ECE 428: Tutorial 5

Mohammad Towhidul Islam

June 10, 2009
HDLC DATA LINK CONTROL

• High-level data link control gives the set of standards for operating a data link over bit synchronous physical layers.

• It is derived from SDLC (Synchronous Data Link Control) developed by IBM.

• It supports both Half-duplex and Full-duplex communication.
HDLC Basics

• Stations:
  – Primary: sends data, controls the link with commands
  – Secondary: receives data, responds to control messages
  – Combined: can issue both commands and responses

• Link configuration:
  – Unbalanced: one primary station, one or more secondary stations
  – Balanced: two combined stations
HDLC

- Two common transfer modes: normal response mode (NRM) and asynchronous balanced mode (ABM)
High-level Data Link Control (HDLC)

- **Frame Format**: HDLC frames contain six fields
  - **Flag Field**: 8-bit contains 01111110 to identify the beginning and end of a frame and serves as a synchronization.
  - **Address Field**: One or several byte long field contains address of either the originator or the destination of the frame. If primary creates the frame, it contains a to address. If a secondary creates the frame, it contains a from address.
  - **Control Field**: One or two byte long contains flow/error info.
  - **Information Field**: Variable length field contains user’s information from network layer or network management info.
  - **FCS Field**: Frame-check-sequence is an error detection field contains 2 to 4 byte CRC data.
HDLC: Frames

- **I**(information)-frames, **S**(upervisory)-frames, **U**(unnumbered frame)-frames
- Flag field: 01111110 to identify both the beginning and the end of a frame and serve as synchronization pattern for receiver
- FCS field: 2- or 4-byte ITU-T CRC for error detection
HDLC: Frames

- Control Field: 1- or 2-byte segment of the frame used for flow and error control
- Determine the type of frame and define its functionality
- Control field for I-frame: P/F (poll/final bit for primary/secondary)

![Diagram](image.png)
HDLC: Frames

- Control field for S-frame
- Receive ready (RR), Receive not ready (RNR), Reject (REJ) Selective reject (SREJ)

<table>
<thead>
<tr>
<th>Code</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>RR</td>
</tr>
<tr>
<td>01</td>
<td>REJ</td>
</tr>
<tr>
<td>10</td>
<td>RNR</td>
</tr>
<tr>
<td>11</td>
<td>SREJ</td>
</tr>
</tbody>
</table>
HDLC: Frames

- Control field for U-frame

<table>
<thead>
<tr>
<th>Code</th>
<th>Command</th>
<th>Response</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 001</td>
<td>SNRM</td>
<td></td>
<td>Set normal response mode</td>
</tr>
<tr>
<td>11 011</td>
<td>SNRME</td>
<td></td>
<td>Set normal response mode, extended</td>
</tr>
<tr>
<td>11 100</td>
<td>SABM</td>
<td>DM</td>
<td>Set asynchronous balanced mode or <strong>disconnect mode</strong></td>
</tr>
<tr>
<td>11 110</td>
<td>SABME</td>
<td></td>
<td>Set asynchronous balanced mode, extended</td>
</tr>
<tr>
<td>00 000</td>
<td>UI</td>
<td>UI</td>
<td>Unnumbered information</td>
</tr>
<tr>
<td>00 110</td>
<td></td>
<td>UA</td>
<td><strong>Unnumbered acknowledgment</strong></td>
</tr>
<tr>
<td>00 010</td>
<td>DISC</td>
<td>RD</td>
<td>Disconnect or <strong>request disconnect</strong></td>
</tr>
<tr>
<td>10 000</td>
<td>SIM</td>
<td>RIM</td>
<td>Set initialization mode or <strong>request information mode</strong></td>
</tr>
<tr>
<td>00 100</td>
<td>UP</td>
<td></td>
<td>Unnumbered poll</td>
</tr>
<tr>
<td>11 001</td>
<td>RSET</td>
<td></td>
<td>Reset</td>
</tr>
<tr>
<td>11 101</td>
<td>XID</td>
<td>XID</td>
<td>Exchange ID</td>
</tr>
<tr>
<td>10 001</td>
<td>FRMR</td>
<td>FRMR</td>
<td>Frame reject</td>
</tr>
</tbody>
</table>
HDLC: Example 1

- Connection and disconnection

![Diagram showing HDLC connection and disconnection example](image-url)
HDLC: Example 2

- Piggybacking without error
HDLC: Example 3

- Piggybacking with error
HDLC: Bit Stuffing and Unstuffing

Frame sent

Flag | Address | Control | Data sent
---|---|---|---
0001111111001111101000

Stuffed

Frame received

Flag | Address | Control | Extra 2 bits
---|---|---|---
0001111110110011111001000

