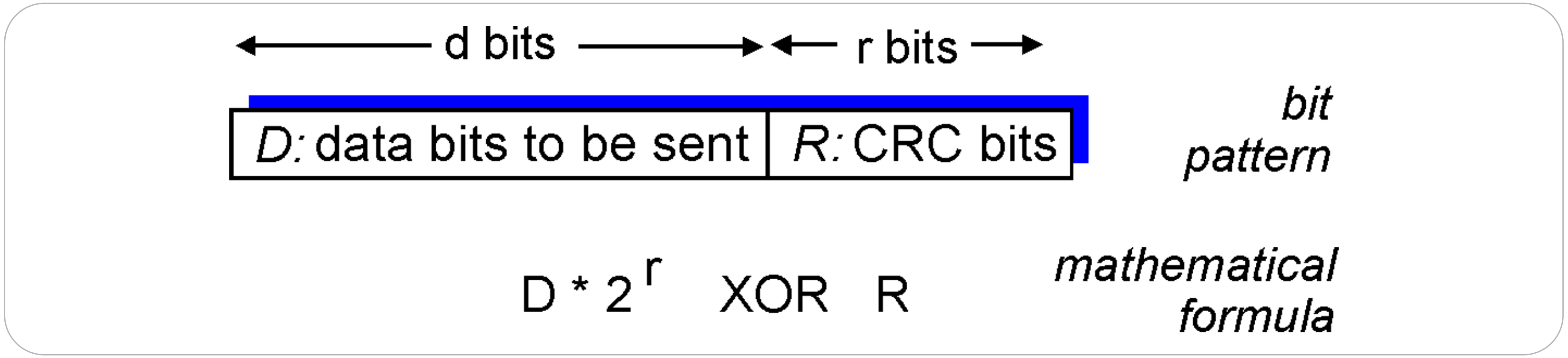
Checksums - CRCs

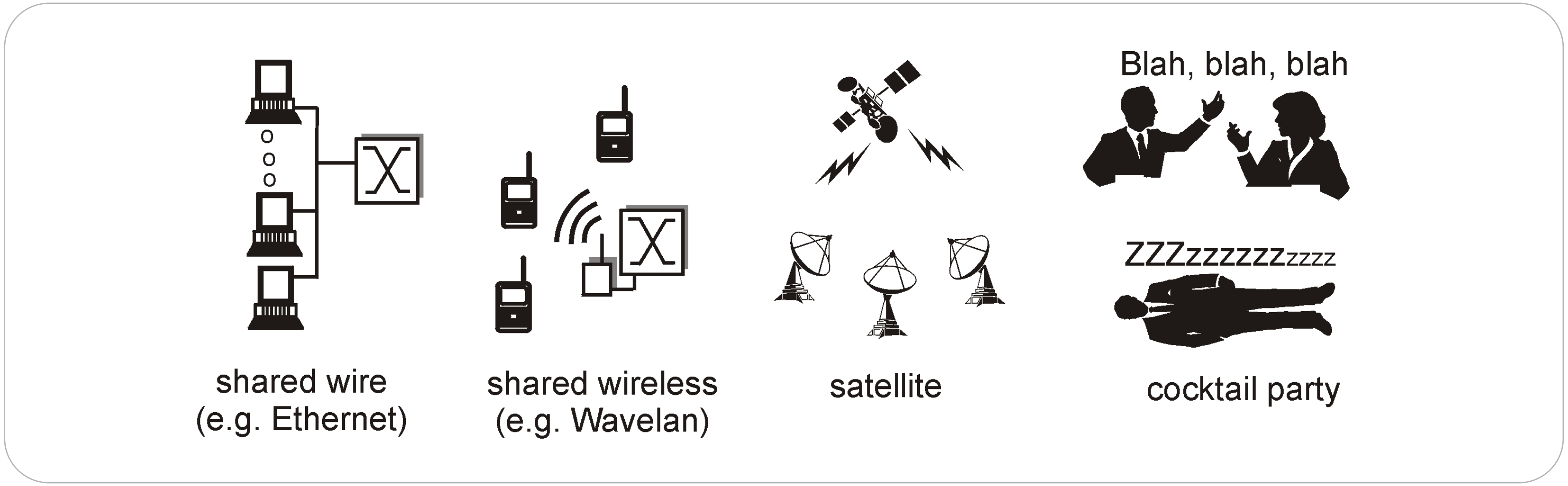
- Cyclic Redundancy Check polynomials
- View data bits, **D** as a binary number
- Choose r+1 bit pattern (generator), **G**
- **Goal:** choose r CRC bits, **R** such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G and divides <D,R> by G
  - if non-zero remainder: error detected!
  - **can detect all burst errors less than r+1 bits**
- Widely used in practice (ATM, HDLC)



