Cyclic Redundancy Check (CRC)

- View data bits, \( d_1 d_2 \ldots d_n \), as a polynomial:
  \[
  A(x) = \sum_{i=0}^{n-1} d_i x^i.
  \]

- Choose \( r+1 \) bit pattern (generator), \( G \) (leftmost and rightmost bits are both 1), viewed again as polynomial:
  \[
  G(x) = \sum_{i=0}^{r} g_i x^i.
  \]

- Choose \( r \) CRC bits, \( R \), such that:
  \[
  A(x)x^r + R(x) = G(x)H(x)
  \]
  for some polynomial \( H(x) \). Here, addition of the polynomial coefficients is modulo 2 arithmetic.

- In other words, the polynomial represented by the concatenation of the data bits and the CRC bits is divisible by \( G(x) \).

CRC (continued)

- Note that, since in modulo 2 arithmetic, \( R(x) = -R(x) \), one can also interpret \( R(x) \) as the remainder when \( A(x)x^r \) is divided by \( G(x) \).

- Error detection: divide the received string by \( G(x) \), and if the remainder is non-zero, announce an error.

- Claim: this CRC can detect burst of errors as long as the burst is of length \( r \) or shorter.
**CRC Example**

Addition of 2 polynomials is the same as mod 2 addition of the components of the two vectors of 0,1's (i.e. without carryover)

\[ R = \text{remainder}[\frac{D2^r}{G}] \]

---

**Link Layer**

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-Layer Addressing
- Ethernet
Multiple Access Links and Protocols

Two types of “links”:

- **point-to-point**
  - point-to-point link between Ethernet switch and host

- **shared wire or medium**
  - traditional Ethernet
  - 802.11 wireless LAN

Multiple Access protocols

- **single shared channel**

- **two or more simultaneous transmissions by nodes: interference**
  - collision if node receives two or more signals at the same time

*multiple access protocol*

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
Ideal Multiple Access Protocol

Shared channel of rate $R$ bps
1. When one node wants to transmit, it can send at rate $R$.
2. When $M$ nodes want to transmit, each can send at average rate $R/M$.
3. Fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. Simple

MAC Protocols: a taxonomy

Three broad classes:
- **Channel Partitioning**
  - divide channel into smaller “pieces” (time slots, frequency, code)
  - allocate piece to node for exclusive use
- **Random Access**
  - channel not divided, allow collisions
  - “recover” from collisions
- **"Taking turns" (Centralized polling or token-based)**
  - Nodes take turns, but nodes with more to send can take longer turns
**Channel Partitioning MAC protocols: TDMA**

**TDMA: time division multiple access**
- access to channel in "rounds"
- each node gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-node LAN, 1,3,4 have pkt, slots 2,5,6 idle

![Diagram of TDMA](image)

**Channel Partitioning MAC protocols: FDMA**

**FDMA: frequency division multiple access**
- channel spectrum divided into frequency bands
- each node assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-node LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle

![Diagram of FDMA](image)
Example: GSM

- Global System for Mobile (GSM): digital cellular standard developed in Europe.
- 25MHz band divided in 200 kHz sub-channels, further divided into time-slots.

Channel Partitioning: Pros and Cons

- **Pro**: no conflict between different nodes.
- **Con**: serious waste of resource when a node has nothing to transmit.
- **Good for continuous traffic like voice**
- **Not very efficient for bursty traffic**.
Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate $R$
  - no a priori coordination among nodes
- two or more transmitting nodes $\Rightarrow$ “collision”
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

**Assumptions**
- all frames same size
- time is divided into equal size slots, time to transmit 1 frame
- nodes start to transmit frames only at beginning of slots
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

**Operation**
- when node obtains fresh frame, it transmits in next slot
- no collision, node can send new frame in next slot
- if collision, node retransmits frame in each subsequent slot with prob. $p$ until success
**Slotted ALOHA**

**Pros**
- Single active node can continuously transmit at full rate of channel
- Highly decentralized: only slots in nodes need to be in sync
- Simple

**Cons**
- Collisions, wasting slots
- Idle slots
- Nodes may be able to detect collision in less than time to transmit packet

---

**Slotted Aloha efficiency**

Efficiency is the long-run fraction of successful slots when there are many nodes, each with many frames to send.

- N nodes with many frames to send, each transmits in slot with probability \( p \) (new arrival or re-Tx)
- Prob that node 1 has success in a slot = \( p(1-p)^{N-1} \)
- Prob that any node has a success = \( Np(1-p)^{N-1} \)

- For max efficiency with \( N \) nodes, find \( p^* \) that maximizes \( Np(1-p)^{N-1} \)
- For many nodes, take limit of \( Np^*(1-p^*)^{N-1} \) as \( N \) goes to infinity, gives \( 1/e = .37 \)

At best: channel used for useful transmissions 37% of time!