Post 2

Thursday, January 24, 2019 10:50 PM

THE DATA LINK LAYER

Purpose of DLL: Data Link Layer: To provides structure to 1s/0s sent over media

TCP/IP network access layer == Data Link (2) & Physical (1)

- DLL: Responsible for frame exchange between nodes over physical media
  - Allows upper layers to access media/controls how data is placed/received

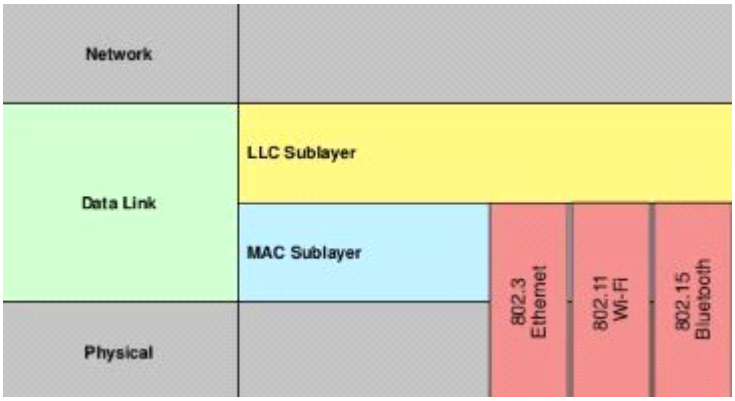
Node: Layer 2 notation for network devices connected to common medium

DLL performs 2 services:

1. **LLC: Logical Link Control:** Accepts Layer 3 packets: Packages them into data units called frames
2. **MAC: Media Access Control:** Error detection
   - **Separates media transitions when packets are forwarded from higher layers**
   - **DLL receives packets from/directs to upper-layer protocols: IPv4/6**
   - **Upper-layer protocol doesn't need to be aware of which media comm will use**

Data Link is divided into 2 sub-layers:

LLC: Logical Link Control	• Upper sub-layer defines software processes that provide services to network layer protocols • Places info in a frame that ID's which protocol is being used • Allows multiple layer 3 protocols (IPv4/6) to utilize the same int/media
MAC: Media Access Control	• Lower sub-layer defines media access processes performed by hardware • DLL addressing of data by signaling requirements of medium/type of protocol in use



Separating Data Link into sub-layers:

- Allows 1 type of frame defined by upper layer to access different types of media defined by lower layer

Examples:

- MAC sub-layer communicates w/Ethernet LAN to send/receive frames over copper/fiber
- MAC sub-layer communicates with Wi-Fi/Bluetooth to send/receive frames wirelessly

Media Access Control Method: Technique used for getting frame on/off media

DLL protocols specify encapsulation of packets into frames:

1. Packets travel from source/dest **->** Pass through diff networks
2. Networks may have diff types of media
3. Packets don't have a way to directly access forms of media
4. DLL prepares network layer packets for transmission & to control access to media
5. Media access control methods define processes which devices can access media
6. Transmits frames

Without DLL: Network protocols (IP) would have to make changes for connecting to media along path

- IP would have to change every time new tech/mediums developed

Serial links: Direct connection bet 2 devices: Data flows sequentially as bits in orderly way

- Routers encapsulate a packet into appropriate frame

- Media access control method used to access each link
- Can be numerous DLL/media transitions

At each hop, a router:

- **Accepts frame from medium**
- **De-encapsulates frame**
- **Re-encapsulates packet into new frame**
- **Fwds new frame to medium of that segment of the network**

Formatting Data for Transmission

- DLL prepares packet for transport by encapsulating it with header/trailer to create frame

DLL frame includes:

Header	**Contains:** Control information/addressing **Location:** Beginning of PDU
Data	**Contains:** IP header/Transport layer header/App data
Trailer	**Contains:** Control information/Error detection **Location:** End of PDU

Framing: Breaks stream of bits into decipherable groups w/control info in header/trailer values in diff fields

