CT2108 – Nets and Comms 1

Data Link Layer

Content

- Design Issues
- Error detection and correction
- Data link in Internet
  - PPP

Data Link Layer Design Issues

- Services Provided to the Network Layer
- Framing
- Error Control
- Flow Control

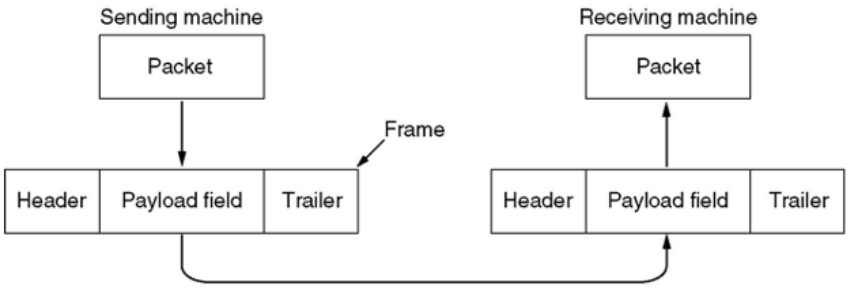
Functions of the Data Link Layer (1)

- Provide service interface to the network layer
- Dealing with transmission errors
- Regulating data flow
  - Slow receivers not swamped by fast senders

To accomplish those functions, data link layer takes packets it gets from the network layer and encapsulates them into **FRAMES** for transmission.

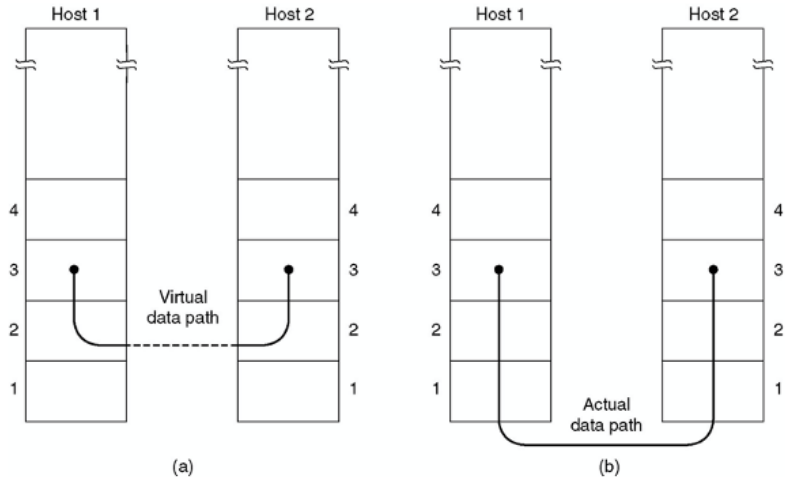
Functions of the Data Link Layer (2)

Relationship between packets and frames.



Each frame has a header, payload and a trailer. Frame management forms the heart of what the data link layer does.

Services Provided to Network Layer (1)



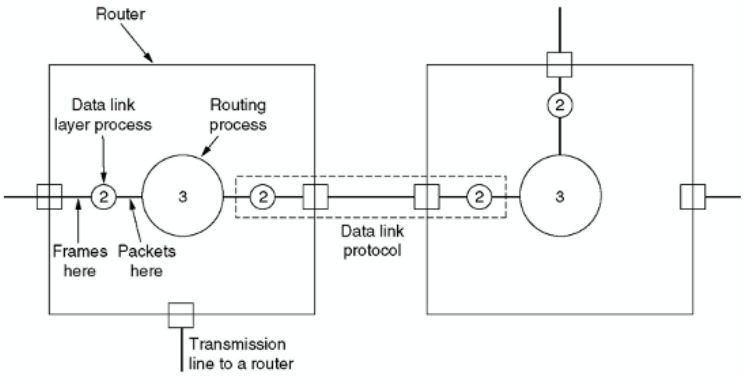
(a) Virtual communication.
(b) Actual communication.

