Data Link Layer
Instructions

- Students are advised to refer online resources and other reference books to get more information related to the subject.
- Figures and formulae used in this PPT are taken as it is from online resources of book of Data communication and Networking by Frouzen.
Introduction to Data Link Layer

- Receives service from physical layer and provides service to the network layer.
- Two models
  - Internet model and IEEE model
- Responsible for carrying data from one hop to the next hop.
- Packet integrity.
- Flow control.
- Access control.
- Examples of LL protocol
  - Ethernet, token ring, FDDI (Fiber distributed Data Interface)
Services Provided by DLL

- Framing and Link access –
  - frame has data filed+header; NL datagram is placed in data field, header includes physical address. Point-to-point, shared
- Reliable delivery
  - Acknowledgement and transmission
- Flow control
  - Prevents from loosing pkts
- Error detection & detection
  - Detection is implemented in HW, ATM provides correction of Header field only.
- Half-duplex & Full-duplex
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.
A frame in a character-oriented protocol
Taxonomy of protocols of data link layer

- Protocols
  - For noiseless channel
    - Simplest
    - Stop-and-Wait
  - For noisy channel
    - Stop-and-Wait ARQ
    - Go-Back-N ARQ
    - Selective Repeat ARQ
Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel.
The design of the simplest protocol with no flow or error control
Flow diagram for simple protocol
Design of Stop-and-Wait Protocol

Sender

Network

Get data

Data link

Receive frame

Send frame

Physical

Receiver

Deliver data

Network

Receive frame

Data link

Send frame

Physical

Data frame

Event: Request from network layer

Repeat forever

Algorithm for sender site

Event: Notification from physical layer

Repeat forever

Algorithm for receiver site

Event: Notification from physical layer

ACK frame
Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We discuss three protocols in this section that use error control.

**Topics discussed in this section:**
- Stop-and-Wait Automatic Repeat Request
- Go-Back-N Automatic Repeat Request
- Selective Repeat Automatic Repeat Request
Error Detection and Correction

- Types of Errors
- Detection
- Correction
Types of Error

- Single bit error: In a single-bit error, only 1 bit in the data unit has changed.

```
0 0 0 0 0 0 0 1 0
```

0 changed to 1

```
0 0 0 0 0 1 0 1 0
```
Types of Error

- Burst (Multiple bit) error: A burst error means that 2 or more bits in the data unit have changed.

```
<table>
<thead>
<tr>
<th>Sent</th>
<th>Corrupted bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1</td>
<td>0 1 0 1 1 1 0 1 0 1 1 0 0 0 1 1</td>
</tr>
</tbody>
</table>
```

Length of burst error (8 bits)
Single-bit error

0 changed to 1

Sent

Received
Multiple-bit error

Two errors

Sent: 0 1 0 0 0 0 1 0

Received: 0 0 0 0 1 0 1 0
Figure 9-4

Burst error

Sent

0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1

Burst error

Received

0 1 0 1 1 1 1 0 1 1 0 1 0 0 0 0 0 1 1
Redundancy

Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination.
Following are the error detection methods widely used

Detection methods

- VRC
- LRC
- CRC
- Checksum
The most common and least expensive mechanism for error detection is VRC, often called a parity check. In this technique, a redundant called a parity check is appended to every data unit so that the total number of 1’s in the unit becomes even. It may be even parity and odd parity.
In longitudinal redundancy check (LRC), a block of bits are organized in a table and a redundant row of bits is added to the whole block.
LRC

Direction of transfer of the whole block

Direction of transfer for each unit

VRCs
Third most powerful of the redundancy checking technique is the cyclic redundancy check (CRC). The redundancy bit used by CRC are divided by dividing the data unit by a predetermined divisor, the remainder is the CRC. Binary division in the CRC generator is shown in the next slide.
Binary Division
Division in CRC encoder

Dataword: 1 0 0 1

Quotient
1 0 1 0

Divisor: 1 0 1 1

Dividend: augmented dataword

Leftmost bit 0: use 0000 divisor

Remainder: 1 1 0

Codeword: 1 0 0 1 1 1 0

Dataword: 1 0 0 1

Remainder: 1 1 0
Division in the CRC decoder for two cases

Codeword \[\text{1001110}\]

Division

Dataword accepted \[\text{1001}\]

Syndrome \[\text{000}\]

Codeword \[\text{1000110}\]

Division

Dataword discarded

Syndrome \[\text{011}\]
The last error detection method we discuss here is called the checksum. The checksum is used in the Internet by several protocols although not at the data link layer. However, we briefly discuss it here to complete our discussion on error checking.
Checksum