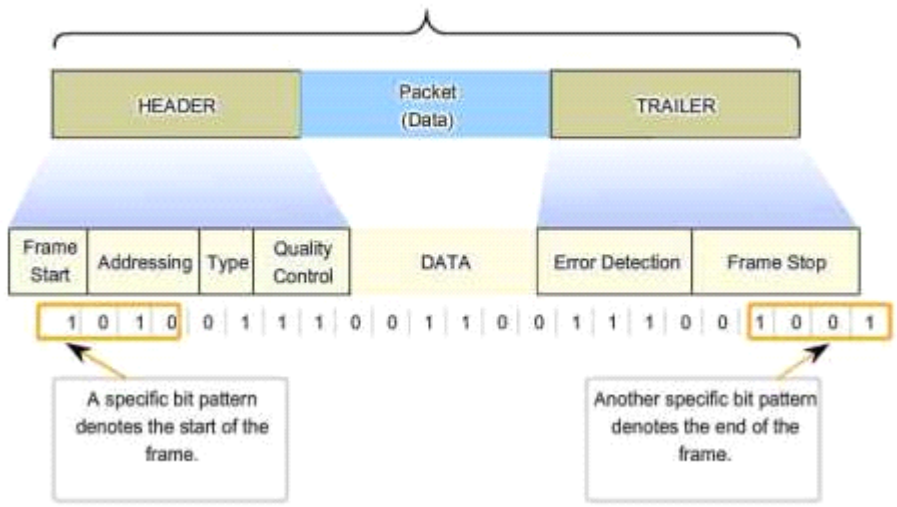
- Format gives physical signals structure that can be received by nodes/decoded into packets

Frame START/STOP flags	**MAC:** ID beginning/end of frame
Addressing	**MAC:** ID source/destination nodes
Type	**LLC:** ID Layer 3 protocol
Control	ID special flow control services
Data	Frame payload (packet/segment header/data)
Error Detection	Included after data to form trailer

Standards for specific DLL protocol define actual frame format

RFC: Requests for Comments: References how tech should work

Formatting Data for Transmission



DLL Standards

IEEE	802.2 Logic Link Control (LLC) 802.3 Ethernet 802.4 Token bus 802.5 Token Ring 802.11 WLAN and Mesh (Wi-Fi Cert) 802.15 Bluetooth 802.16 WiMAX

ITU-UT	G.992: ADSL G.8100-G.8199: MPLS over Transport aspects Q 921: ISDN Q 922: Frame Relay
ISO	HDLC (High Level Data Link Control) ISO 9314: FDDI Media Access Control
ANSI	X3T9.5 and X3T12: Fiber Distributed Data Interface (FDDI)

Topologies

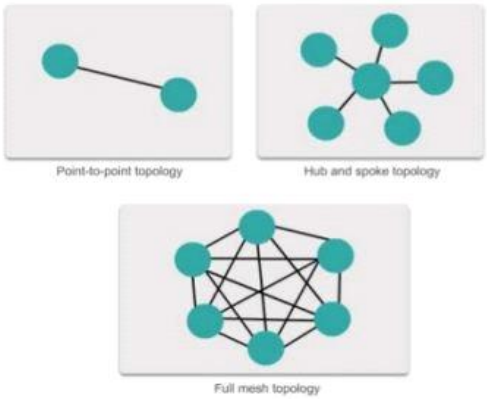
Nodes: Network devices connected on a medium:

- How connected/communicate is described by topology
- Rules specify how/when a node can place data onto media
- Regulating placement of data frames onto media is controlled by MAC sublayer

MAC: Traffic rules on highway (not all roads/entrances/signs the same)

Method used depends on:

Topology	How connection between nodes appears to DLL
Media Sharing	How nodes share media (point-to-point WAN connections) • DLL "sees" logical topology of network when controlling data access to media • Influences type of network framing/MAC used
Physical Topology	Physical connections • ID's how end devices (routers/switches/AP's) interconnect • Usually point-to-point/star
Logical Topology	The way network transfers frames from 1 node to the next • Arrangement consists of virtual connections bet nodes • Logical signal paths defined by DLL protocols



Common Physical WAN Topologies

Point-to-point	Consists of permanent link bet 2 endpoints
Hub-and-Spoke	WAN version of star topology: Central site interconnects branch sites using point-to-point links
Mesh	High availability: Requires every end system to be connected to every other system • Each link point-to-point link to other node • Variations: Partial mesh (some but not all end devices connected)

Physical Point-to-Point: Directly connect 2 nodes: They don't have to share media with other hosts

- Nodes don't have to determine incoming frame destination for other nodes
- Frames placed on media by node at 1 end/taken off at other of point-to-point circuit

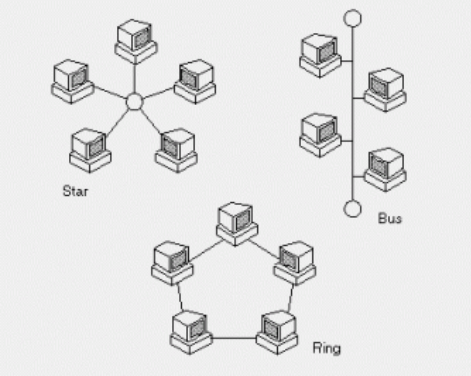
Logical Point-to-Point: Physical devices don't affect logical topology