Multiple access links

Three types of ‘links’

1. Point-to-point (single wire, eg PPP, SLIP)
2. Broadcast (shared wire or medium, eg Ethernet, Wavelan etc)



3. Switched (eg switched Ethernet, ATM etc)

Multiple access protocols

- Single shared communication channel
- Two or more simultaneous transmissions by nodes: interference
  - only one node can send **successfully** at a time
- Multiple access protocol
  - **distributed** algorithm that determines how stations **share** channel, ie determine when station can transmit
  - communication about channel sharing must use channel itself!
  - what to look for in multiple access protocols
    - synchronous or asynchronous
    - information needed about other stations
    - robustness (eg to channel errors)
    - performance

Multiple access protocols - taxonomy

Three broad classes:

1) Channel partitioning

- Divide channel into smaller ‘pieces’ (time slots, frequency)
- Allocate piece to node for exclusive use

2) Random access

- **Allow** collisions (deal with them)
- ‘Recover’ from collisions

3) ‘Taking turns’

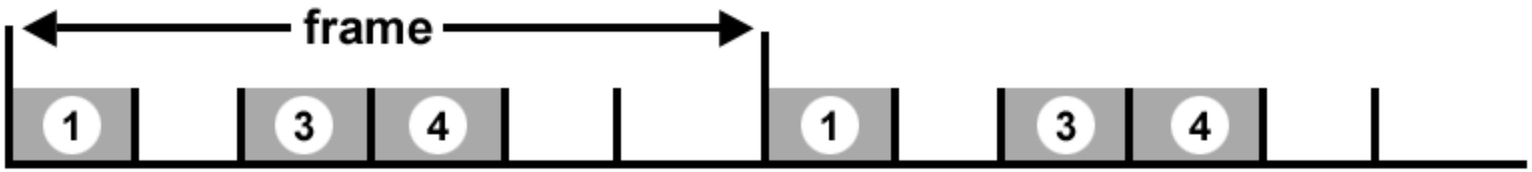
- Tightly coordinate shared access to **avoid** collisions

Goal: efficient, fair, simple, decentralised

Time Division Multiple Access (TDMA)

- Access to channel in ‘rounds’
- Each station gets fixed length slot (length = packet transmission time) in each round
- Unused slots go idle

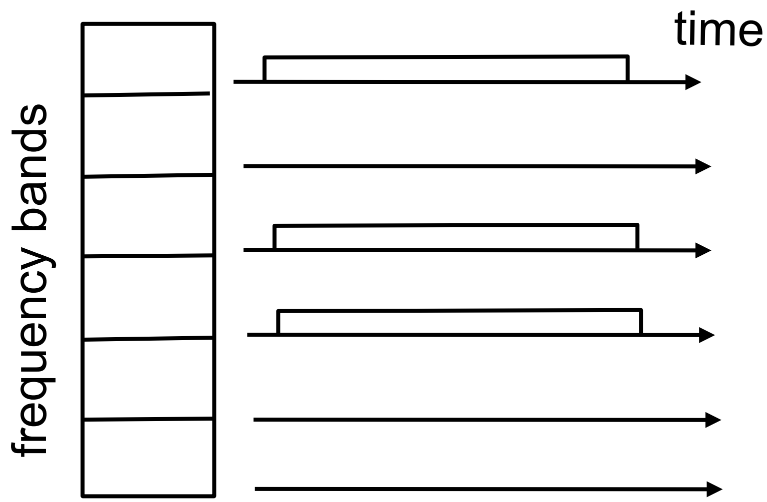
eg 6-station LAN, 1, 3, 4 have packets, slots 2, 5, 6 idle