Section K  Section 1
\[ n \text{ bits} \quad \ldots \quad n \text{ bits} \quad n \text{ bits} \]

Section 1 \[ n \text{ bits} \]
Section 2 \[ n \text{ bits} \]
\[ \ldots \]
Section K \[ n \text{ bits} \]

Sum \[ n \text{ bits} \]
Complement \[ n \text{ bits} \]
Checksum

Sender

Section 1 \[ n \text{ bits} \]
Section 2 \[ n \text{ bits} \]
\[ \ldots \]
Section k \[ n \text{ bits} \]
\[ \ldots \]
Section 1 \[ n \text{ bits} \]

Checksum \[ n \text{ bits} \]

Receiver

All 1s, accept
Otherwise, reject
Sender site:

1. The message is divided into 16-bit words.
2. The value of the checksum word is set to 0.
3. All words including the checksum are added using one’s complement addition.
4. The sum is complemented and becomes the checksum.
5. The checksum is sent with the data.
Receiver site:
1. The message (including checksum) is divided into 16-bit words.
2. All words are added using one’s complement addition.
3. The sum is complemented and becomes the new checksum.
4. If the value of checksum is 0, the message is accepted; otherwise, it is rejected.
The receiver adds the data unit and the checksum field. If the result is all 1s, the data unit is accepted; otherwise it is discarded.
Hamming Code: As Error correction

Redundancy bits
Hamming Code

r₁ will take care of these bits

1011 1001 0111 0101 0011 0001
11  9  7  5  3  1

d d d d r₈ d d d d r₄ d r₂ r₁

r₂ will take care of these bits

10111010 01110110 0011 0010
11 10 7 6 3 2

d d d d r₈ d d d d r₄ d r₂ r₁
Hamming Code

$r_4$ will take care of these bits

011101100101 0100
7 6 5 4

d d d $r_8$ d d d $r_4$ d $r_2$ $r_1$

$r_8$ will take care of these bits

101110101001 1000
11 10 9 8

d d d $r_8$ d d d $r_4$ d $r_2$ $r_1$
Example of Hamming Code

Data: 1 0 0 1 1 0 1

Adding $r_1$

Adding $r_2$

Adding $r_4$

Adding $r_8$

Code: 1 0 0 1 1 1 0 0 1 0 1
Error Detection

The bit in position 7 is in error.
Data Link Control

- Line Discipline
- Flow Control
- Error Control
Data Link Layer

Application
Presentation
Session
Transport
Network
Data link
Physical

Line discipline
Flow control
Error control
Data link control

- Line discipline: Who should send now?
- Flow control: How much data may be sent?
- Error control: How can errors be corrected?
Flow control

- Stop and wait
  - Send one frame at a time

- Sliding window
  - Send several frames at a time
Stop and Wait

Sender

WT

WT

WT

Time

Receiver

WT = Wait time

Data

ACK

Data

ACK

Data

ACK

EOT

Time
Sliding Window

Window

6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5
This wall moves to the right, frame by frame, when a frame is **sent**.

This wall moves to the right, the size of several frames at a time, when an ACK is **received**.
Receiver Sliding Window

Receiver window

This wall moves to the right, frame by frame, when a frame is received.

This wall moves to the right, the size of several frames at a time, when an ACK is sent.
Sliding Window Example
In data link layer error control refers primarily to method of error detection and retransmission. Error detection in DLL is based on automatic repeat request (ARQ) which means retransmission of data in three cases: Damaged frame, Lost frame and lost acknowledgement. Following are the techniques of error control:

- Stop-and-wait ARQ
- Sliding window ARQ
  - Go-back-n
  - Selective-reject
Damaged Frame

Sender

Data 1
ACK 1
Data 0
ACK 0
Data 1
NAK
Data 1
ACK 1

Receiver

Error in Frame 1

Time

Time
Damaged Frame

Sender

Data 0
Data 1
Data 2
Data 3
Data 4
Data 5
Data 3
Data 4
Data 5

Receiver

Data 0
Data 1
Data 2
ACK 3
NAK 3

Error in frame 3
Discarded
Discarded

Data 3
Data 4
Data 5
Lost Frame

Sender

Data 0
Data 1
Data 2
Data 3
Data 4
Data 2
Data 3
Data 4

Receiver

Data 0
Data 1

Lost

Discarded

NAK 2
Discarded

Discarded

Data 2
Data 3
Data 4
Selective Reject

Sender

Data 0
Data 1
Data 2
Data 3
Data 4
Data 5
Data 2

Receiver

Data 0
Data 1
Data 2
NAK 2
Data 3
Data 4
Data 5
Data 2

Resent
Figure 12-1

OSI Model and Project 802

- Other layers
- Network
- LLC
- MAC
- Physical

Project 802

Other layers
- Network
- Data link
- Physical

OSI Model
Project 802

Other layers

- 802.1 Internetworking
- 802.2 Logical link control (LLC)
- 802.3 CSMA/CD
- 802.4 Token bus
- 802.5 Token ring

