The Link Layer

Introduction to the Link Layer

- The **link layer** is responsible for **transferring datagrams** from one **node** to another **physically adjacent** node over a **link**.
- When working at the **link layer**, we refer to **hosts and routers** as **nodes**.
- When working at the **link layer**, we refer to **communication channels** that **connect adjacent nodes** as **links**.
  - These links can be wired and wireless.
- When working at the **link layer**, **frames** are the **unit of data** we are interested in **transferring**; they **encapsulate packets**.
- Different **types of links** have **different transfer protocols**. Different protocols provice different services (eg one may be reliable, the other may not be reliable).

Link Layer Services

- The **framing service** is responsible for **encapsulating datagrams** into frames (adding headers, etc). If there is a shared medium, the channel access needs to be specified. MAC addresses are used in frame headers to identify devices.
- The **reliable deliver service** is responsible for ensuring **every frame is correctly delivered**.
- The **flow control service** is responsible for **pacing sending and receiving rates**.
- The **error detection service** is responsible for **detecting errors** caused by frame drops, noise, signal attenuation, etc.
- The **error correction service** is responsible for **identifying incorrect bits, and correcting them** without the need for retransmission.
- With a **half-duplex service** both nodes can transmit and receive, but not both operations at the same time.
- With a **full-duplex service** both nodes can transmit and receive at the same time.

Link Layer Implementation

- The **link layer** is **implemented** in **each and every node**.
- **Implementations** are located in the node's **Network Interface Card (NIC)**. These cards implement the link and physical layer.
- The **implementations** are **connected** to the **system's bus** which allow for data transfer. And a combination of **hardware, software, and firmware control the NIC**.

Error Detection

- An **Error Correction and Detection (EDC) bit** is used.
- If an **error is detected**, the device will either **correct the error**, or request a **retransmission**.
- **Error detection** is not 100% reliable, but it is still useful. Larger EDC fields result in better detection and correction.
- **Parity checking** is also used.
- **Cyclic Redundancy Check (CRC)** is a good way of **detecting errors**.

Multiple Access Protocols

- There are two types of links:

  1. **Point-To-Point** — Two devices connected directly by a link (for example ethernet between two devices).
  2. **Broadcast** — Several devices connected by a shared medium that can communicate (for example a shared wire, a shared radio, a switch).
  3. When **two or more** nodes **simultaneously transmit**, **interference can occur**.
  4. **Collision occurs** if a **single node** receives **two or more signals at the same time**.
  5. There are **protocols** to **avoid interference and collision**; they involve a **distributed algorithm** that determins **how nodes share a channel**.
  6. **Communication** about **how to use a channel** must be **on the channel**, out-of-band coordination is not allowed in these protocols.

The Media Access Control (MAC) Protocol

- In the **MAC protocol**, there are **three classifications of channel access control**:

  1. **Channel Partitioning** — Channels are divided into smaller pieces (time slots, requency, code, etc), and pieces of the channel are allocated to nodes for exclusive use.
  2. **Random Access** — Channels are not divided, and allow collision, and provides a way to recover from collisions.
  3. **Taking Turns** — Nodes take turns using the channel, but nodes with more to send can take longer turns.

MAC Channel Partitioning Protocols

- **Time Division Multiple Access (TDMA)** gives nodes access to the channel in **rounds**, each node gets a **fixed length slot** (length = packet transmission time) in each round.

  - Unused slots go idle.

- **Frequency Division Multiple Access (FDMA)** divides the channel into **frequency bands**, and each nodes gets a **fixed frequency band**.

  - Unused frequency bands go idle.

MAC Channel Random Access Protocols

- When a **node has a packet to send**, it transmits to the **full channel** at a data rate R **without any prior coordination**.

- If **two nodes transmit** at the **same time**, **collision occurs**.

- The **Random Access protocol** specifies how to **detect and recover** from **collisions**.

- Exampels of **Random Access protocols**:

  1. **ALOHA**:
     - Assumptions:
       (a) All frames are the same size.
       (b) Time is divided into equal size transmission slots.
       (c) Nodes can only start transmitting at a slot beginning.
       (d) The nodes are all time-synchronized.
       (e) If two or more nodes transmit in a slot, all nodes detect the collision.