Frequency Division Multiple Access (FDMA)

- Channel spectrum divided into frequency bands
- Each station assigned fixed frequency band
- Unused transmission time in frequency bands go idle

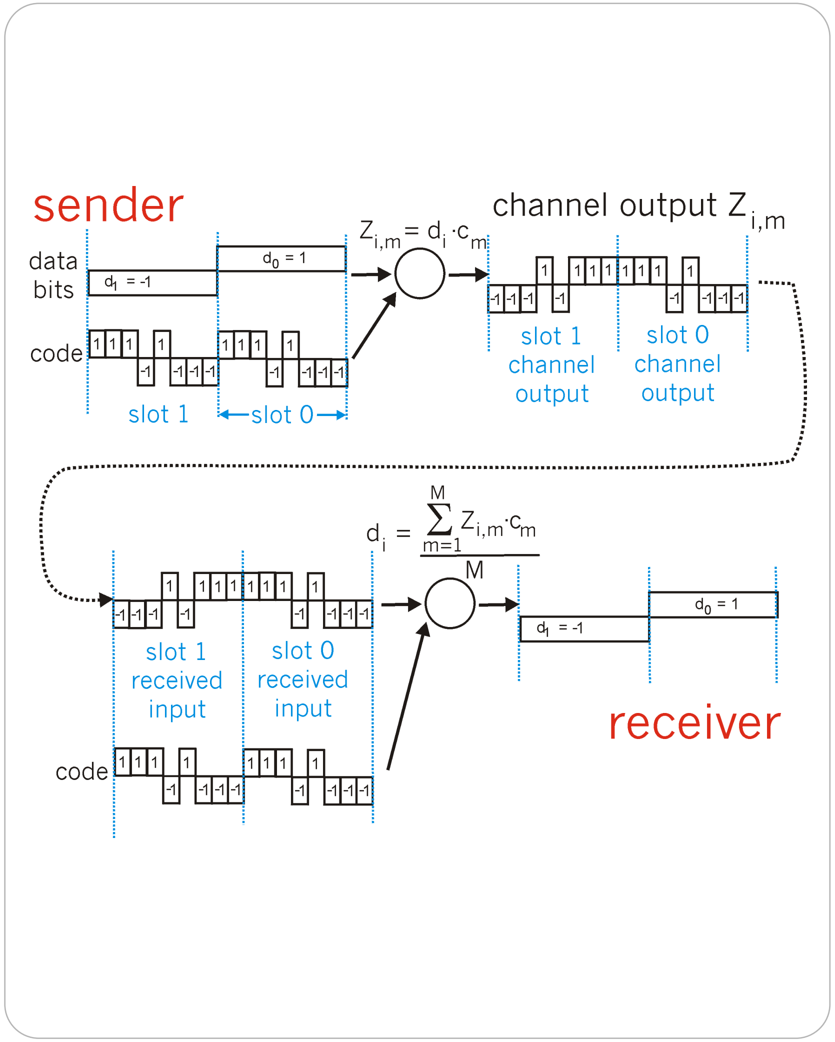
eg 6-station LAN, 1,3,4 have packets, frequency bands 2,5,6 idle



