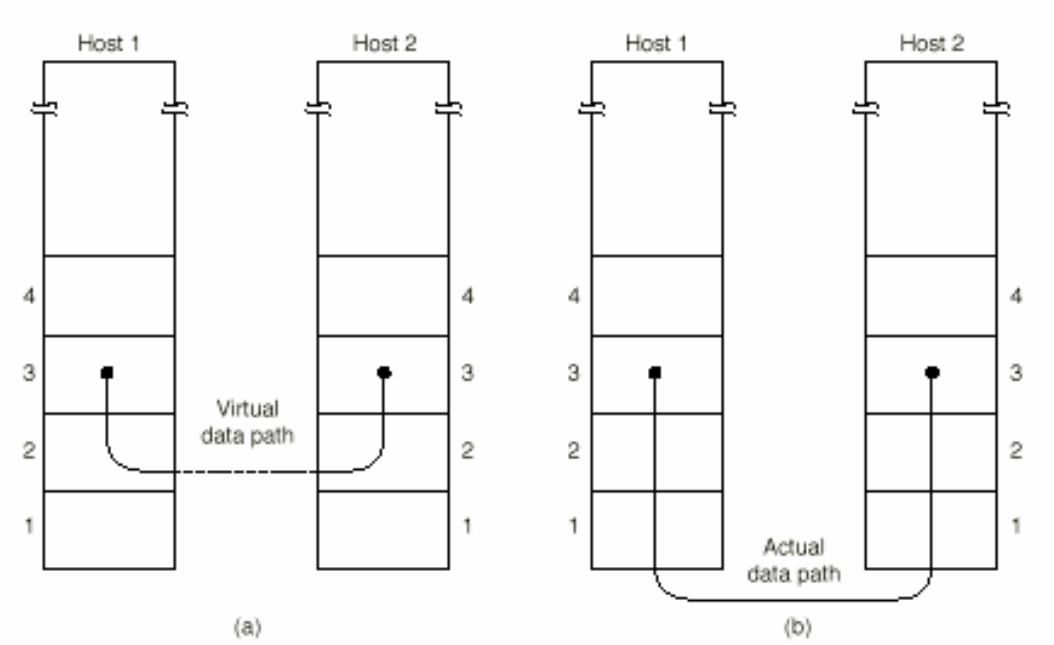
Data Link Layer(1)

Principal service:

Transferring data from the network layer of the source machine to the one of the destination machine

Virtual communication versus actual communication:



Specific functions to carry out:

- determining how the bits of the physical layer are grouped into frames
- dealing with transmission errors
- providing flow control

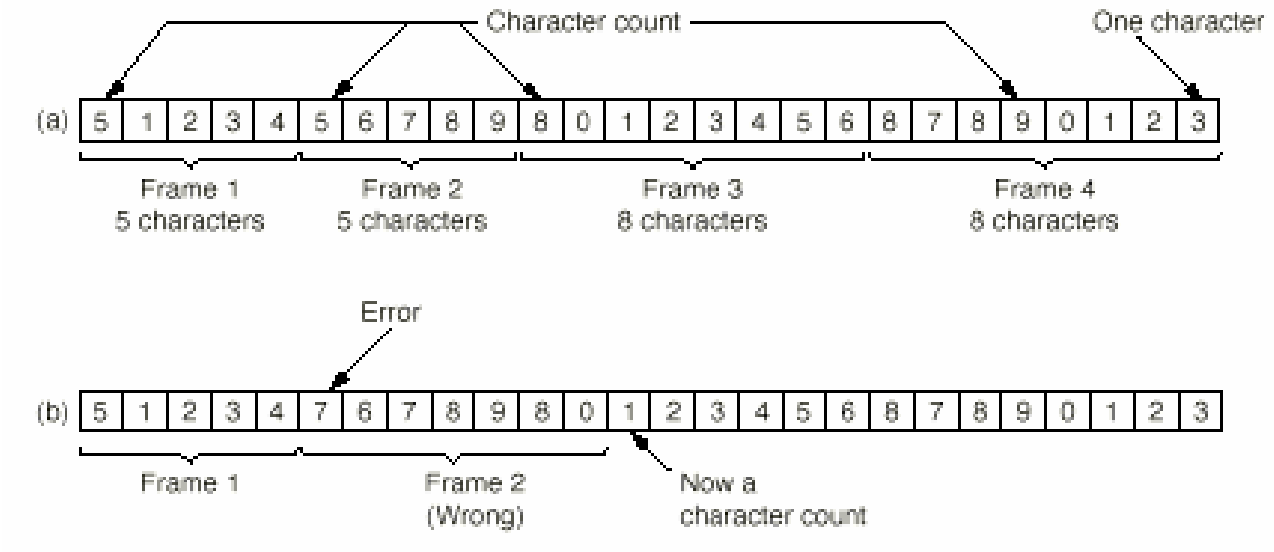
Data Link Layer(2)

Grouping into frames

Methods for framing:

