06. The Link Layer and LANs

6.1 Introduction to the Link Layer

The link layer is responsible for transporting frames from one node to another. This can happen over multiple communication channels that can be wired or wireless.

The link-layer may provide several services such as:

- Framing: encapsulate datagram into the frame, adding header + trailer.
- Reliable delivery between adjacent nodes: rarely ever used on wired links because these are pretty reliable.
- Flow control: pacing between adjacent sending and receiving nodes
- Error detection: detect errors caused by signal attenuation, noise, etc. The receiver detects an error, signals retransmission, or drops the frame.
- Error correction: receiver identifies and corrects bit errors without retransmission.
- Half-duplex and full-duplex: nodes at both ends of the link can transmit, but not at the same time

The link layer is typically implemented in hardware or a Network Interface Card on the node's physical chip.

The sending side on the link-layer encapsulates the datagram into a frame, adding error-checking bits and other services the node provides. The receiver can use these services to look for errors or request retransmission before extracting the datagram and passing it to the next layer.

6.2 Error-Detection and -Correction Techniques

Error detection at the link layer is not 100% reliable meaning that some protocols may miss errors under extreme conditions. There are typically three different techniques in use:

1. Parity checking: set a parity bit such that there is an even amount of 1's in the datagram + parity bit. This can be done two-dimensionally, checking both column and row. Two-dimensional parity checking may also be able to do some corrections.

2. Checksum: uses the same checksum method as the Internet Checksum, treating datagram contents as a sequence of 16-bit integers and performing a 1-s complement on them.
3. Cyclic Redundancy Check (CRC): this method is widely used in practice. Sender and receiver determine a number G, and for each D bit of data, we add r bits R such that $\frac{d*2^r}{G} = R$.

6.3 Multiple Access Links and Protocols

Multiple access protocols are needed where multiple nodes broadcast to multiple nodes at the same time. You cannot broadcast two messages at once, because you wouldn't be able to piece them together later. There are three types of Media Access Control (MAC) protocols.

1. Channel partitioning: each node receives time where they can send packets. TDM divides time into time frames, where each node gets a slot. Maximum speed is thus $\frac{R}{n}$ for R speed over n nodes. FDM also gives each node a frequency. This means FDM only works with analog signals.
2. Random Access Protocols:
    a. Slotted ALOHA: Splits time into slots large enough to transfer a frame. Nodes transmit when needed, and if a collision occurs, each node has a p probability of retransmitting the next slot, otherwise, wait until the next.
    b. ALOHA: Same as slotted ALOHA, except there is no synchronizing of slots. This means frames may overlap, and the probability of collision is higher.
    c. Carrier Sense Multiple Access: Ask other nodes if they are sending something, if nobody else is transmitting, transmit yourself. Collisions may still occur due to propagation delay
    d. Carrier Sense Multiple Access with Collision Detection: Attempt to send. If a collision occurs, use an exponential backoff algorithm to determine how long to wait until re-trying.
3. Taking turn protocols:
    a. Polling protocol: The master node determines when other nodes can send frames

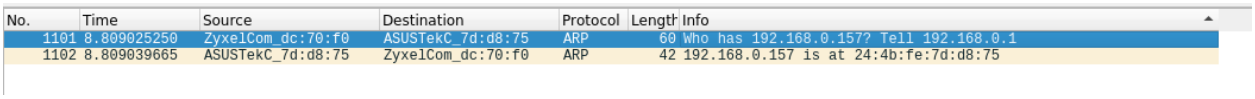
b. Token passing protocol: There is a token passed around, which lets nodes communicate who should transmit. If a node isn't transmitting, it'll just pass the token to the next node.

6.4 Switched Local Area Networks

Each network adapter (one might say router, but it's actually the adapter) has its own MAC address. This is a 48-bit address typically written in hexadecimal notation.

MAC addresses are unique globally, and they're distributed by the IEEE, meaning manufacturers can buy ranges of MAC addresses to assign to their adapters. Each interface on a LAN has its own adapter, and thus its own MAC address.