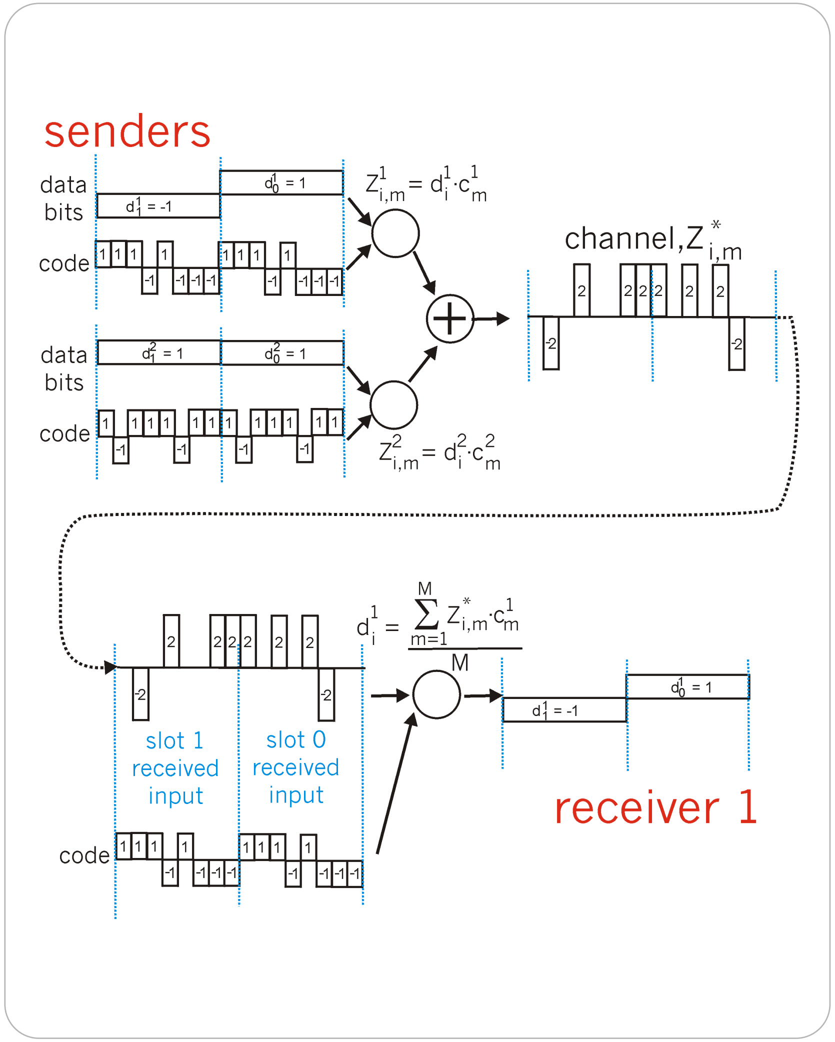
Code Division Multiple Access (CDMA)

- Unique ‘code’ assigned to each user, ie code set partitioning
- Used mostly in wireless broadcast channels (cellular, satellite etc)
- All users share same frequency, but each user has own ‘chipping’ sequence (ie code) to encode data
- **Encoded signal** = (original data) X (chipping sequence)
- **Decoding:** inner product of encoded signal and chipping sequence
- Allows multiple users to ‘coexist’ and transmit simultaneously with minimal interference (if codes are ‘orthogonal’)