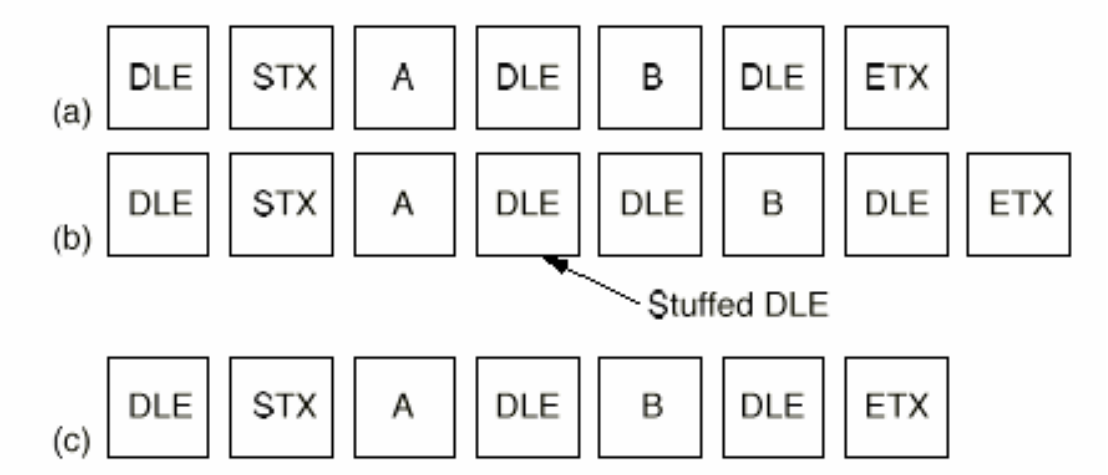
- character count
- character stuffing
- bit stuffing
- exploiting redundancy in the physical layer

Example for character count:

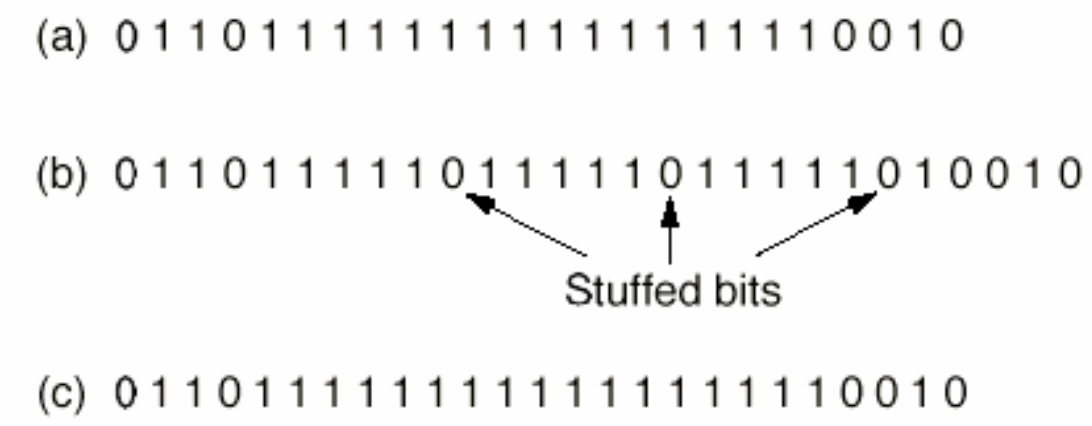


Data Link Layer(3)

Example for character stuffing:



Example for bit stuffing:



Data Link Layer(4)

Dealing with transmission errors

Error Detecting Codes (EDC) and Error Correcting Codes (ECC)

Definitions:

Codeword:= source (original, payload) word + (redundant) check (control) bits

m:= length of the source word (number of information bits)

r:= number of check bits

n:= m + r := length of the codeword --->

A binary code is a subset of R^2_n . Ist elements (words) also can be considered as code vectors. /

Use of ECCs :

•Heavily disturbed transmission channels

•Data transmission via wireless transmitters (insbesondere bei militärischen Anwendungen)

•Secondary storage media, to correct reading errors without repeating the reading procedure

Hamming distance:

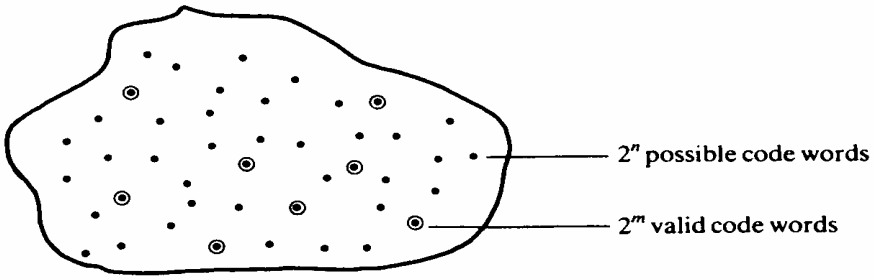
Let x and y be codewords in R^2_n . The function

$d(x,y) := \sum x_i + y_i$ witht i= 1,.....,n is called Hamming distance of x and y.