- When a node obtains a fresh frame, it will transmit it in the next slot, if collision occurs, retransmit, otherwise send the next frame.
- Suppose N nodes with many frames to send, each transmit in a slot with probability P, each time a node attempts to transmit, 37% ($\frac{100}{e}$%) of the time, nodes will be able to transmit without collision.
- Pros:
  (a) Single active node can continuously transmit at full rate.
  (b) Highly decentralized.
  (c) Simple.
- Cons:
  (a) Collision occurs, wasting time slots.
  (b) Some slots are idle.
  (c) Nodes may be able to detect collision in lass than time to transmit a packet.
  (d) Clock synchronization is difficult.

2. **Pure ALOHA**:
   - **Pure ALOHA** is **ALOHA without the timeslots**. When a frame first arrives, nodes can attempt to transmit it immediately.
   - The probability for collision increases with no synchronization.
   - Suppose N nodes with many frames to send, each transmit with probability P, each time a node attempts to transmit, 18% ($\frac{100}{2e}$%) of the time, nodes will be able to transmit without collision.
3. **Simple Carrier Sense Multiple Access (Simple CSMA)**:
   - CSMA requires nodes to listen before they transmit. If the channel is idle, they can transmit the entire frame. If the channel is busy, they can defer the transmission.
4. **CSMA/CD**:
   - **CSMA/CD** is **Simple CSMA** with **Collision Detection**.
   - Collision can be detected within a short period of time.
   - Colliding transmissions are aborted, and rescheduled, reducing channel waste.
   - Collision detection is easy in wired links, but difficult in wireless links.
   - **CSMA/CD** is more efficient than **ALOHA**.

MAC Taking Turns Protocols

- **Taking turns** uses **polling**, a **master** invites **other nodes to transmit in turn**.
- Concerns with this type of protocol are: polling overhead, latency, and a single point of failiure (master).
- Another way to implement taking turns is with **token passing**. A **control token** is **passed sequentially** from **one node to the next**.
- Concerns with token passing are: token overhead, latency, single point of failiure (token).

Local Area Networks (LANs)

MAC Addresses

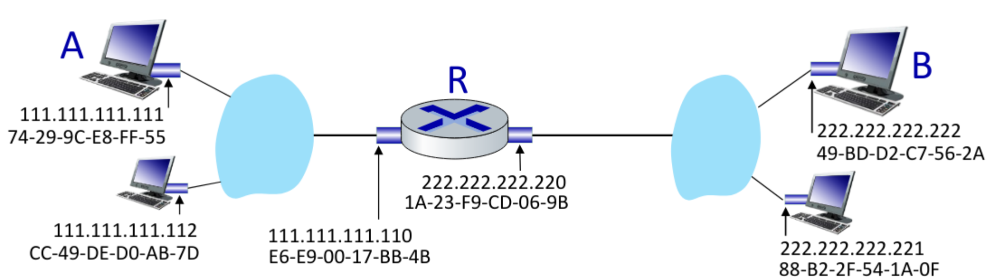
- A **MAC Address** is a **48-bit number** that **uniquely identifies** a **Network Interface Controller (NIC)**.
- **MAC Address allocation** is adiministered by **IEEE**. **Manufacturers** buy **MAC Address ranges** and assign those to devices they make (this is done to ensure uniqueness).
- **MAC Addresses** are used on **layer 2**.

The Address Resolution Protocol (ARP)

- The **Address Resolution Protocol (ARP)** is a **communication protocol** that is used for **discovering** the **link-layer address of devices**.
- **ARP** is a **request-response protocol**, where messages are directly encapsulated by a link layer protocol. Such messages are **only communicated within the boundaries of a single network**, they are never routed across internetworking nodes.
- **Each IP node** on a **local area network** has an **ARP table**. These tables are used to maintain a **mapping** between each **MAC address** and it's corresponding **IP addresses**.
- **ARP table entries** consist of: **a MAC address, IP Addresses, and a TTL**.
- To find the **MAC Address of a node**, the following must be done:
  1. The request node **broadcasts an ARP query** containing the **target node's IP addres**. The **target MAC Address field** in the request is set to **FF:FF:FF:FF:FF:FF** (which broadcasts it to all devices in the local network).
  2. **All nodes** receive the **ARP query**, and check if the target IP matches their IP Address.
  3. The **node** whose **IP Address matches** will send an **ARP response** giving it's **MAC Address**. All other nodes will ignore the **ARP request**.
  4. The **requesting node** will **recieve the ARP response**, and put it into it's **ARP table**.