CDMA encode/decode sequence



CDMA interference

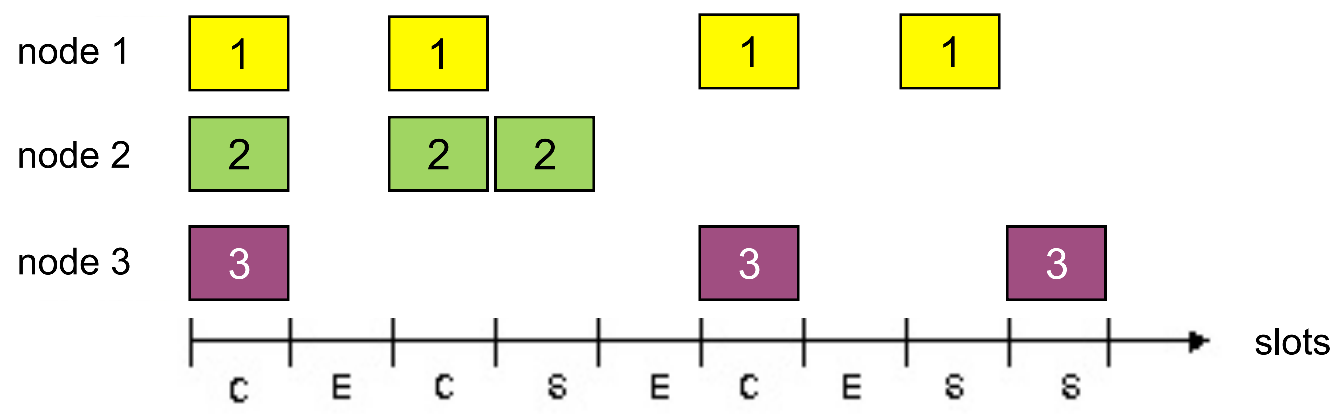


Random access protocols introduction

- When node has packet to send
  - transmit at full channel data rate R
  - no a **priori** coordination among nodes
- Two or more transmitting nodes > collision
- Random access MAC protocol specifies
  - how to detect collisions
  - how to recover from collisions (eg via delayed retransmissions)
- These examples will be covered in the next podcast episode
  - Slotted ALOHA
  - ALOHA
  - CSMA and CSMA/CD

Slotted ALOHA

- Time is divided into equal size slots (= packet transmit time)
- Node with new arriving packet: transmit at beginning of next slot
- If there's a collision: re-transmit the packet in future slots with probability p, until successful



Success (S), Collision (C), Empty (E) slots

Slotted ALOHA - efficiency

Q: What is the maximum fraction of slots that are successful?

A: Suppose N stations **have packets to send**

- each transmits in slot with probability p
- probability of successful transmission, S is:

by single node: S $= p(1-p)^{(N-1)}$

by any of N nodes: S = probability (only one transmits)

$= Np(1-p)^{(N-1)}$

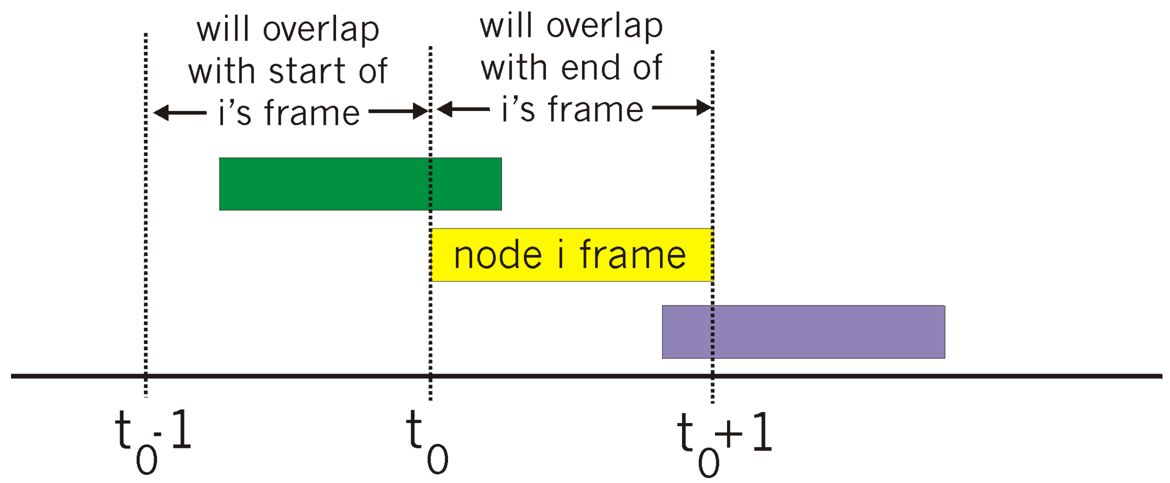
…choosing optimum p as n -> infinity...

= 1/e = 0.37 as N -> infinity

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

Pure ALOHA

- Unslotted ALOHA: simpler, no synchronisation
- Packet needs transmission
  - send without awaiting for beginning of slot
- Collision probability **increases**
  - packet sent at t_0 collides with other packets sent in $[t_0-1, t_0+1]$