Network
Data link
Physical

Project 802
OSI Model
IEEE model takes the structure of HDLC and divided it into two parts:

- **LLC** – One set contain end user portion of the frame – Logical address, control information and data
- **MAC** – Second part resolve the problem due to shared medium.
- **MAC** protocol are specific to the LAN i.e. CSMA/CD, token bus and token ring etc.
Protocol Data Unit (PDU)

- The data unit in LLC is called PDU
- PDU contains four fields as shown in the diagram
- DSAP and SSAP are addresses used by the LLC to identify the protocol stackes on receiving and sending machines
- control Field of PDU is identical to HDLC
Figure 12-3

PDU Format

**DSAP**: Destination service access point
**SSAP**: Source service access point

Upper-level addressing

**DSAP**
- 0 individual
- 1 group

**SSAP**
- 0 command
- 1 response

Used by IEEE
MAC

- In any broadcast network, the stations must ensure that only one station transmits at a time on the shared communication channel.
- The protocol that determines who can transmit on a broadcast channel are called Medium Access Control (MAC) protocol.
- The MAC protocol are implemented in the MAC sublayer which is the lower sublayer of the data link layer.
MAC Protocol

- Random Access Protocol
  - ALOHA – Designed for wireless Network
  - Types: Pure Aloha
  - Slotted ALOHA
  - CSMA
  - CSMA/CD
  - CSMA/CA

- Channelization Protocol
  - FDMA: Frequency division Multiple Access
  - TDMA: Time Division
  - CDMA: Code Division
Random Access

- When there is a collision a sender waits for random length of time and retransmits the frame
  - Aloha: slotted, unslotted (pure)
  - CSMA – ethernet.
ALOHA

- **Basic idea:**
  - let users transmit whenever they have data to be sent.
  - When collision occurs, wait a *random time* (*why?*) and retransmit again.

- **Differences between regular errors & collision**
  - Regular errors only affect a single station
  - Collision affects more than one
  - The retransmission may collide again
  - Even the first bit of a frame overlaps with the last bit of a frame almost finished, then two frames are totally destroyed.
CSMA (Carrier sensing multiple access)

- Problem with ALOHAs: low throughput because the collision wastes transmission bandwidth.

- Solution: avoid transmission that are certain to cause collision, that is CSMA. Any station listens to the medium, if there is some transmission going on the medium, it will postpone its transmission.
Suppose $t_{\text{prop}}$ is propagation delay from one extreme end to the other extreme end of the medium. When transmission is going on, a station can listen to the medium and detect it.

After $t_{\text{prop}}$, A’s transmission will arrive the other end; every station will hear it and refrain from the transmission, so A captures the medium and can finish its transmission.

Vulnerable period = $t_{\text{prop}}$  

But in ALOHAs, it is $X$ or $2X$. In LAN, generally, $t_{\text{prop}} < X$

CSMA random access scheme

Figure 6.19
Figure 12-7

MAC Frame

<table>
<thead>
<tr>
<th>Preamble</th>
<th>SFD</th>
<th>Destination address</th>
<th>Source address</th>
<th>Length PDU</th>
<th>Data</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 bytes</td>
<td>1 byte</td>
<td>6 bytes</td>
<td>6 bytes</td>
<td>2 bytes</td>
<td></td>
<td>4 bytes</td>
</tr>
</tbody>
</table>

Preamble
Start field delimiter, flag (10101011)
Token Ring Frame

Data / command

SD  Start delimiter (flag)
AC  Access control (priority)
FC  Frame control (frame type)
ED  End delimiter (flag)
FS  Frame status

Token  SD  AC  ED
Abort  SD  ED

Figure 12-16
Fiber Distributed Data Interface (FDDI)

- FDDI is LAN protocol standardized by ANSI and ITU-T
- It supports data rate of 100Mbps and required fiber optic media
- Access method of FDDI is limited to time or based on token passing
- Each station captured the token may frames as it can within its allotted period giving highest priority to real time data.
**FDDI Frames**

