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 a bit pattern (generator), \(G(x)\) (leftmost and rightmost bits are both 1), viewed 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 two polynomials is the same as mod 2 addition of the components of the two vectors of 0,1's (i.e., without carryover):

\[
\begin{align*}
\text{G} & : 1001 \\
\text{D} & : 1010110000 \\
\end{align*}
\]

No carryover:

\[
\begin{align*}
\text{G} & : 1001 \\
\text{D} & : 10101001000101 \\
\end{align*}
\]

\[ R = \text{remainder} \]

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

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

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

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 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

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

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

- 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 \( 1/e \approx 0.37 \)

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