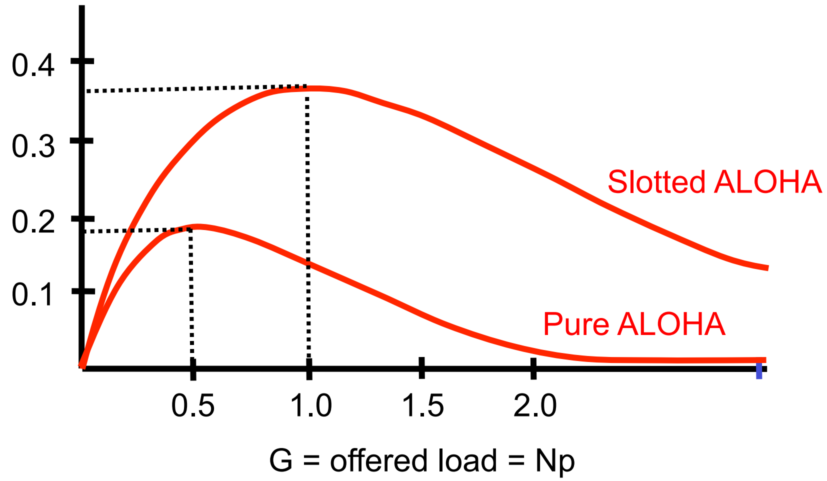
Pure ALOHA - efficiency

P (success by given node) = P(node transmits) .
P(no other node transmits in $[p_0-1,p_0]$.
P(no other node transmits in $[p_0-1,p_0]$
= p . (1-p) . (1-p)

P (success by any of N nodes) = Np . (1-p) . (1-p)

…choosing optimum p as n > infinity...

protocol constrains effective channel throughput!



Carrier Sense Multiple Access (CSMA)

CSMA - listen before transmit

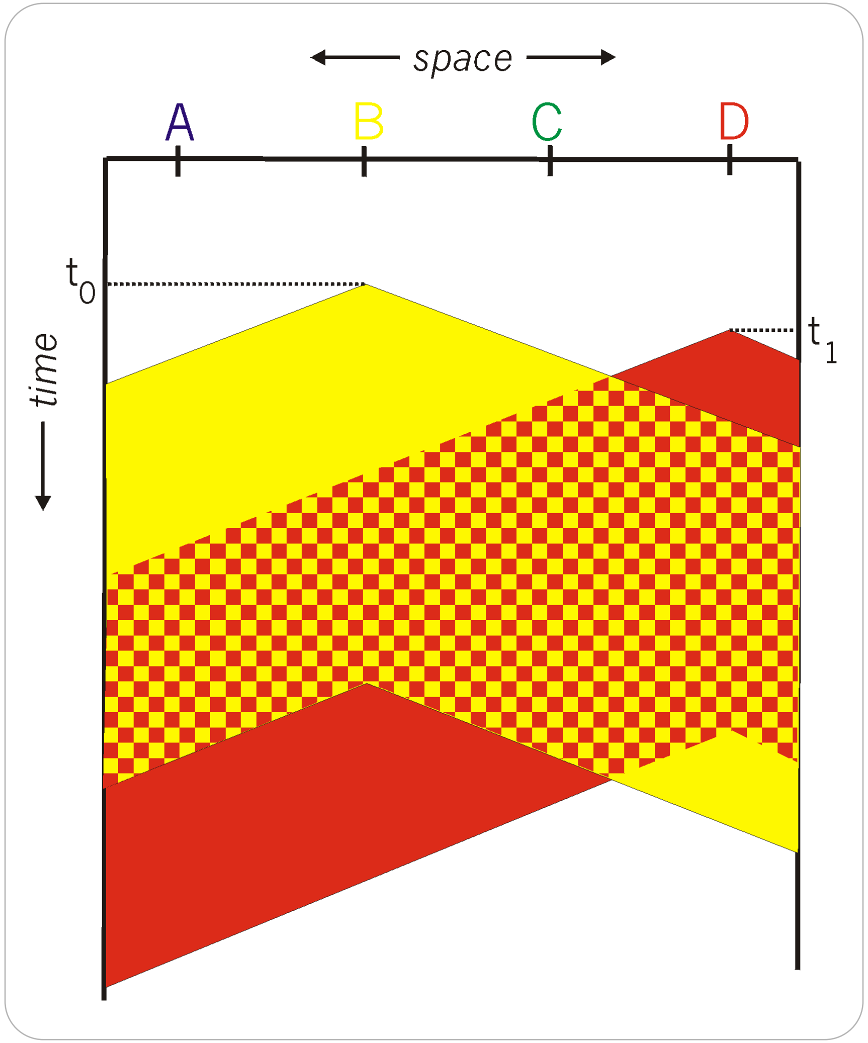
- If channel sensed idle: transmit entire packet
- If channel sensed busy, defer transmission
  - **persistent CSMA**: retry immediately with probability *p* when channel becomes idle (may cause instability)
  - **non-persistent CSMA**: retry after random interval
- Human analogy: don't interrupt others!