- Source/destination node can be indirectly connected to each other over geo distance
- **Virtual Circuit:** Logical connection w/in network bet 2 devices (nodes exchanges frames)

- Connection bet 2 nodes might not be bet 2 physical nodes at each end of physical link

Point-to-point networks, data flows 1 of 2 ways:

Half-Duplex	• Both devices transmit/receive on media: Can't simultaneously • Ethernet rules for resolving issues when more than 1 station attempts simultaneous transmission
Full-Duplex	• Both devices transmit/receive on media simultaneously • DLL assumes media avail for transmission for both nodes at any time • No media arbitration in DLL



Physical LAN Topologies:

Star	End devices connected to central intermediate one • Easy to install/scalable/troubleshoot
Extended Star/Hybrid	Combo of other topologies (like star using bus)
Bus	All end systems chained to each other/terminated in some form at each end • Switches aren't required: Legacy Ethernet
Ring	End systems connected to neighbor, forming ring • Doesn't need to be terminated • Legacy FDDI: Fiber Distributed Data Int networks • FDDI networks: Employ 2nd ring for fault tolerance/performance

Logical Topology for Shared Media: 2 MAC methods for:

Contention-based access	All nodes compete for use of medium but have plan if collisions • Doesn't scale well under heavy media use/No overhead of controlled access
Controlled access	Each node has its own time to use medium

Contention-Based Access:

CSMA (Carrier sense multiple access): Prevents chaos on media by detecting whether media is carrying signal

- If signal from another node detected: Another device transmitting
- When device attempts to transmit: Sees media busy: Tries again shortly after
- If no signal detected: Transmits data
- Ethernet/Wireless use contention-based MAC
- If process fails/collision occurs: Data by both devices drops/needs to be resent

CSMA has a method for resolving media contention:

CSMA/CD	**Carrier Sense Multiple Access/Collision Detection** • End device monitors media for signal • If signal absent: Device transmits • If signal detected/shows device transmitting at same time ○ All devices stop sending/try again later Traditional forms of Ethernet use this
CSMA/CA	**Carrier Sense Multiple Access/Collision Avoidance** • End device examines media for signal • If media free: Device sends notification of intent to use

- After receiving clearance: Sends data

Used by 802.11 wireless

Characteristics of contention based access:

1. Stations can transmit onto media any time
2. Collisions exist
3. Mechanisms resolve media contention

Multi-access Topology (Enables number of nodes to communicate by using same shared media)

- Data from 1 node can be placed on medium at any time
- Every node sees all frames on medium

Controlled Access: Devices take turns, in sequence, to access medium

- If end device doesn't need access: Passes to next device
- Process facilitated by use of token
- End device acquires token/Places frame on media
- No other device can do so until frame processed/releases token
- **Examples:** Token Ring (IEE 802.5) & FDDI (based on the IEE 802.4 token bus protocol)

Characteristics of controlled access:

- Only 1 station can transmit at time
- Devices wanting to transmit must wait their turn
- No collisions on media
- Can use token passing to avoid contention

Ring Topology:

Token passing: MAC technique: Allows each node to receive frame: If frame isn't addressed to node: Passes to next node

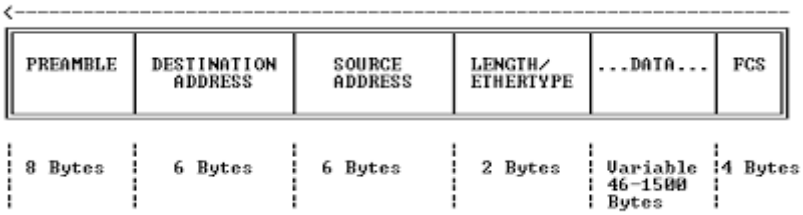
DLL Frame

3 basic parts:

1. Header
2. Data
3. Trailer
    - Frame control unique to each type of protocol

Ethernet frame header fields:

Start Frame	Beginning of frame: Tells other devices frame is starting to transmit on medium
Source/Dest Address	Source/destination of nodes on media
Type	Upper-layer service contained in frame/length of frame



Other Layer 2 protocol header frame fields could include:

Priority/QoS	Particular type of communication service for processing
Logical Connection Control	Establishes logical connection bet nodes
Physical Link Control	Establishes media link
Congestion Control	Indicates congestion on media

Trailer: Error detection used because of interference

- Transmitting node creates logical summary of frame contents
- Summary calculated based on frame data: CRC

CRC: Cyclic Redundancy Check is the value

- Placed in **FCS: Frame Check Sequence** field of frame to represent contents
- Receiving node compares CRC values (if same than intact)

Common DLL protocols include:

Ethernet PPP: Point-to-Point 802.11 Frame Relay HDLC: High-Level Data Link Control