ARP — Routing to Another Network

- If a **node A** wants to send a **datagram** to another **node B**, through a **router R**, the following must happen:
  1. **Node A** creates an **IP datagram** with **IP source A, and IP destination B**.
  2. **Node A** creates a **link-layer frame** containing an **A-to-B IP datagram**, with **R's MAC Address** in the **frame's destination**.
  3. The **frame** is **sent from A to R**.
  4. The **frame** is **received at R**, and the **datagram is removed**, and passed up to the **IP**.
  5. **R** will change the **source MAC Address** to be **it's own address**, and the **destination MAC Address** to **B's MAC Address**.
  6. **R** creates a **link-layer frame** containing the **datagram**.
  7. **R** will **send** the new **frame**.
  8. **B** will **receive** the **frame**, and extract the **datagram**.
  9. **B** will pass the **datagram** up the protocol stack to the **IP**.



- Every time a packet passes through a router, the source and destination MAC Addresses change.

Ethernet

- **Ethernet** is the first **widely used Local Area Network (LAN) technology**.
- **Ethernet** is simple, cheap, and has a fast data transfer rate.
- Another bennified to **ethernet** is that **a single chip can support multiple transfer speeds**.
- Ways to implement ethernet connections:
  1. **Bus** — A bus allows **several devices** to **communiate** with each other over a **single coaxial cable**. Such a implementation is susceptible to collision.
  2. **Switches** — A switch is a **physical device** that allows **several devices** to **connect to it over a physical link**. It then performs **switching**, which is essentially **forwarding frames** from one **node** to **another**. Such an implementation is not susceptible to collision.

- **Using switches** is a modern way of **connecting devices on a LAN**.
- **Ethernet frames** consist of **six parts**:
  1. **Preamble** — Preamble is used to synchronize sender and receiver clock rates. This consists of **7-bytes** of **10101010**, followed by one byte of **10101011**.
  2. **Destination Address** — The 6-byte destination MAC Address.
  3. **Source Address** — The 6-byte source MAC Address.
  4. **Type** — The type indicates the higher-level protocol (for example IP). This is also used to demultiplex at the receiver.
  5. **Payload** — The datagram.
  6. **CRC** — A cyclic redundancy check at the receiver. If an error is detected, the frame is dropped.
- **Ethernet proerties**:
  1. **Connectionless** — No handshaking is performed between the sending and receiving NICs.
  2. **Unreliable** — The receiving NIC does not send ACKs or NAKs to the sending NIC. Dropped frames are only recovered if the inital sender uses a higher layer RDT (eg TCP).
  3. **MAC Protocol** — Ethernet's MAC protocol is the unslotted CSMA/CD with binary backoff.
- There are **many ethernet standards**, they all have the same **MAC protocol, and frame format**. However, they can **transmit data at different rates**.

Ethernet Switches

- An **ethernet switch** is a **link-layer device** that takes an **active role**.
- **Ethernet switches store and forward ethernet frames**.
- **Ethernet switches** also examing the **MAC addresses** of **incoming frames**, and **selectively forward** the **frame** to **on or more outgoing links** via **CSMA/CD**.
- **Ethernet switches** are **transparent**; **nodes** are **unaware** of the **presence of switches**.
- **Ethernet switches** are **plug-and-play** and **self-learning**. They **do not** need to be **configured**.
- **Nodes** have a **direct, dedicated connection to switches**. This avoids collisions.
- **Ethernet switches** are **full duplex** with **buffering** used on the **incoming data** to **switches**.
- **Each ethernet switch** has a **switch table** that stores the **MAC address, interface number, and timestamp** or **each node** that is **connected to it**.
- As the **ethernet switch** is **self-learning** it can **learn** which **nodes** can be **reached**, and what **interfaces they can be reached through**. As it **learns this information**, it **stores it in the switch table**.
  - There are two ways a switch can learn the MAC address of connected devices:
    1. When a **frame** is sent to the **switch**, it contains a **source MAC address**, the switch "learns" that address, and adds it to the switch table.
    2. When a **frame** is sent to the **switch**, and has a **unmapped destination address**, it will **flood all interfaces** (except the sending interface), with the **frame** until one **responds**. Once a **node responds**, the **switch** will know that the **MAC address blongs to that node**.