CSMA collisions

- Collisions can occur due to propagation delay
- Collision means: entire packet transmission time is wasted - up to **two packet times**...

NOTE

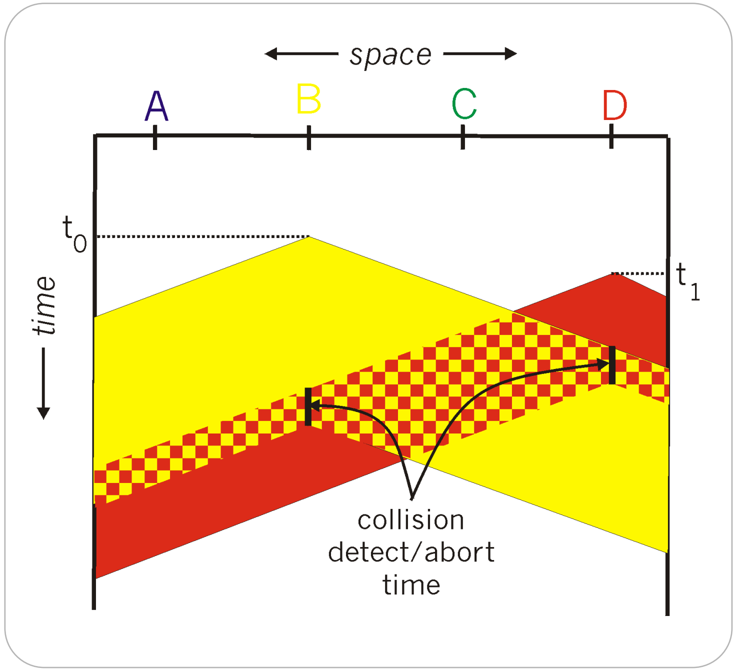
- The propagation delay determines the collision detection time!
- Don't forget the speed of light!



CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

- Collisions **detected** within short time
- Colliding transmissions aborted, reducing channel wastage
- Persistent or non-persistent retransmission



Collision detection:

- Easy in wired LANs: measure signal strengths, compare transmitted, received signals
- Difficult in wireless LANs: receiver shut off while transmitting

Human analogy: the polite conversationalist

Taking turns

Channel partitioning MAC protocols

- Share channel efficiently at high load
- Inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

Random access MAC protocols

- Efficient at low load: single node can fully utilise channel
- High load: collision overhead

Taking turns protocols

- Look for best of both worlds!

Taking turns protocols

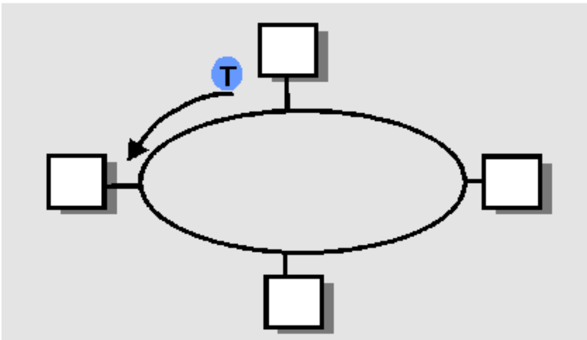
Polling

- Master node **invites** slave nodes to transmit in turn
- Request to ‘Send’ and ‘Clear to Send’ messages
- Concerns
  - polling overhead
  - latency
  - single point of failure (master)

Taking turns protocols (cont.)

Token passing

- Control **token** passed from one node to next sequentially
- Token message
- Concerns
  - token overhead
  - latency
  - single point of failure (token)



Reservation based protocols

Distributed polling

- Time divided into slots
- Begins with N short **reservation slots**
  - reservation slot time equal to channel end-end propagation delay
  - station with message to send posts reservation
  - reservation seen by all stations
- After reservation slots, message transmissions ordered by known priority