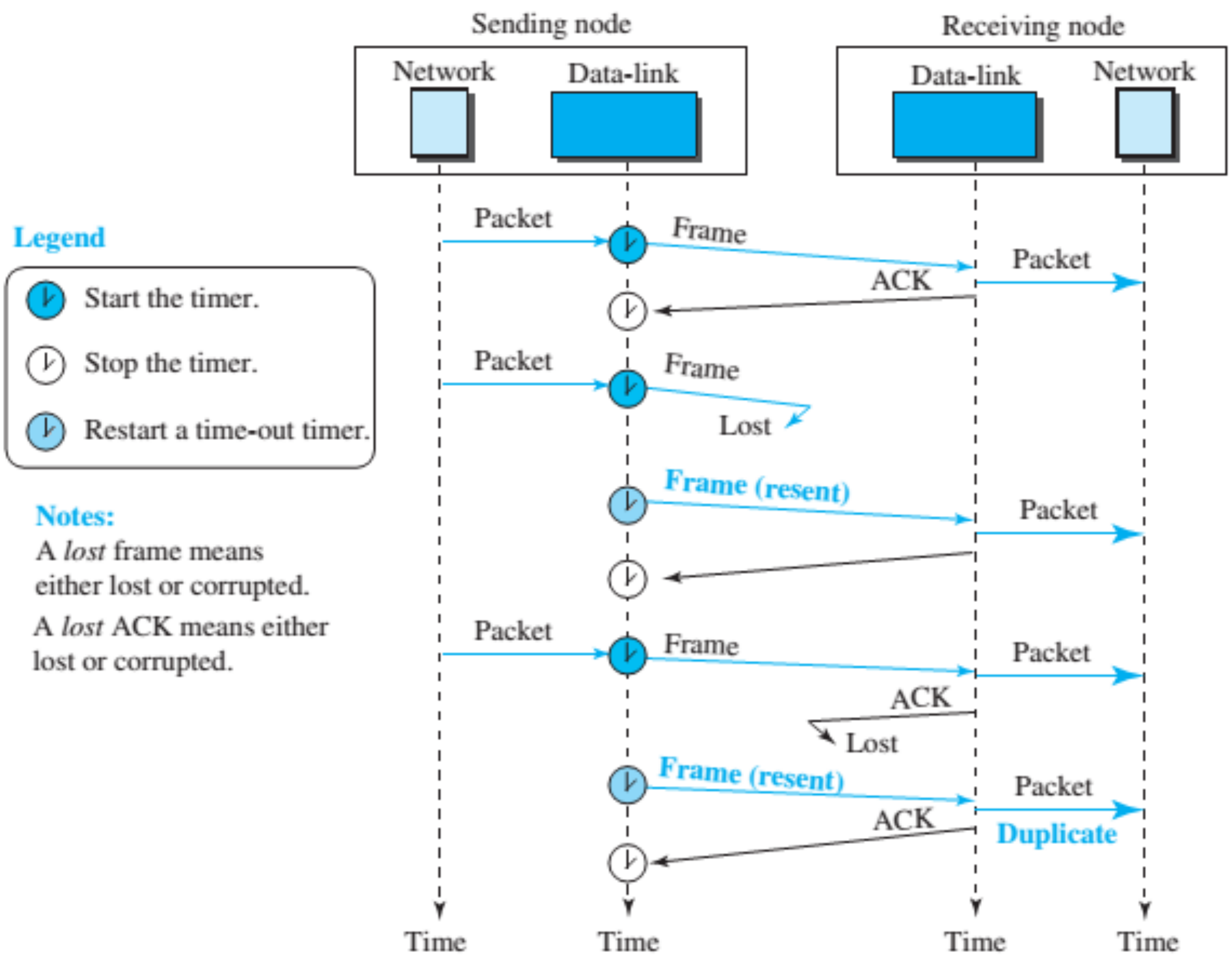
Stop and Wait Protocol

- Stop-and-Wait protocol uses both flow and error control.
- In this protocol, the sender sends one frame at a time and waits for an acknowledgment before sending the next one.
- To detect corrupted frames, we need to add a CRC to each data frame.
- When a frame arrives at the receiver site, it is checked. If its CRC is incorrect, the frame is corrupted and silently discarded.
- The silence of the receiver is a signal for the sender that a frame was either corrupted or lost.

Stop and Wait Protocol

- Every time the sender sends a frame, it starts a timer.
- If an acknowledgment arrives before the timer expires, the timer is stopped and and the sender sends the next frame (if it has one to send).
- If the timer expires, the sender resends the previous frame, assuming that the frame was either lost or corrupted.
- This means that the sender needs to keep a copy of the frame until
- When the corresponding acknowledgment arrives, the sender discards the copy and sends the next frame if it is ready.

Flow Diagram of Stop and Wait Protocol



Stop and Wait Protocol

- We saw a problem in previous example that needs to be addressed and corrected. Duplicate packets, as much as corrupted packets, need to be avoided.
- To correct the problem, we need to add *sequence numbers* to the data frames and *acknowledgment numbers* to the ACK frames.
- Sequence numbers are 0, 1, 0, 1, 0, 1, . . . ; the acknowledgment numbers can also be 1, 0, 1, 0, 1, 0, …
- In other words, the sequence numbers start with 0, the acknowledgment numbers start with 1.
- An acknowledgment number always defines the sequence number of the next frame to receive.

Stop and Wait Protocol with Sequence Number and Acknowledgment numbers