**Token**

<table>
<thead>
<tr>
<th>SD</th>
<th>FC</th>
<th>ED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- **SD**: Start delimiter (flag)
- **FC**: Frame control (frame type)
- **ED**: End delimiter (flag)
- **CRC**: Cyclic redundancy check
- **FS**: Frame status

**LLC Data unit**

- **DSAP**: Destination Service Access Point
- **SSAP**: Source Service Access Point
- **Control**: Control information
- **Inform.**: Information

**Data**

<table>
<thead>
<tr>
<th>SD</th>
<th>FC</th>
<th>Dest. address</th>
<th>Source address</th>
<th>Data</th>
<th>CRC</th>
<th>ED</th>
<th>FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>0-4500</td>
<td>4</td>
<td>0.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>
FDDI Rings

Primary ring

Secondary ring
Figure 12-27

FDDI Ring Failure

Primary ring

Failure

Secondary ring
Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, one of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). PPP is a byte-oriented protocol.
Point –to-Point Protocol

- **SLIP (Serial Line IP)**
  - First protocol for sending IP datagrams over dial-up links (from 1988)
  - Encapsulation, not much else

- **PPP (Point-to-Point Protocol):**
  - Successor to SLIP (1992), with added functionality
  - Used for dial-in and for high-speed routers

- **HDLC (High-Level Data Link)**:
  - Widely used and influential standard (1979)
  - Default protocol for serial links on Cisco routers
  - Actually, PPP is based on a variant of HDLC
Point-to-Point Link
PPP Layers

Data link:
- A variation of HDLC

Physical:
- ANSI standards
Authentication: Authentication play very important role in PPP because PPP is designed for use over dial-up links where verification of user identity is necessary. Authentication means validating the identity of the user who need to access a set of resources. TWO PROTOCOLS USED –PAP AND CHAP

PAP: Is a simple two steps procedure
1. The user who wants to access a system sends an authentication ID and password
2. Systems checks the validity of the ID and Password and either accept or denies connection
Figure 15-7

PAP Packets

<table>
<thead>
<tr>
<th>PAP Packets</th>
<th>1 byte</th>
<th>1 byte</th>
<th>2 bytes</th>
<th>1 byte</th>
<th>Variable</th>
<th>1 byte</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticate-request</td>
<td>Code = 1</td>
<td>ID</td>
<td>Length</td>
<td>User name</td>
<td>User name</td>
<td>Password</td>
<td>Password</td>
</tr>
<tr>
<td>Authenticate-ack</td>
<td>Code = 2</td>
<td>ID</td>
<td>Length</td>
<td>Message length</td>
<td>User name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authenticate-nak</td>
<td>Code = 3</td>
<td>ID</td>
<td>Length</td>
<td>Message length</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flag | Address | Control | Protocol | Payload (and padding) | FCS | Flag

C023_{16}
CHAP: Is a simple three steps procedure
1. The system sends to the user a challenge packet containing a challenge value, usually a few bytes.
2. The user applies a predefined function that takes the challenge value and the user’s own password and creates a result.
3. The system does the same. It applies the same function to the password of the user and the challenge value to create a result.
### CHAP Packets

**Challenge**
- 1 byte: Code = 1
- 1 byte: ID
- 2 bytes: Length
- 1 byte: Challenge length
- Variable: Challenge value
- Variable: Name

**Response**
- 1 byte: Code = 2
- 1 byte: ID
- 2 bytes: Length
- 1 byte: Response length
- Variable: Response value
- Variable: Name

**Success**
- 1 byte: Code = 3
- 1 byte: ID
- 2 bytes: Length
- Variable: Message

**Failure**
- 1 byte: Code = 4
- 1 byte: ID
- 2 bytes: Length
- Variable: Message

---

**Flag** | **Address** | **Control** | **Protocol** | **Payload (and padding)** | **FCS** | **Flag**
---|---|---|---|---|---|---

**C223_{16}**
PPP frame format

Flag | Address | Control | Protocol | Payload | FCS | Flag
---|---|---|---|---|---|---
1 byte | 1 byte | 1 byte | 1 or 2 bytes | Variable | 2 or 4 bytes | 1 byte
PPP is a byte-oriented protocol using byte stuffing with the escape byte 01111101.
PAP packets encapsulated in a PPP frame

Authenticate-request

Authenticate-ack or authenticate-nak

Authenticate-request

Authenticate-ack

Authenticate-nak

Flag | Address | Control | C02316 | Payload (and padding) | FCS | Flag

PAP packets
CHAP packets encapsulated in a PPP frame