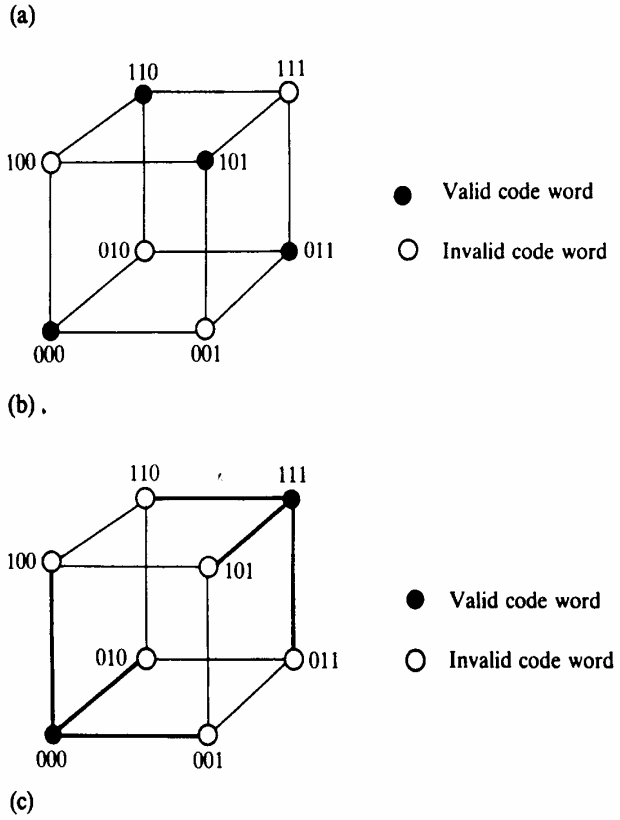
The Hamming distance of the entire code (set of all correct codewords) is defined as the minimal Hamming distance between any 2 codewords of this code.

Data Link Layer(4a)

Illustration of the principle of Error-Correcting Codes (ECC´s) and Error-Detecting Codes (EDC´s):



3-dimensional representation of codes:



Data Link Layer(5)

Lower limits on the number of check bits needed to correct single errors depending on the word size:

Each of the 2^m legal codewords has n illegal codewords at a distance 1 dedicated to it --->

$(n+1)\ 2^m <= 2^n$. Using $n = m+ r$ --->

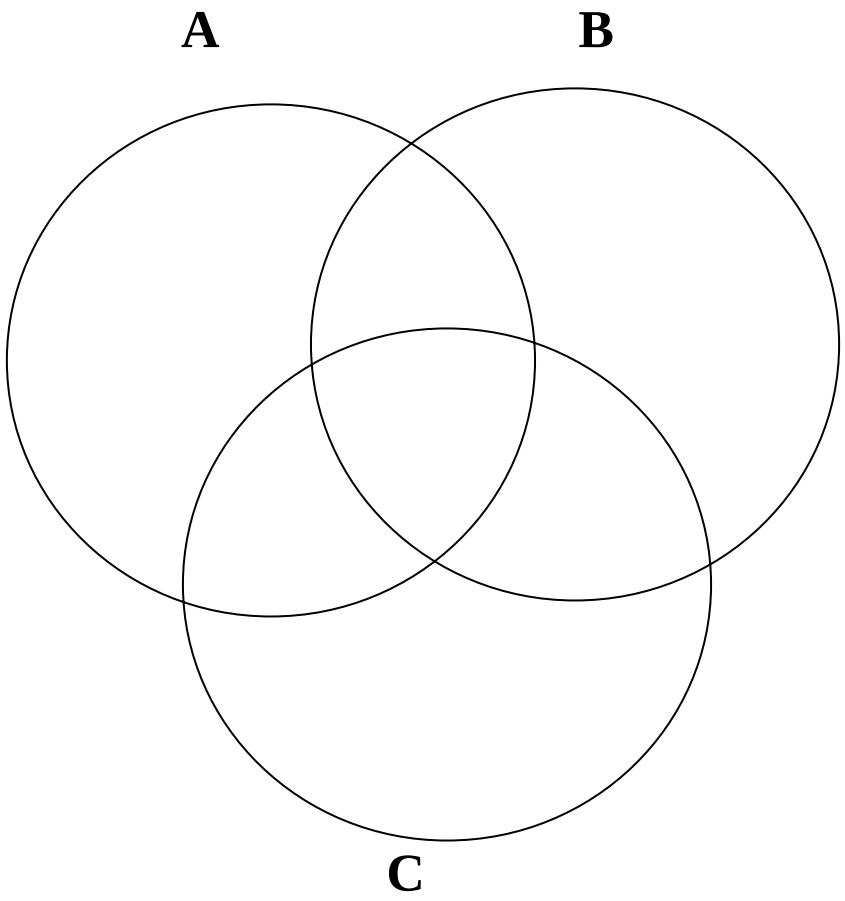
$(m+ r +1) <= 2^r$

Some numbers for r depending on m:

Word size	Check bits	Total size	Percent overhead
8	4	12	50
16	5	21	31
32	6	38	19
64	7	71	11
128	8	136	6
256	9	265	4
512	10	522	2

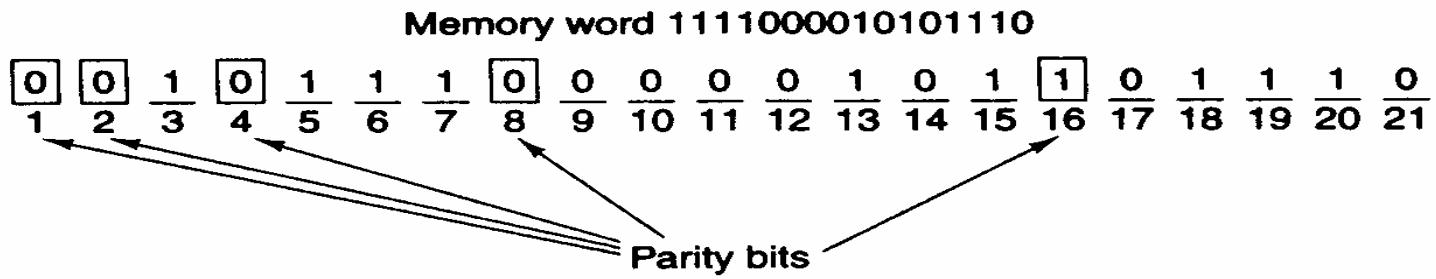
Data Link Layer(6)