The principal service that the data link layer has to provide to the network layer is to transfer packets from network layer of the source machine to the network layer on the destination machine. An entity on the source machine (process), at network layer is handing bits to the data link layer for transmission to the destination. The job of the data link layer is to transmit the bits to the destination machine so they can be handed over to the network layer there (a). The actual data follows the path described in (b). However, it is easier to think about two data link layer processes communicating using data link protocol.

Services Provided to Network Layer (2)

- Data link layer can provide the following types of services:
  - Unacknowledged connectionless service
    - no acknowledge by destination DLL
    - good for reliable services with a low error rate
    - speech where data loss is not important
  - Acknowledged connectionless service
    - all frames acknowledged individually
    - if there is no response within a specified time - resend
    - good for unreliable channels (radio vs optical fibre)
  - Acknowledged connection oriented service
    - most complex
    - most reliable
      - establish a connection
      - transmit and acknowledge numbered frames
    - three stages
      - set up link
      - transfer data
      - disconnect

Services Provided to Network Layer (3)

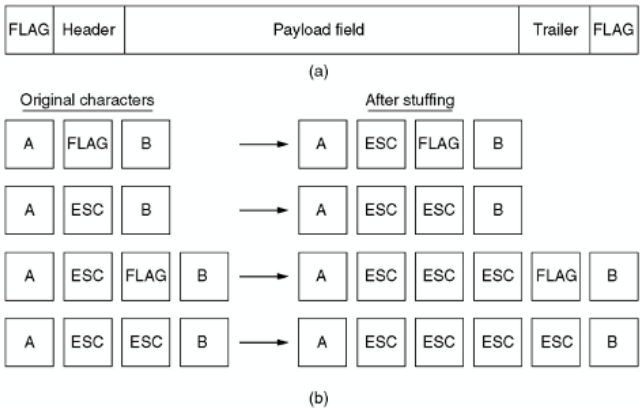


Placement of the data link protocol.

Consider a WAN subnet consisting of routers connected by point to point leased lines. When a frame arrives at the router, the hardware checks to see if the frame is error free, then passes the frame to the data link layer. The data link layer checks to see if that is the expected frame (frame number, sequence, etc..) and if so, it delivers the payload contained in the payload to the routing software (network layer). The routing software than chooses the appropriate outgoing line and passes the packet back down to the data link layer software, which transmits it.

It is up to the data link layer software to make sure that unreliable communication lines look perfect , or at least good to the network layer.

Framing (1)

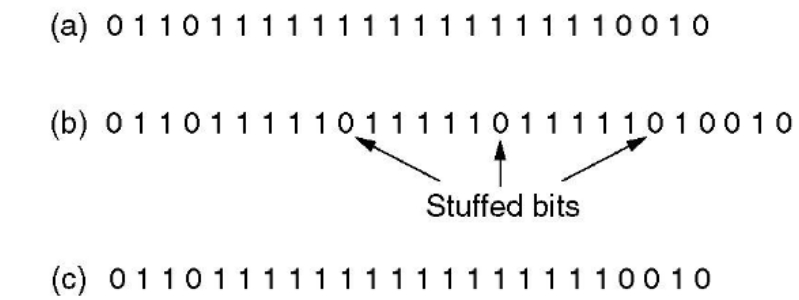


(a) A frame delimited by flag bytes.
(b) Four examples of byte sequences before and after stuffing.

Flag bytes with bye stuffing - this method solves the problem with resynchronization after an error by having each frame start and end with special bytes. A special character called FLAG BYTE is used to show the start and end of the frame. Two consecutive flag bytes indicate the end of one frame end beginning of next one. If the receiver is loosing sync, then it will look for the flag byte.

If binary data is sent, then the flag byte can appear within the data. One way to deal with this problem is to have the sender insert a special character called ESC just before each “accidental” occurrence of the FLAG byte. The data link layer on the receiving end will remove the ESC characters before delivering the payload to the network layer. This process is called BYTE STUFFING.

Framing (2)



Bit stuffing

(a) The original data.

(b) The data as they appear on the line.

(c) The data as they are stored in receiver's memory after destuffing.

The previous method didn't allow any arbitrary number of bits in the frame, but a multiple of 8 (bytes). To allow any number of bits in the frame, a different framing schema is used: bit stuffing.