Ethernet: Traditionally unacknowledged connectionless service over shared media using CSMA/CD as access method

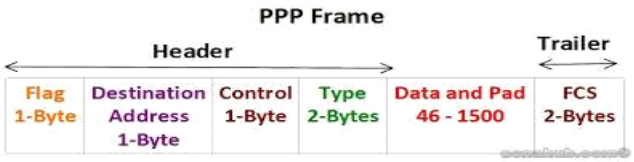
- Shared media required frame header use DLL address to ID source/destination nodes
- MAC address of the node
- Ethernet MAC address 48 bits // Hex

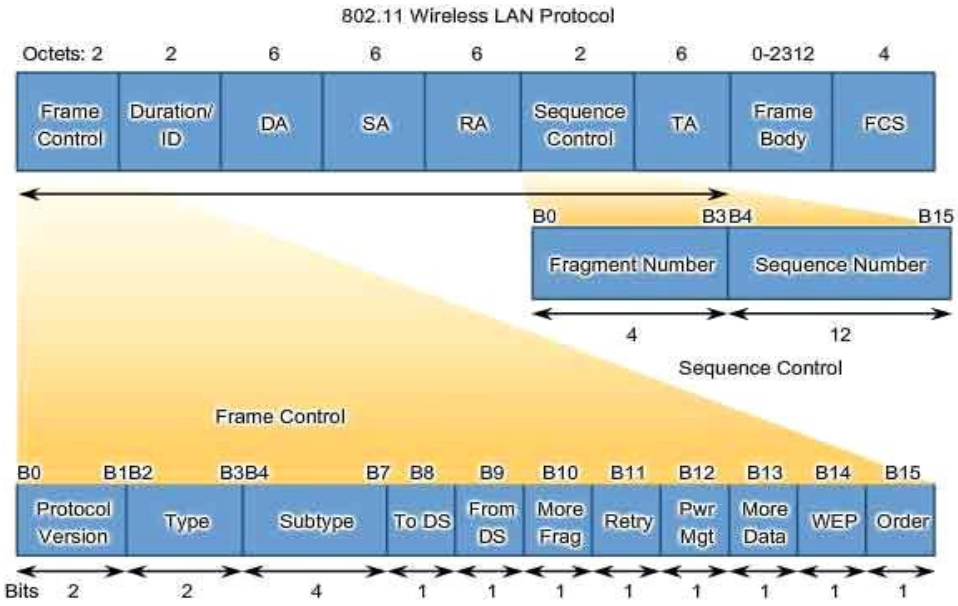
Preamble	Time sync: Contains delimiter to mark end of timing info
Destination Address	48 bit MAC address: Destination node
Source Address	48 bit MAC address: Source node
Type	Indicates which upper-layer protocol will receive data after process complete
Data/Payload	PDU (typically IPv4 packet) to be transmitted over media
FCS: Frame Check Sequence	CRC value used to check for damaged frames

PPP (Point-to-Point Protocol) Frame: Used to deliver frames bet 2 nodes: Dev as WAN protocol

- Can be used on various physical media including twisted-pair/fiber/satellite/virtual connections etc…
- Uses layered architecture
- Establishes logical connections bet 2 nodes
- PPP session hides underlying physical media from upper layers of PPP
- Sessions also provide method for encapsulating multiple protocols over point-to-point link
- Each protocol encapsulated over link establishes its own PPP session
- Also allows 2 nodes to negotiate options (auth/compression/multilink)

Flag	Single byte: Indicates beginning/end of frame: Binary sequence 01111110
Address	Single byte: Contains standard PPP broadcast address: Doesn't assign individual station addresses
Control	Single bye: Contains binary sequence 00000011: Calls for transmission of user data in unsequenced frame
Protocol	2 bytes: ID protocol encapsulated in data field of frame • Most up2date values of protocol field specified in most recent Assigned Numbers RFC
Data	Zero/more bytes: Contain datagram for protocol specified in protocol field
FCS	Normally 16bits (2bytes): Can use 32bit (4bytes) FCS for improved error detection




802.11 Wireless LAN Protocol
Octets: 2
2
6
6
6
2
6
0-2312
4
Frame Control
Duration/ ID
DA
SA
RA
Sequence Control
TA
Frame Body
FCS
B0
B3
B4
B15
Fragment Number
Sequence Number
4
12
Sequence Control
Frame Control
B0
B1
B2
B3
B4
B7
B8
B9
B10
B11
B12
B13
B14
B15
Protocol Version
Type
Subtype
To DS
From DS
More Frag
Retry
Pwr Mgt
More Data
WEP
Order
Bits
2
2
4
1
1
1
1
1
1
1
1