Principal idea of the Hamming - Code:



Data Link Layer(7)

Example:



parity bit 1 **wrong** (1,3,5,7,9,11,13,15,17,19,21 contain altogether **5** Ones)

parity bit 2 **correct** (2,3,6,7,10,11,14,15,18,19, contain altogether **6** Ones)

parity bit 4 **wrong** (4,5,6,7,12,13,14,15,20,21 contain altogether **5** Ones)

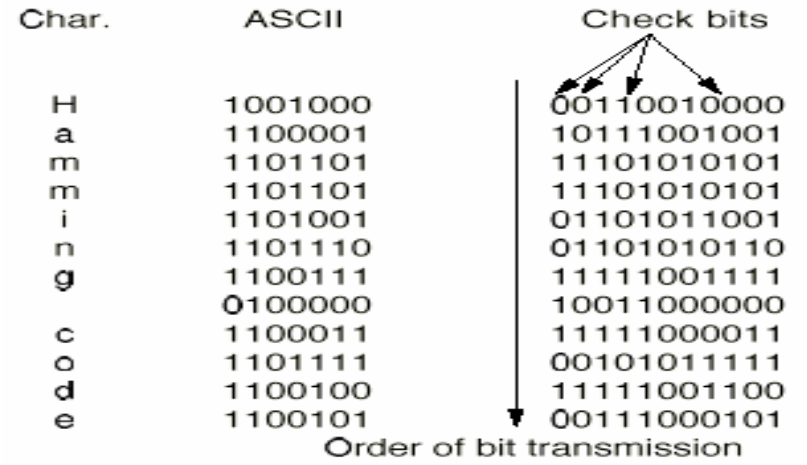
parity bit 8 **correct** (8,9,10,11,12,13,14,15 contain altogether **2** Ones)

parity bit 16 **correct** (16,17,18,19,20,21 contain altogether **4** Ones)

---> Bit No. 5 is wrong and has to be inverted!

Data Link Layer(8)

Use of the Hamming - Code to correct burst errors:



Parity code with odd and even parities:

Message	Code word (even parity)	Code word (odd parity)
000	0000	1000
001	1001	0001
010	1010	0010
011	0011	1011
100	1100	0100
101	0101	1101
110	0110	1110
111	1111	0111
	↑ Even parity bit	↑ Odd parity bit

Data Link Layer(10)

Idea of the Polynomial or Cyclic Redundancy Code (CRC):

- Treat bit strings as representations of polynomials with coefficients of 0 and 1 only.
- Sender and receiver must agree upon a **generator polynomial *G(x)***
- The sender computes the checksummed polynomial *T(x)* to be sent such that *T(x)* is divisible by *G(x)*
- The receiver divides the received *T(x)* by *G(x)*. If there is a remainder ---> **Error!!!**

Algorithm for computing the checksum:

1. Let *r* be the degree of *G*(*x*). Append r zero bits to the low-order end of the frame, so it now contains *m*+*r* bits and corresponds to the polynomial $x^r M(x)$.

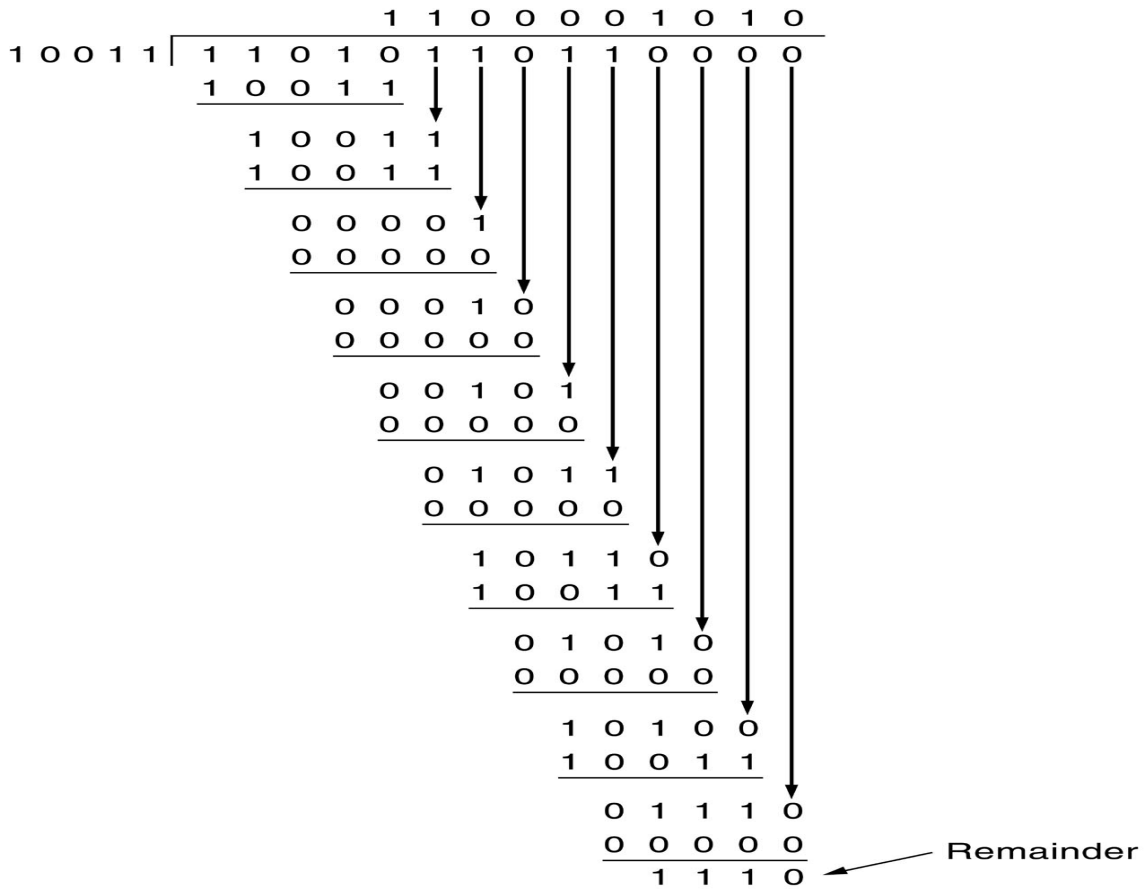
2. Divide the bit string corresponding to *G(x)* into the bit string corresponding to $x^r M(x)$ using modulo 2 division.