Each frame begins and ends with a special bit pattern: 01111110.

Whenever data link layer on the sender encounters five consecutive 1's in the input bit stream, it will automatically insert (stuff) a 0 bit into the outgoing bit stream. On the receiver, when five consecutive 1's followed by a 0, automatically, 0 is removed (destuffed). The process is transparent to the network layer in both computers. If six 1's are received, that means the start or end of frame has been reached.

Usually a combination of the presented methods are used.

Error Control

- Frames may be received incorrectly so the receiver will provide some feedback to the sender
  - Acknowledge of receiving a frame
- A frame may vanish completely
  - Dealt with by introducing timers that could expire if no ACK is received, triggering retransmission
- One frame could be received two or more times
  - Assign frame sequence numbers to outgoing frames so the receiver can distinguish retransmissions from originals

Managing the timers and sequence numbers so as to ensure that each frame is passed to the network layer at the destination exactly once, no more and no less, is an important part of data link layer duties.

Flow Control

- Feedback based flow control
  - The receiver sends back information to the sender giving it permission to send more data
  - The sender is not allowed to send any data if the receiver doesn't allow it
- Rate based flow control
  - The sender has a built in mechanism that limits the speed at which the data is sent, without having a feedback from the receiver
  - i.e. the maximum amount of data sent in one second can be negotiated

Error Detection and Correction (1)

- Error-Detecting Codes
  - Include enough redundancy to allow the receiver to realize that an error has occurred (but not which error)
  - Typically, the receiver requests a retransmission
- Error-Correcting Codes
  - Include enough redundancy to allow the receiver to deduce what the transmitted data must have been
  - Called also ***forward error correction***

Transmission errors are present very often over analog loops (local loops). They are also present in the digital data, therefore mechanisms to deal with them have to be in place.

Error detecting codes are useful to use on channels with low probability of error (such as an optical fiber). This is because it is cheaper (in terms of wasted bandwidth) to resend now and then an frame, rather than including redundant information with every transmitted frame.

Forward error correction is more useful on transports that are exposed to errors (such as wireless transports).

Error Detection and Correction (2)

- A frame consists of
  - ***m*** data bits (message)
  - ***r*** redundant bits (check bits)
  - n= m +r is referred as n bit codeword
- For any two codewords, it is possible to determine how many bits differ (by XOR-ing bit wise the two codewords)

10001111
11111111 (XOR)

01110000

The number of bit positions in which two codewords differ is called ***HAMMING DISTANCE***

The significance of the hamming distance is that if two codeword are distance d apart, then it will require single bit errors to convert one into the other.

Error Detection and Correction (3)

- In most data transmission apps, 2^m messages are possible, but not all of 2^n possible codewords are used.
  - The two codewords whose Hamming distance is minimum give the Hamming distance of the complete code
- To detect d errors you need a d+1 distance code
  - Because with such distance there is no chance that d single bit errors will change one valid code into another valid code
- To correct d errors, you need a distance 2d+1 code
  - The legal codewords are so far apart that even with d changes, the original code word is still closer than any other codeword, so it can be uniquely determined

Error Detection and Correction (4)

- Error Detection – Parity
  - Single parity bit is appended to the data
  - If the number of the 1 bits in the codeword is even, than the parity bit is 0 (in even parity system)
  - A code with a single parity bit has a distance of 2, since any single bit error produces a codeword with the wrong parity
  - It can be used to detect single errors

1001 in even parity becomes 10010
1001 in odd parity becomes 10011

Error Detection and Correction (5)

- Error Correction Code Example
  - Consider a code with only four valid codewords:
    - 0000000000, 0000011111, 1111100000 and 1111111111
  - This code has a distance of 5, which means that it can correct double errors. If the codeword 00000000111 arrives, than the receiver will know that the original must have been 0000011111.
  - If a triple error changes original codeword 0000000000 into 0000000111 than the error will not be corrected properly

Error Detecting Codes (1)