Switches vs Routers

- **Routers** are **network-layer devices**.
- **Switches** are **link-layer devices**.
- **Routers** and **switches** both **store-and-forward** units of data.

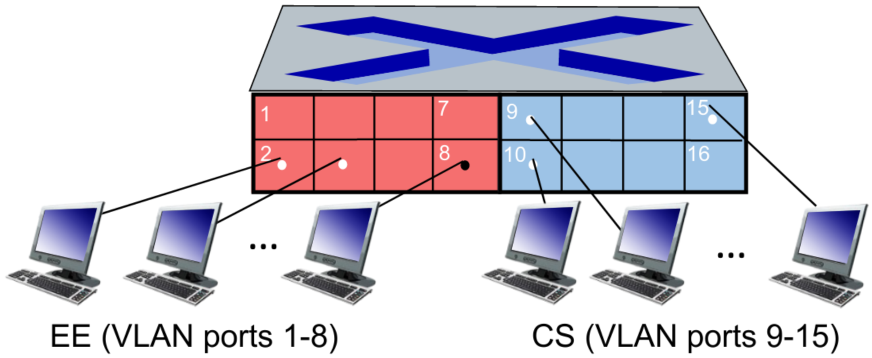
Virtual Local Area Networks (VLANs)

VLAN Motivation

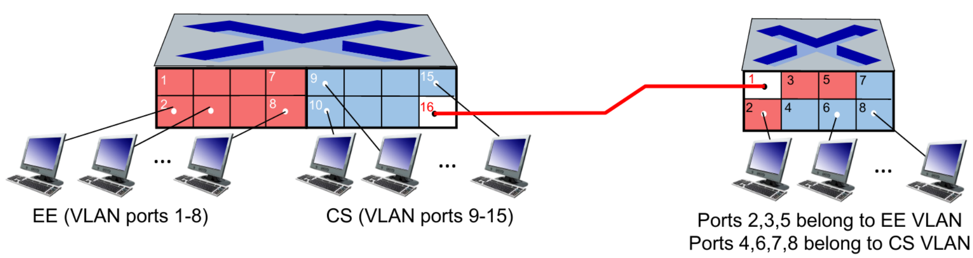
- If a **Local Area Network (LAN) scales** to a **very large size**, then all **layer-2 broadcast traffic** (ARP, DHCP, unknown MAC, etc) must cross the **entire LAN**.
- **Broadcast traffic** on a **large scale** can lead to **low efficiency** and **security / privacy issues**.
- To overcome this issue, we can create **Virtual Local Area Networks (VLANs)**.

Port-Based VLANs

- **Port-based Virtual Local Area Networks (VLANs)** can be configured so specific **port ranges** are part of specific **VLANs**. Effectively allowing a **single physical switch** to operate as **several virtual switches**.



- **Traffic isolation** refers to the **isolation of traffic within virtual networks**. The traffic within a virtual network cannot leave that network on layer 2.
- **Dynamic membership** refers to the **dynamic assigning of ports** among **VLANs**.
- **Forwarding between VLANs** can be done **via routing** through a **router**. In practice, vendors sell switches with build-in routers for this reason.
- **VLANs** that **span multiple switches** use **trunk ports** that **carry frames** between **VLANs** defined over **multiple physical switches**.



- A **problem** with **Port-Based VLANs** is that a **malicious actor** can **obtain access** to a **virtual network** simply by **connecting their device** to **one of the ports on the virtual network**.
  - To get around this issue, **VLANs** can be defined by **device MAC addresses**. You simply **list all of the MAC addresses** that are a part of the **VLAN**.