To determine an interface's MAC address, knowing its IP address is done through the Address Resolution Protocol (ARP). Each IP node on the LAN has an ARP table used to map IP addresses to MAC addresses. This works similarly to DNS, except ARP requests will be broadcast across the entire LAN (as opposed to DNS typically contacting one DNS server)



No.	Time	Source	Destination	Protocol	Length	Info
1101	8.809025250	ZyxelCom_dc:70:f0	ASUSTekC_7d:d8:75	ARP	60	Who has 192.168.0.157? Tell 192.168.0.1
1102	8.809039665	ASUSTekC_7d:d8:75	ZyxelCom_dc:70:f0	ARP	42	192.168.0.157 is at 24:4b:fe:7d:d8:75

When sending a frame to a router, targeted for another host in the network, the sender must set the target MAC to the router's MAC, otherwise, the router's adapter will simply drop the frame.

Ethernet is a protocol for wired LAN technology from the 1980s, becoming the most used and widespread LAN technology in use today. Ethernet is typically placed behind a switch on a network. An ethernet frame has a payload between 46 and 1500 bytes

802.3 Ethernet packet and frame structure

Layer	Preamble	Start frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46-1500 octets	4 octets	12 octets
Layer 2 Ethernet frame			← 64–1522 octets →						
Layer 1 Ethernet packet & IPG	← 72–1530 octets →								← 12 octets →

The preamble consists of a 56-bit pattern, allowing devices on the network to easily synchronize their receiver clocks. (Not sure what the actual purpose of this is). The type indicates the higher layer protocol (typically IP, but might be others. This is used to demultiplex up at receiver).

Ethernet is connectionless and unreliable, meaning if the higher layer doesn't provide reliable data transfer, you might lose frames without knowing. There are also multiple standards for different speeds and physical media, but the frame format is common for all.

A switch is a link-layer device that stores and forwards Ethernet frames among a local set of hosts. It examines incoming frame's MAC addresses, and selectively forwards these to one or multiple links. Uses CSMA/CD to access segment. It's also transparent, meaning hosts are completely unaware that a switch is connected. They do also not need any configuration.

The switch uses the ethernet protocol on each incoming link, such that there are no collisions, it's full-duplex and each link is its own collision domain. This means that transmission to $A \rightarrow A$' and $B \rightarrow B$' may happen simultaneously without collisions (but $A \rightarrow A$' and $B \rightarrow A$' can't).

A switch has a self-learning forwarding table. It keeps the MAC address of each connected host, the physical interface to reach the host, and a time-to-live timestamp. It's similar to a routing table. When frames are received, the switch "learns" the location

of the sender by inspecting the MAC address and associating it with the link it came from.

If an incoming frame doesn't have an entry found, it’s broadcast to the entire network (apart from the arriving link). If the destination is the same as the sender the frame gets dropped. Otherwise, it’s forwarded to the frame as indicated in the switch table.

Switches are similar to routers but serve a different purpose. Routers operate on IP addresses using routing algorithms, whereas switches operate on MAC addresses by learning about their neighbors themselves.

Virtual local networks are networks made for scale. With a single broadcast domain you get a scaling problem when ARP, DHCP, unknown MACs, etc. must all traverse across the entire network. It’s also a security concern because all of those messages are broadcast to the entire network.

There are two primary types of VLANs:

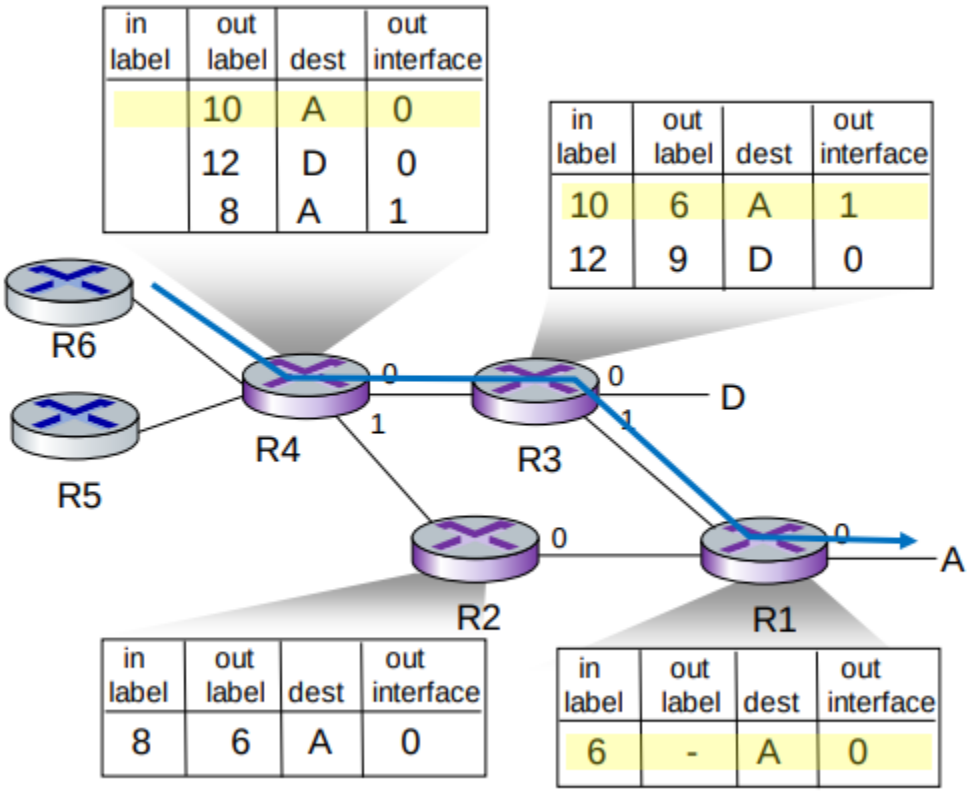
1. Port-based: ports on the physical switch are grouped together by switch management software such that they act as a single switch. It allows dynamic membership (as ports are dynamically assigned through software) and forwarding between VLANs is done via routing. In practice, this is done by selling switches that have routing functionality in them.
2. MAC-address based: groups nodes on the network by their MAC addresses. Also configurable through software.

VLAN frames differ slightly from the regular Ethernet format in the way you need a Tag Protocol Identifier (TPID) of two bytes and a Tag Control Information attached (which results in the Ethernet CRC being recomputed). Any traffic leading out of the VLAN discards this info as it’s only used for internal routing.

6.5 Link Virtualization: A Network as a Link Layer

Multiprotocol label switching (MPLS) is a “protocol” sitting between the Ethernet (or other) and IP (or other) layer with the goal of making high-speed IP forwarding possible by doing lookup using a fixed-length identifier.

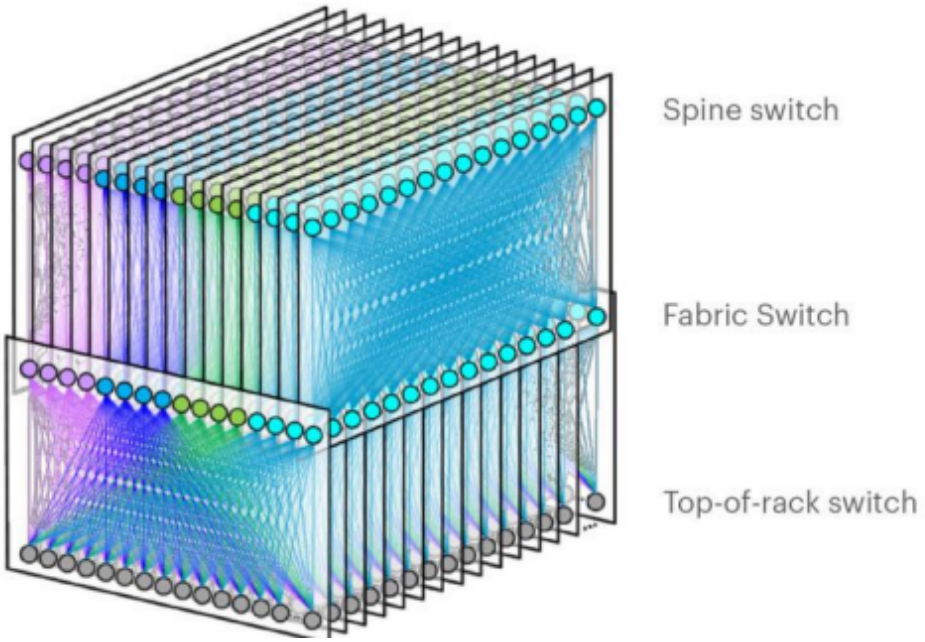
It forwards packets to the outgoing interface based only on the label value, meaning it doesn't have to look into the IP address at all. MPLS routers have their own forwarding tables which are used when frames arrive.