3. Subtract the remainder (which is always r or fewer bits) from the bit string corresponding to $x^r M(x)$ using modulo 2 subtraction. The result is the checksummed frame to be transmitted. Call its polynomial *T(x)*.

Data Link Layer(11)

Example:

Frame : 1 1 0 1 0 1 1 0 1 1
Generator: 1 0 0 1 1
Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0



Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

Data Link Layer(12)

Three (generator) polynomials have become international standards:

CRC-12 $= x^{12}+x^{11}+x^3+x^2+x^1+1$

CRC-16 $= x^{16}+x^{15}+x^2+1$

CRC-CCITT $= x^{16}+x^{12}+x^5+1$

Properties of a 16-bit checksum (such as CRC-16 and CRC-CCITT):

It catches

- all single and double errors
- all errors with an odd number of bits
- all burst errors of length 16 or less
- 99.997% of 17-bit error bursts
- 99.998% of 18-bit and longer bursts

Data Link Layer(13)

Providing Flow Control

A Simplex Stop-and-Wait Protocol

Assumptions:

- No infinite amount of buffer space available on the receiver's side
- Communication channel is assumed to be error-free and data traffic is simplex

Problem:

How to prevent the sender from flooding the receiver with data faster than the latter is able to process

Solution approach:

Having the receiver provide feedback to the sender by acknowledging each frame sent

---> the sender must wait until an *ack* frame arrives before fetching the next frame from the network layer

Remark:

Data traffic is simplex, but frames do travel in both directions (bidirectional information transfer)

---> the physical communication channel must be duplex or at least half-duplex

Piggybacking (if data traffic is duplex)**:**

Delaying outgoing *acks* so that they can be hooked on the next outgoing data frame (having a separate *ack* field in its header)

Principal advantage over having separate ack frames:

Better use of the available channel bandwidth

Data Link Layer(14)

Sliding window protocols

Assumptions:

- Data traffic is duplex
- Sender`s data frame could be lost or garbled ---> time-out mechanism for the sender

Problem:

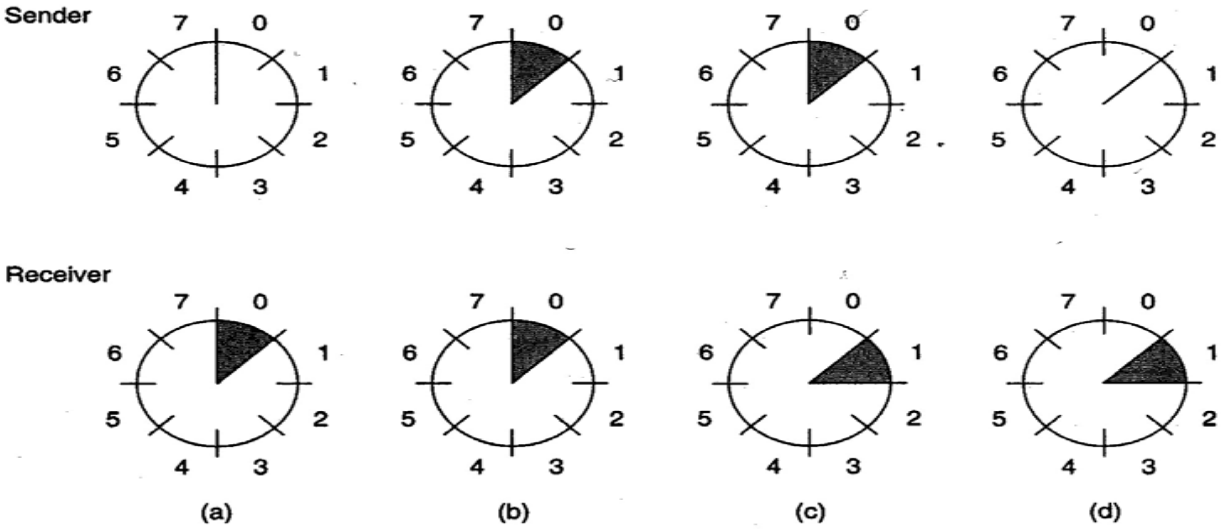
---> How long is timeout? ---> Early time-out ---> synchronization problem!

