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

-