6.6 Data Center Networking

Data centers are networks with hundreds of thousands of hosts, closely coupled in close physical proximity. This is used for large online services like streaming, search engine, expensive computation, and so on. The biggest challenges faced are reliability and how to optimize the networking across these machines.

In a data center, machines are stack into racks, which each have their Top of Rack (TOR) switch. These switches are typically interconnected at another layer. This is what Facebook's F16 data center topology looks like.



There have been plenty of neat innovations with the development of data center networks and networks on a large scale. SDN is widely used within data centers, and features like RoCE at the link layer are also commonly used.

6.7 Retrospective: A Day in the Life of a Web Page Request

In this scenario, Bob connects to a webpage using his school's ethernet. This is the sequence of events that unfold:

1. Bob's operating system produces a DHCP request message and puts this into a UDP segment with port 67 (DHCP server). This is put into an IP datagram with the broadcast destination address 255.255.255.255 and a source of 0.0.0.0 since Bob has no IP as of now.
2. The IP datagram is put inside an Ethernet frame with destination MAC of ff:ff:ff:ff:ff:ff to be broadcast over all devices on the switch. The frame has a source MAC of Bob's network adapter.
3. The router receives the Ethernet frame, and the MAC and IP are extracted from the frame. The broadcast IP indicates that it should be demultiplexed at this stage, and the UDP segment is extracted. The UDP segment is demultiplexed such that the DHCP message is revelaed. The DHCP sever now has the DHCP message.

4. The DHCP server allocates a new address, and forms a DHCP ACK message containing the IP, as well as the IP of the default DNS server. This is put back into a UDP segment, wrapped in an IP datagram, wrapped in an Ethernet frame headed for Bob's computer.
5. The frame with the DHCP ACK is cast out to the switch again, which knows where to direct the MAC address, because it learnt Bob's MAC address and physical interface when Bob sent the DHCP request.
6. Bob receives the DHCP ACK message, and his DHCP client records the assigned IP address as well as the DNS server address.
7. Bob's web browser creates a TCP connection which will be used to handle transport the HTTP request. Before it can do so, it needs the IP of the target webpage. This is done with DNS.
8. Bob's operating system creates a DNS query, wrapped in an UDP segment, wrapped in an IP datagram with a destination address of the default DNS server returned in the DHCP ACK message.
9. Because Bob's computer only knows the IP address of the gateway router it needs to acquire its MAC address. It creates an ARP query placed in an Ethernet frame which is broadcast with a destination address of ff:ff:ff:ff:ff:ff.
10. The gateway router notices that it's the requested node, and prepares an ARP reply with its own MAC address in an Ethernet frame which is to be sent back to Bob. Bob can finally send his DNS query because he has the details needed to send frames to the gateway router.
11. Bob's gateway router receives the frame and unwraps it to find the destination IP address contained in the IP segment and determines the outgoing interface from its forwarding table.
12. The frame finally reaches the DNS server where a DNS reply is formed and is sent back to Bob's laptop which extracts the IP address from the reply.
13. The TCP socket connection is established through the three-way-handshake (SYN, SYN ACK, ACK), and a HTTP GET request is formed.
14. The receiving server handles the HTTP GET request and forms a HTTP response which is sent back to Bob's browser which renders the web page.