Unstuffed

Data received

0001111111001111101000
POINT-TO-POINT PROTOCOL

- PPP can be used as a data link control to connect two routers

- It can be used to connect a personal computer to an internet service provider (ISP)

- It can operate over asynchronous links, but asynchronous links

- The PPP protocol uses HDLC-like frame format to encapsulate data grams over point-to-point links
Point-to-Point Protocol: *PPP*

- **PPP defines/provides**
  - the format of the frame to be exchanged between devices
  - how two devices negotiate the establishment of the link and the exchange of data
  - how network layer data are encapsulated in the data link frame
  - how two devices can authenticate each other
  - multiple network layer services
  - connection over multiple links
  - Network address configuration

- **But, several services are missing for simplicity**
  - no flow control, simple error control (detection and discard), no sophisticated addressing for multipoint configuration
PPP Frame

- Flag: 01111110 the same as HDLC, but it treated as a byte because of PPP is a byte-oriented protocol
- Address: 11111111 (broadcast address)
- Control: No need because PPP has no flow control and limited error control
- PPP is a byte-oriented protocol using byte stuffing with the escape byte 01111101
PPP: Transition States

- **Dead**
  - Carrier detected
  - Carrier dropped

- **Establish**
  - Options agreed by both sides
  - Failed

- **Authenticate**
  - Authentication successful
  - Failed
  - If authentication not needed

- **Terminate**
  - Done

- **Open**
  - Network layer configuration

- **Network**
PPP: Multiplexing

- PPP uses another set of other protocols to establish the link, authenticate the parties, and carry the network layer data.
- Three sets of protocols defined for powerful PPP: LCP, two APs, several NCPs.

LCP: Link Control Protocol
AP: Authentication Protocol
NCP: Network Control Protocol

PPP uses another set of other protocols to establish the link, authenticate the parties, and carry the network layer data.

Three sets of protocols defined for powerful PPP: LCP, two APs, several NCPs.

LCP: 0xC021
AP: 0xC023 and 0xC223
NCP: 0x8021 and ...
Data: 0x0021 and ...
LCP: Encapsulated in a Frame

<table>
<thead>
<tr>
<th>Code</th>
<th>Packet Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>Configure-request</td>
<td>Contains the list of proposed options and their values</td>
</tr>
<tr>
<td>0x02</td>
<td>Configure-ack</td>
<td>Accepts all options proposed</td>
</tr>
<tr>
<td>0x03</td>
<td>Configure-nak</td>
<td>Announces that some options are not acceptable</td>
</tr>
<tr>
<td>0x04</td>
<td>Configure-reject</td>
<td>Announces that some options are not recognized</td>
</tr>
<tr>
<td>0x05</td>
<td>Terminate-request</td>
<td>Request to shut down the line</td>
</tr>
<tr>
<td>0x06</td>
<td>Terminate-ack</td>
<td>Accept the shutdown request</td>
</tr>
<tr>
<td>0x07</td>
<td>Code-reject</td>
<td>Announces an unknown code</td>
</tr>
<tr>
<td>0x08</td>
<td>Protocol-reject</td>
<td>Announces an unknown protocol</td>
</tr>
<tr>
<td>0x09</td>
<td>Echo-request</td>
<td>A type of hello message to check if the other end is alive</td>
</tr>
<tr>
<td>0x0A</td>
<td>Echo-reply</td>
<td>The response to the echo-request message</td>
</tr>
<tr>
<td>0x0B</td>
<td>Discard-request</td>
<td>A request to discard the packet</td>
</tr>
</tbody>
</table>
LCP: Common Options

- Options are inserted in the information field of the configuration packets

<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum receive unit (payload field size)</td>
<td>1500</td>
</tr>
<tr>
<td>Authentication protocol</td>
<td>None</td>
</tr>
<tr>
<td>Protocol field compression</td>
<td>Off</td>
</tr>
<tr>
<td>Address and control field compression</td>
<td>Off</td>
</tr>
</tbody>
</table>
Authentication

• Authentication means *validating the identity of a user* who needs to access

• PPP is designed for use over dial-up links
  ⇒ User authentication is necessary

• PPP has two protocols for authentication
  – Password Authentication Protocol (PAP)
  – Challenge Handshake Authentication Protocol (CHAP)
Password Authentication Protocol (PAP)
Challenge Handshake Authentication Protocol (CHAP)

- Three-way hand-shaking authentication protocol with greater security than PAP
Network Control Protocol: NCP

• PPP is a multiple-network layer protocol.
• It can carry a network data packet from protocols defined by the Internet, OSI, Xerox, DECnet, AppleTalk, Novel
• IPCP (IP Control Protocol)
  – Configures the link used to carry IP packets in the Internet
Example (1)
Example (2)