

CSIS 625 Week 5

Error Detection and Correction, Data Link Layer. HDLC

Copyright 2001, 2002, 2003 – Daniel R. Oelke

For use by students of CSIS 625 for purposes of this class only.

I. Overview

A. Error Control

1. Error Detection
2. Error Correction

B. Data Link Layer

1. Line Discipline
2. Flow Control
3. Error Control

C. Byte Oriented Link Control

D. Bit Oriented Link Control

1. HDLC

II. Error Detection

A. Errors always occur during transmission

1. Point of Error Detection is to detect these errors so we can correct them.
2. Some systems fix these errors by re-transmission.
3. Some systems fix these errors using Forward Error Correction (FEC)

B. Vocabulary

1. Bit Error - an error that changes only one bit
2. Burst Error - an error that changes several adjacent bits.
3. Coding Violations - When the line coding mechanism tells us of an error.
 - a. Need to use something like Bipolar-AMI, 8b10b, etc.

C. Error Detection - Parity Bits

1. Parity Bit - one extra bit of data sent with every data packet.
 - a. Even Parity - The extra bit is set so that the number of bits is always even
 - b. Odd Parity - The extra bit is set so that the number of bits is always odd
 - c. Receiver checks parity and discards data if parity is not valid
 - d. Regular parity is also known as Vertical Redundancy Check
 - e. Parity term is sometimes mis-used to refer to any redundant bit

D. Error Detection - BIP

1. BIP-8 - Bit Interleaved Parity - 8 bits
 - a. BIP is also known as Longitudinal Redundancy Check
 - b. Takes bytes and calculates a parity bit for each bit position
 - c. Provides better detection for burst errors.
2. It is technically possible to have BIP-16 or BIP-32
 - a. Not commonly seen
 - b. BIP-8 is by far the most common system.

E. Error Detection - Checksum

1. Checksum - a byte added to the end of the frame to catch errors
2. Simplest form is calculated by adding up all the bytes in the frame
3. There are several algorithms - so check specifics on any one algorithm

4. BIP-8 is sometimes called a checksum
 - a. Start with 0x00 and do bitwise XOR of every byte in the message
5. Checksums do not catch all kinds of errors.
 - a. For example: They don't catch blocks of data being re-arranged.

F. Error Detection - CRC

1. CRC - Cyclic Redundancy Check
2. Catches many errors that Parity or BIP-8 will miss.
3. Adds bits to the end of a frame so that it can be evenly divided by a number
4. Usually 8, 16, or 32-bit CRC
5. Easy to implement in hardware with shift register and feedback xor's.
6. See book for examples
7. CRC - Strength
 - a. CRC's are considered quite strong
 - b. All bursts of errors $\leq r$ ($r = \text{length of CRC}$)
 - c. All odd number of errors
 - d. Probability of missing error
 - (1) if burst is $r+1 \Rightarrow 0.5^{r-1}$ ($0.5^{32-1} = 4.6E-10$)
 - (2) if burst is $> r+1 \Rightarrow 0.5^r$ ($0.5^{32} = 2.3E-10$)
8. See also: <http://www.ross.net/crc/>

G. Error Rates - BER

1. BER - Bit Error Rate - the probability of a bit being changed
 - a. Used to describe transmission lines
 - b. All transmission lines have some non-zero BER
 - c. Calculate by counting the number of errors and dividing by the number of bits through the system.
 - (1) Need to adjust for errors that occur but are not detected
 - (2) For example - an odd number of errors when using parity
2. Normal practice is to measure 10 times the period to report a BER
 - a. This means that to state the line has a $1E-10$ error rate (or better) you must measure $1E+11$ bits
 - b. A T1 would require 18 hours to have $1E+11$ bits go through.
3. BER and real transmission systems
 - a. Voice sounds ok at BER of $1E-5$
 - b. Voice till intelligible at BER of $1E-3$
 - c. Data doesn't work very well at BER $1E-7$ or higher (that is $1E-6$, $1E-5$, etc.)
 - (1) Retransmission of packets occurs so often that it is difficult to get a very high data rate.

III. Error Correction

A. FEC - Forward Error Correction

1. FEC – A method by which errors in received data can be corrected without requiring retransmission
2. Sometimes it is advantageous to correct an error instead of just detecting it.
 - a. May take too long for retransmission to occur
 - (1) Real-time Data may not be able to wait for the round trip time
 - b. May not be possible to ask for retransmission
 - (1) One way transmissions – Satellite, Broadcast data, etc.
3. Brute force method:

- a. Send data repeatedly
 - b. To correct n errors, send data $2n+1$ times
 - c. Not very efficient - there are better ways.
 - d. This is almost never used in real systems.
4. Hamming codes or Reed-Solomon codes do it much more efficiently

B. Hamming Distance

1. Hamming distance - the number of bits that have to be changed to go from one valid code to another.
 - a. To detect d errors - Hamming distance of $d+1$
 - (1) parity gives a Hamming distance of 2, so it detects up to one bit error
 - b. To correct d errors - Hamming distance of $2d+1$
 - (1) Need to be able to distinguish "closest" valid code

C. Hamming Code

1. A symbol is a block of bits to be transmitted from source to destination.
2. Hamming Code - an error correcting code to correct 1 bit error in a symbol
 - a. For 4 data bits, 3 redundancy bits needed
 - b. For 8 data bits, 5 redundancy bits needed
3. Hamming Codes may also be used to detect up to 2 bit errors in the code word.
4. Rule for Hamming Codes:
 - a. $(d + p + 1) \leq 2^p$
 - (1) d = message bits (data)
 - (2) p = redundant bits (parity)
5. Codes often notated as (c,d)
 - a. $c = d + p$
 - b. Therefore a $(12,8)$ code has 8 data bits and 4 parity bits.
 - (1) A 50% overhead.
 - (2) $(12 + 1) < 2^4$
6. To handle burst errors, multiple codes can be interleaved.
 - a. Errors normally occur in bursts.
 - b. Hamming code can handle only one bit error in the code word.
 - c. Solution is to spread out the bits so that more than one won't be hit by a burst of errors.
7. Perfect code – an error correcting code where all possible codes are within hamming distance of a symbol.
 - a. No "wasted" space in the code.
 - b. Most efficient code that is possible for that symbol size, and code size.
8. For Hamming codes a perfect code is when there is equality in the Hamming Code equation above
 - a. $(7,4)$ code is an example – $m = 4, r = 3$
 - b. $(4 + 3 + 1) = 2^3$

D. Reed-Solomon codes

1. Another error correcting code
2. Designed to have redundant symbols
3. Allows for whole symbols to be errored
 - a. It handles bursts of errors
4. Common applications
 - a. CD Players,

- b. Spacecraft communication
 - c. DSL lines
- 5. Notation - RS(NN, KK)
 - a. MM - the code symbol size in bits
 - b. NN - the block size in symbols ($2^{MM} - 1$)
 - c. KK - the number of data symbols per block
 - (1) $KK < NN$
- 6. Can correct $(NN - KK)/2$ errors per block
- 7. If known that there are missing symbols, then RS can correct $NN - KK$ "erasures"
- 8. Common Reed Solomon Codes
 - a. RS (255, 223)
 - (1) Sends 255 8 bit symbols, 223 of them are data and 32 are parity
 - (2) Can correct 16 errors or 32 erasures
 - b. RS(204, 188) - Used in Digital Video
 - (1) Sends 204 8 bit characters, 188 of them are data and 16 are parity
 - (2) Can correct 8 symbol errors
- 9. Can be done on other than 8-bit characters

E. More on FEC

- 1. FEC is often used to improve the BER of lines.
 - a. Can be more cost effective than boosting the power sent over the line
 - b. Retransmission at higher levels may not be practical
- 2. Error Correcting web pages
 - a. <http://www.piclist.com/techref/method/errors.htm>
 - b. <http://people.qualcomm.com/karn/code/fec/>
 - c. <http://www.engelschall.com/u/sb/hamming/>

IV. Line Discipline

A. Line Discipline defines who can send and when they can send.

B. Simplex system

- 1. Who is always known
- 2. When is usually any time

C. ENQ/ACK system

- 1. Used in point to point systems
- 2. Process
 - a. Sender sends a ENQ (Enquiry) to the intended receiver.
 - b. Receiver sends back an ACK (Acknowledgement) when it is ready to receive
 - c. Sender sends data
 - d. Sender sends EOT (End of Transmission) when done sending.
 - e. Receiver may send a NAK (Negative Acknowledgement) if it isn't ready.
- 3. Common to have one node a primary and another a secondary
 - a. Primary is responsible for initiating communication
- 4. Some systems have nodes as peers
 - a. Either can initiate a transfer
- 5. In full-duplex system, data and control messages can be sent simultaneously

D. Poll & Select system

- 1. Always has a primary and secondary nodes
 - a. May be point-to-point or multi-point
- 2. Process - Poll

- a. Primary node requests data from the secondary
 - b. If no data, secondary responds with a NAK
 - c. Otherwise secondary responds with the data, and the primary ACKs this data.
3. Process - Select
 - a. Primary node sends message to secondary
 - b. Secondary responds with ACK if it can accept data
 - c. Primary sends data to secondary
 - d. Secondary responds with ACK after successful transfer of data

E. Network Addresses

1. In a multi-point system, addresses needed so that nodes know who is talking to who
2. Simple in some systems where physical position or dip-switch sets it.
3. Ethernet defines that all manufactures of devices include a unique 48-bit (6-byte) address in every device.
 - a. Polling 2^{48} devices isn't practical
 - b. at 10Mbps, with a 60 byte poll it would take
 - (1) $2^{24} / 10\text{Mbps} * 60 * 8 \text{ bytes} = 1.35\text{E}+10$ seconds
 - (2) or about 428 years

F. Media Access Control

1. MAC - Media access Control
2. A multipoint network with no clear master and slave.
 - a. master == primary
 - b. slave == secondary
3. In this system, there needs to be more complicated protocol than ENQ/ACK or select/poll to figure out who sends when
4. See later on in this lecture....

V. Flow Control

A. Flow control - a protocol to ensure that the sender does not overwhelm a receiving station

B. Incoming data must be checked and processed before it can be used

1. This is often slower than the transmission rate
2. Receivers need to buffer received data until it is processed
3. Buffers are always limited

C. Stop-n-Wait

1. Stop and Wait flow control is simplest form of flow control
2. Process
 - a. Sender sends one frame of data
 - b. Sender waits for ACK before sending more
 - c. Receiver can slow down process by waiting to send ACK
3. May be simple - but it is inefficient
4. Stop-n-Wait efficiency
 - a. Entire set of data is divided into n frames
 - b. t_{prop} = time to go from sender to receiver
 - (1) depends on distance and medium
 - c. t_{frame} = time to send one data frame
 - (1) depends on bit rate and frame size
 - d. $T_D = n(2 t_{\text{prop}} + t_{\text{frame}}) = \text{Time to send data}$

- e. Actual transmission of data is $n * t_{\text{frame}}$
- f. Efficiency is $(n * t_{\text{frame}}) / n(2 t_{\text{prop}} + t_{\text{frame}})$
- g. $= 1 / (1 + 2(t_{\text{prop}} / t_{\text{frame}}))$
- h. If t_{prop} is small and t_{frame} for frame is large, it approaches 100%, but never reaches it.

D. Sliding Window Protocol

1. Sliding Window flow control allows for more than one frame to be sent before an acknowledgement is received
 - a. frames are numbered from 0 to n-1
 - b. Sender sends frames up until it has sent n-1 frames
 - c. Receiver sends back an ACK with the number of the next frame it expects
 - d. Sender can now send up to this new number
 - e. Maximum window size is n-1 frames
2. Sliding Window Efficiency
 - a. If the sliding window is big enough, and the receiver is fast enough, then 100% efficiency can be achieved
 - b. To get 100% -
 - (1) $2 * t_{\text{prop}} \leq ((n - 1) * t_{\text{frame}})$
 - (2) t_{prop} = Time for a frame to propagate from one end to the other of the system
 - (3) t_{frame} = Time to send a frame (based on frame size and bit rate)

VI. Error Control

- A. Error Control is the use of error detection combined with retransmission of data that has had an error detected in it.
- B. Error detection is achieved
 1. using parity, checksums, CRC, etc
 2. Extra data added to each frame
- C. ARQ - Automatic Repeat Request
 1. This abbreviation isn't really used much
- D. Stop-n-Wait Error Control
 1. Extension of Stop-n-Wait flow control
 - a. When a good frame is received, an ACK sent back to originator.
 - b. When a damaged frame is received, a NAK is sent back to originator
 - c. When a NAK is received, the same frame is retransmitted
 2. Any frame may be dropped - data or NAK or ACK.
 - a. If ACK isn't received after some time, a NAK is assumed
 - b. If a NAK is lost - timeout and retransmission
 - c. If a ACK is lost timeout and retransmission
 - (1) Frames must be numbered so receiver can throw away duplicates
 - d. If data is lost - timeout and retransmission
 3. Has same efficiency issues
- E. Go-back-N Error Control
 1. Uses a Sliding window
 2. If a NAK is received or an ACK timeout occurs
 - a. all data since last ACK received is resent
 - b. This may be several frames
 3. ACKs contain next expected frame number

4. NAKs contain errored frame number

F. Selective-Reject Error Control

1. Uses a Sliding window
 - a. If a NAK is received, only the damaged frame is resent
2. Selective Reject is more complicated to implement
 - a. Receiver must contain the sorting logic to enable it to reorder frames
 - b. Sending device must contain searching mechanism to enable it to find and select only the requested frame for retransmission
 - c. Buffer in the receiver must keep all previously received frames on hold until all retransmissions have been sorted, duplicates identified and discarded
 - d. ACK numbers must refer to the frame received instead of next frame expected
3. More efficient as it resends less data
 - a. If errors don't occur very often - there isn't a big difference in efficiency
4. Selective Reject is rarely implemented over Go-back-N method
 - a. Complication makes it slower to get products made
 - b. Increased Efficiency is minimal and not worth the extra effort

VII. Byte Oriented Link Control

A. List here is all from Modem communications world

1. Normally people don't address these as link layer controls

B. Not just data-link as most of these protocols addressed issues of file-transfer

C. XMODEM

1. Stop-n-Wait error control
2. Fixed data field of 128 bytes, CRC-8

D. YMODEM

1. increased data field to 1024 bytes, CRC-16

E. ZMODEM

1. More features
2. Sliding window

F. Kermit

1. sliding window
2. Also a terminal emulation package

G. BSC - Binary synchronous communication

1. Few people care about this one

VIII. Bit Oriented Link Control

A. Many bit Oriented protocols today are oriented around HDLC

B. Ethernet, Token-Ring and other LAN technologies are also bit-oriented

C. SONET, T1, and most telephony systems are bit-oriented.

D. It is common to use HDLC over a SONET or T1 system

IX. HDLC – High-Level Data Link Control

A. HDLC node types

1. Primary station
 - a. Has complete control over the link

- b. Can be used in point-to-point or multi-link
- 2. Secondary station
 - a. Receives commands from the primary and responds accordingly
- 3. Combined station
 - a. Can both command and respond to commands
 - b. Used in peer to peer (or balanced) configurations

B. HDLC Configurations

- 1. Unbalanced
 - a. One primary node and one or more secondary nodes
 - b. May be point-to-point or multi-point
 - c. Full or half-duplex
- 2. Balanced
 - a. point-to-point topology only
 - b. Both nodes are combined stations

C. HDLC Modes of Communication

- 1. NRM - Normal Response mode
 - a. Standard Primary-secondary relationship
 - b. Secondary can transmit only when polled
- 2. ARM - Asynchronous Response Mode
 - a. A secondary may initiate a transmission without permission from the primary
- 3. ABM - Asynchronous Balanced Mode
 - a. Combined stations - either can initiate
 - b. Point-to-point topology

D. HDLC Frame structure

- 1. Frame of data consists of:
 - a. Flag: 8 bits
 - b. Address: 1 or more bytes
 - c. Control: 8 or 16 bits
 - d. Information: many bits
 - e. Frame Check Sequence (FCS): 16 or 32 bits
 - f. Flag: 8 bits
- 2. HDLC Flag byte
 - a. The flag is a special bit pattern that signifies the start/end of a frame.
 - b. A single flag byte may be used to end one frame and start another
 - c. Flag bit pattern is 0111 1110
 - d. Bit stuffing is used to make sure this pattern does not occur anywhere else.
 - e. Bit stuffing occurs on all bytes (control or data) that are not flag bytes
- 3. HDLC Address field
 - a. First seven bits are used for address
 - b. Last bit - if a 1 then there are no more address bytes
 - c. Last bit - if a 0 then there is another address byte
 - d. Last bit of last address byte is always a 1
 - e. Number of address bits is a multiple of 7
- 4. HDLC Control Field
 - a. One or two byte segment for flow management
 - b. Control fields change based on type of frame
 - c. If first bit is 0

- (1) frame is an I-Frame
 - d. If first bit is 1 and second 0
 - (1) frame is an S-Frame
 - e. If first bit is 1 and second 1
 - (1) frame is a U-Frame
 - f. I-Frame - Information Frame
 - (1) 3 bits for send sequence number
 - (2) 3 bits for receive sequence number (ACKs)
 - (3) 1-bit - Poll/Final bit
 - (a) From secondary - set to 1 if last I-frame of response
 - (b) From primary - set to 1 if polling secondary for data
 - g. S-Frame - Supervisory Frame
 - (1) 3 bits for receive sequence number
 - (2) 2 bits for “supervisory” functions
 - (3) 1 bit - Poll/final bit
 - (4) S-Frames used for ACK when no data
 - h. U-Frame - Unnumbered frame
 - (1) 1 bit- Poll/Final bit
 - (2) 5 bits - “Unnumbered function” bits
 - i. S-frames can be used for:
 - (1) reject
 - (2) selective rejects
 - (3) receiver ready
 - (4) receiver not ready
 - j. U-frames can be used for
 - (1) Mode setting (primary/secondary)
 - (2) unnumbered information transfer
 - (3) recovery - bad commands, etc
 - (4) initialization
 - (5) disconnect
 - (6) test command/response
5. HDLC - Information field
- a. The information field is a series of bits until the next flag byte is found
 - b. S-Frame has no information field
 - c. The last 16 or 32 bits of the Information field is actually the FCS
6. FCS - Frame check sequence
- a. a 16 or 32-bit CRC on the data