- Adding parity bits to frames
  - OK for single bit errors
  - If a single, long burst of noise would garble the frame, then we have a probability of 0.5 to detect the errors, which is not acceptable
- Organize the frame in a matrix with m lines by k columns, and append a parity bit every column, sending every line
  - The parity bit is computed separately for each column and affixed to the matrix as a last row
  - Matrix is transmitted one row at the time
  - This method can detect burst errors of length m (if a m+1 burst error occurs, and only first and last bits are changed, it will go undetected)

Consider a channel whose error rate is 10 power -6 per bit (one error bit every 1 million bits). Let the block size be 1000 bits and assume that we could have a hamming code with 10 check bits (meaning that every frame will contain 1010 bits instead of 1000 bits). To detect that one frame had an error out of 1000 frames (to achieve 1 megabit of data, so the error will occur), we had to send 10000 bits of redundant data.

To detect that one frame had an error, it would have been enough to actually append 1 bit of parity to each frame. Every 1000 frames, one 1001 bit frame would need to be retransmitted. The extra traffic in this case would be 2001 bits (when an error detection schema would be used) versus about 10000 bits (when an error detection schema would be used).

A burst error doesn't imply necessarily that all the bits are wrong. It just implies that at least some of the bits are wrong.

Error Detecting Codes (2)

- ***CRC*** (***C***yclic ***R***edundancy ***C***heck) or known also as ***polynomial code***
  - Bit strings are treated as representations of polynomials with coefficients of 0 and 1 only
  - For instance frame 110001 is represented by polynomial: $X^5 + X^4 + 1$
  - Polynomial arithmetic is done modulo 2. There are no carries for addition nor borrows for subtraction. Both addition and subtraction are identical to a XOR operation
  - Division is carried out the same way as in binary, except that the subtraction is done modulo 2

Error Detecting Codes (3)

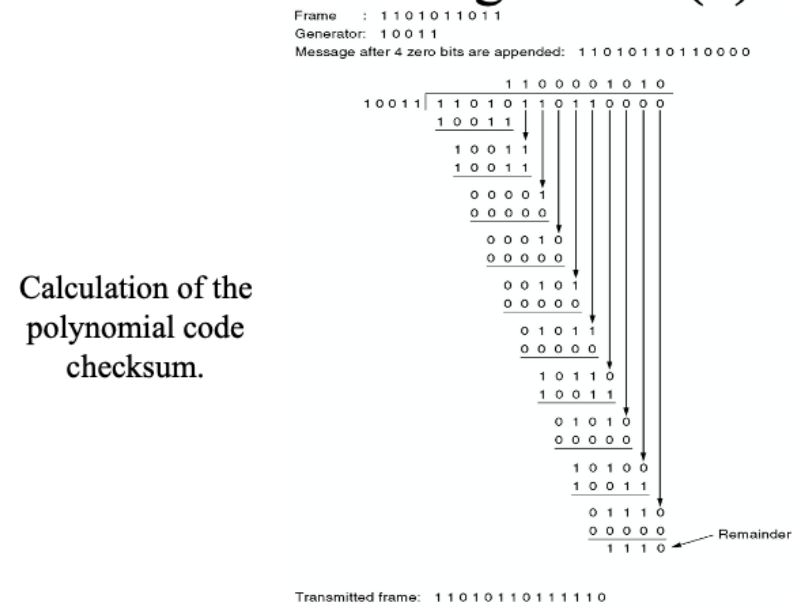
- CRC
  - The sender and the receiver must agree on a generator polynomial G(x), in advance – both low and high order bits of the generator should be 1
  - For a frame with m bits, with corresponding polynomial M(x), the frame must be longer than the generator polynomial
  - The idea is to append a checksum to the end of the frame in such a way that the polynomial represented by the checksummed frame is divisible by the G(x)
  - At the other end, the receiver would divide the received frame (checksummed) by agreed G(x) and if the remainder is not zero, then an error condition occurred

Error Detecting Codes (4)

- CRC algorithm
  - r is the degree of G(x). Append r zero bits to the frame, so now it contains m+r bits and corresponds to the polynomial $x^r M(x)$
  - Divide the bit string corresponding to $x^r M(x)$ to the bit string corresponding to G(x), using modulo 2 division
  - Subtract the reminder from the bit string corresponding to $x^r M(x)$ using modulo 2 subtraction. The result is the checksummed frame to be sent, with corresponding polynomial T(x)

In base ten, if you divide 5432 to 1000, you obtain the reminder 432. It is obvious that if you subtract 432 from 5432 you obtain a result that is divisible by your divisor (1000). Exactly the same idea is used in binary to obtain the checksummed frame T(x) that should be divisible by G(x).

Error-Detecting Codes (5)

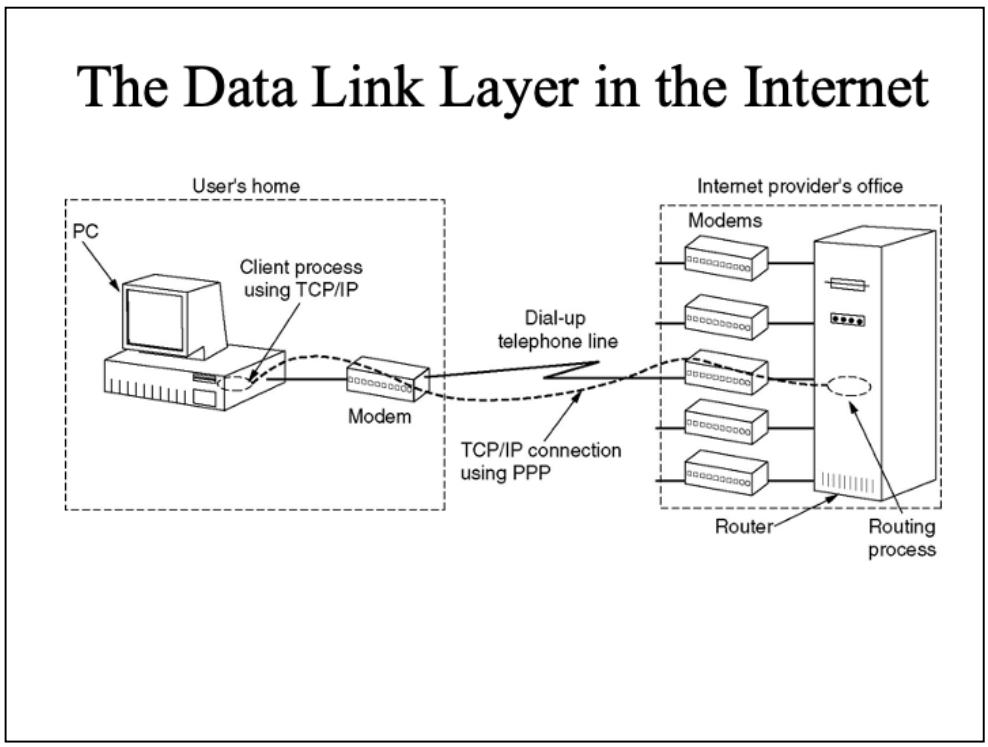


Calculation of the polynomial code checksum.

If a transmission error occurs, then instead of T(x), at the receiver we will have T(x) + E(x). Each 1 in E(x) corresponds to a bit that has been inverted. If there are k bits in E(x) than there will be k single bit errors.

G(x) is chosen to be a large and prime polynomial to be able to catch as many errors as possible.

Certain polynomials have become standards.



Internet consists of individual machines (hosts and routers) and the communication infrastructure that connects them. Some of the machines are interconnected using LANs and some are interconnected using point to point lines (especially the ones that are far apart).

PPP - Point to Point Protocol (1)

- RFC1661, RFC1662 and RFC 1663
- Provides three features
  - A framing method; the frame format also handles error detection
  - A link control protocol called LCP (Link Control Protocol)
  - A way to negotiate network specific options in a way that is independent of the network layer protocol to be used. The method chosen is to have a different NCP (Network Control Protocol) for each network layer supported

PPP handles error detection, supports multiple protocols, allows IP addresses to be negotiated at connection time, permits authentication (using two methods – PAP and CHAP)