Solution approach:

Having sliding window protocols were each outbound frame contains a sequence number

---> sender (receiver) maintains a set of sequence numbers up to some maximum 2^n-1, called sending (receiving) window, corresponding to frames it is permitted to send (accept)

Illustration of the main idea of sliding windows (maximum window size = 1, n = 3 ---> 3-bit sequence number)

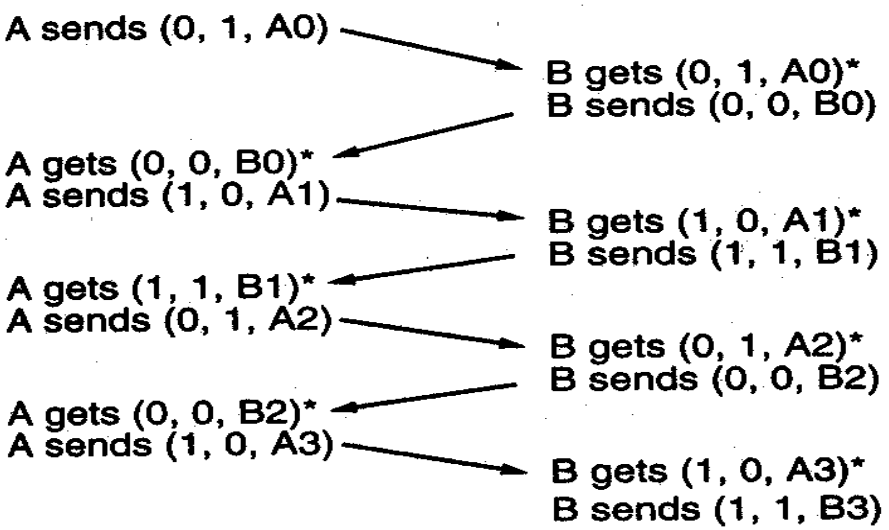


Data Link Layer(15)

One Bit Sliding Window Protocol (Stop-and-Wait)

- Ack field contains the number of the last frame received without error
- Agreement with the number of the frame being sent ---> fetch the next frame from the network layer
- No agreement ---> continue trying to send the same frame
- Whenever a frame is received, a frame is also sent back

Example



Resulting Quality of Service

- No delivering of duplicate packets to either network layer
- No packet is skipped
- No deadlock

Data Link Layer(16)

Implicit assumption so far:

Transmission times for a data frame + its ack are negligible

If this assumption is false ---> exploitation of the bandwidth may be disastrous

---> In such a case, requiring a sender to wait for an ack before sending the next frame must be relaxed

Pipelining

The sender is allowed to transmit up to *w* frames before blocking, instead of just 1.

Remaining problem:

What happens if a frame is lost or damaged in the middle of a long stream of transmitted frames?

A protocol using go back n

All frames arriving after an erroneous one are simply discarded

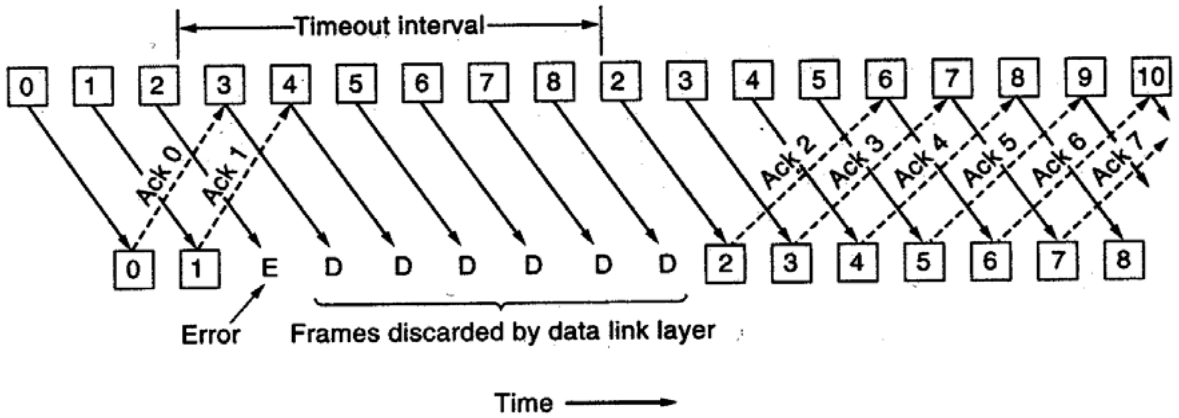
---> the data link layer refuses to accept any frame except the next one to be delivered to the network layer

---> eventually, the sender will time out and retransmit all unacknowledged frames in order starting with the erroneous one

This strategy corresponds to a receive window of size 1.

Data Link Layer(17)

Example:

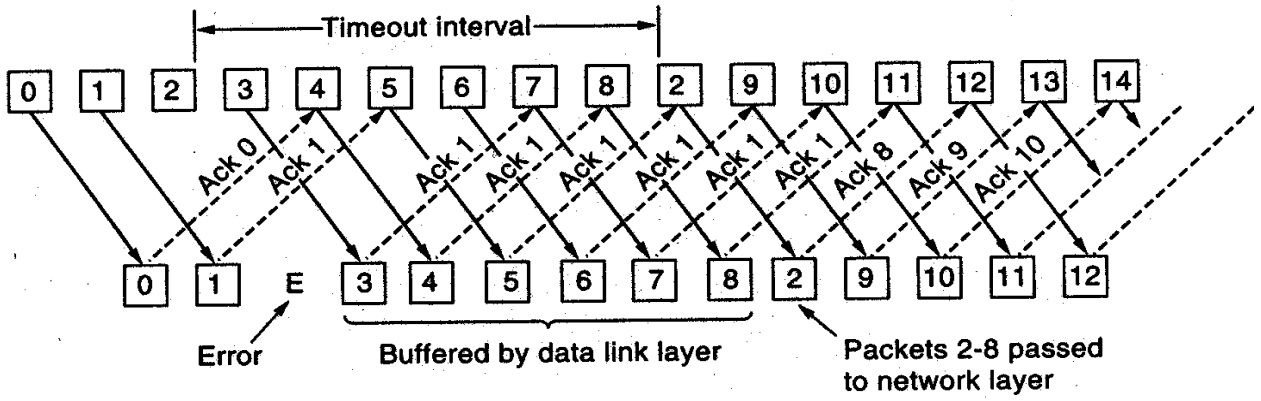


Main drawback: It can waste a lot of bandwidth if the error rate is high

A protocol using selective repeat

All correct frames arriving after an erroneous one are accepted by the receiver

Example:



This strategy corresponds to a receive window of size > 1.

Main drawback: It can require large amounts of data link layer buffer space

Data Link Layer(18)

Pipelining implies multiple outstanding frames.

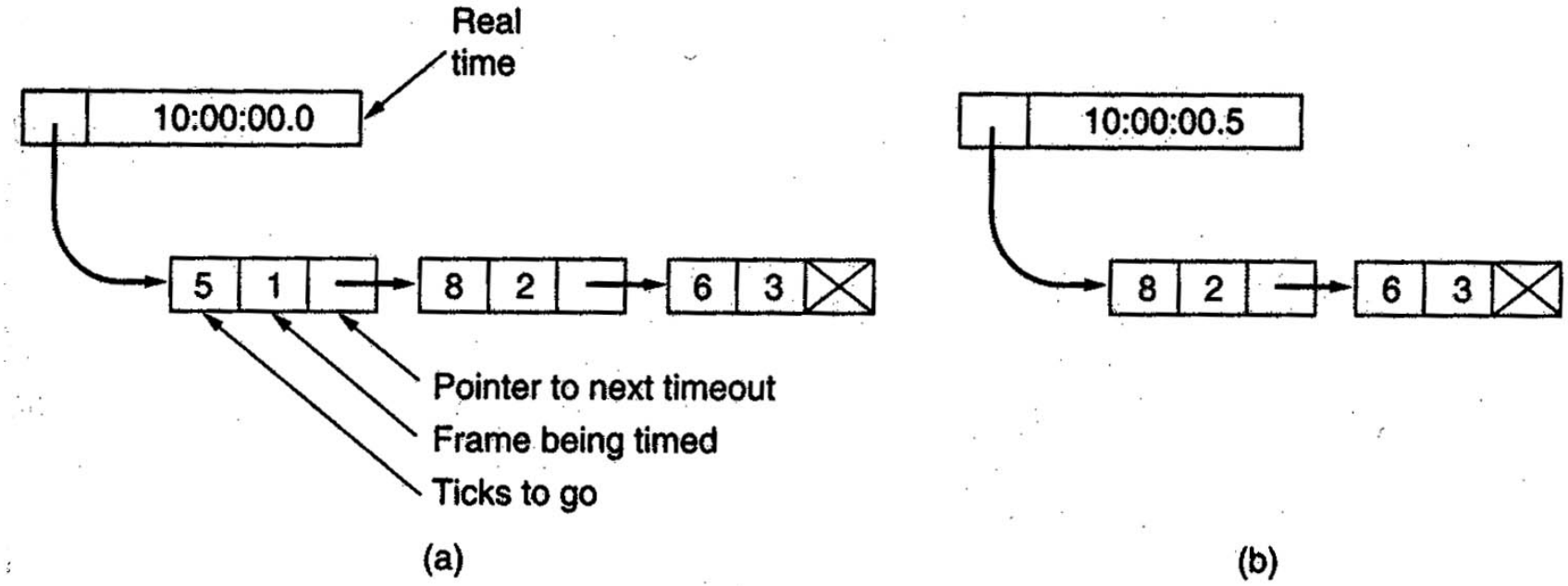
---> Each frame times out independent of all the other ones

---> It logically needs multiple timers

Simulation of multiple timers in software using a single hardware clock

The pending timeouts form a linked list

Example:

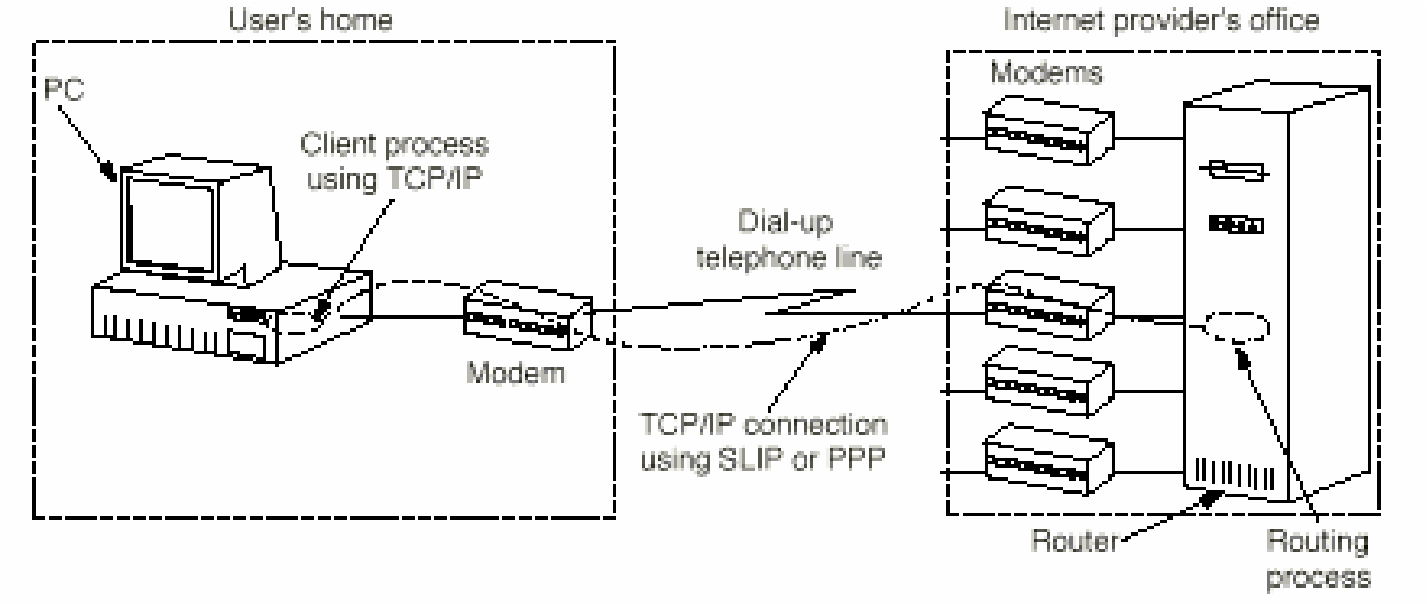


Data Link Layer(13)

Internet protocols for the Data Link Layer

Two protocols, SLIP and PPP, are widely used in the Internet as point-to-point data link protocols.

Typical application example: A home PC acting as an Internet host



Data Link Layer(14)

The Serial Line Internet Protocol (SLIP):

- Designed in 1984 to connect SUN workstations to the Internet over a dial-up line using a modem.
- It is very simple:
  - sends raw IP packets over the line with a special flag byte at the end for framing
  - uses some form of character stuffing

- Drawbacks:
  - does not do any error detection or correction
  - supports only IP
  - each side must know the other´s IP address a priori
  - does not provide any form of authentication
  - no approved Internet Standard

Data Link Layer(15)

The Point-to-Point Protocol (PPP):

PPP basically provides three things:

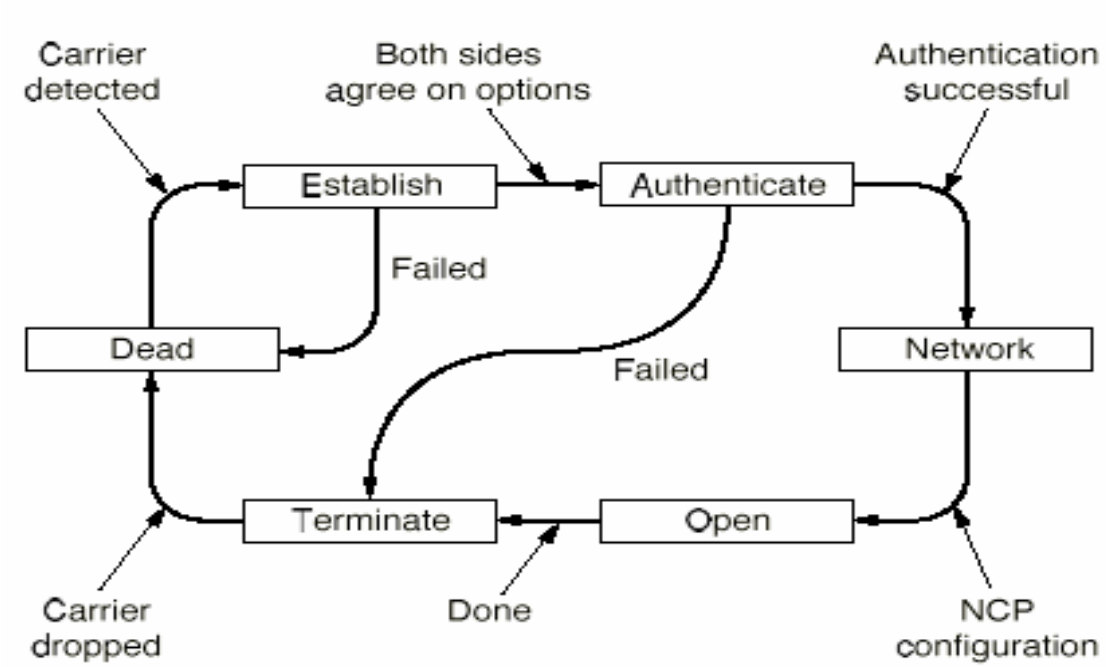
1. A framing method that unambiguously delineates the end of one frame and the start of the next one. The frame format also handles error detection.

2. A link control protocol for bringing lines up, testing them, negotiating options, and bringing them down again gracefully when they are no longer needed. This protocol is called **LCP** (**Link Control Protocol**).

3. A way to negotiate network-layer 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.

Data Link Layer(16)

A simplified phase diagram for bringing a line up and down:



The PPP full frame format for unnumbered mode of operation: