9. Shared Direct Links

Shared Media Access

Local Area Networks

Examples: Aloha, Ethernet, and Token Ring

Shared …

Problem of Shared Media Access

- How to co-ordinate multiple independent senders and receivers using a shared medium for data exchange?

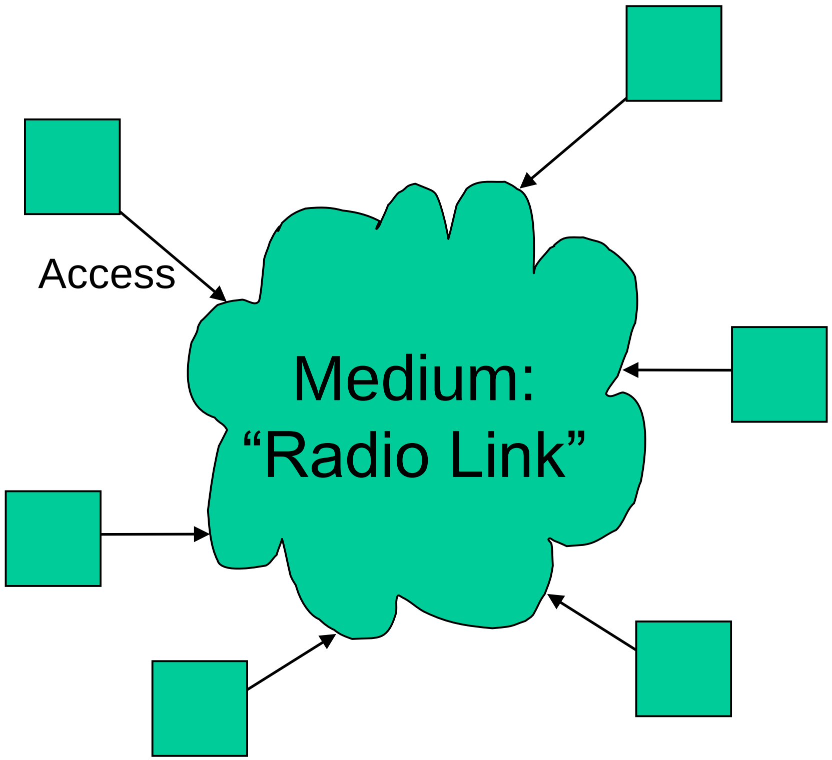
- Various solutions to be considered:
  - Distributed or centralized co-ordination
  - Pre-allocation of medium to each sender
    - Synchronous Time Division Multiple Access (TDMA)
  - Allocation of medium on demand
  - Constant frame lengths:
    - Cell-based approaches, *e.g.*, Asynchronous Transfer Mode (ATM)
  - Variable frame lengths:
    - Random access and contention-based, *e.g.*, Ethernet
    - Reservation-based, *e.g.*, Token Ring

Contention-based Access Methods

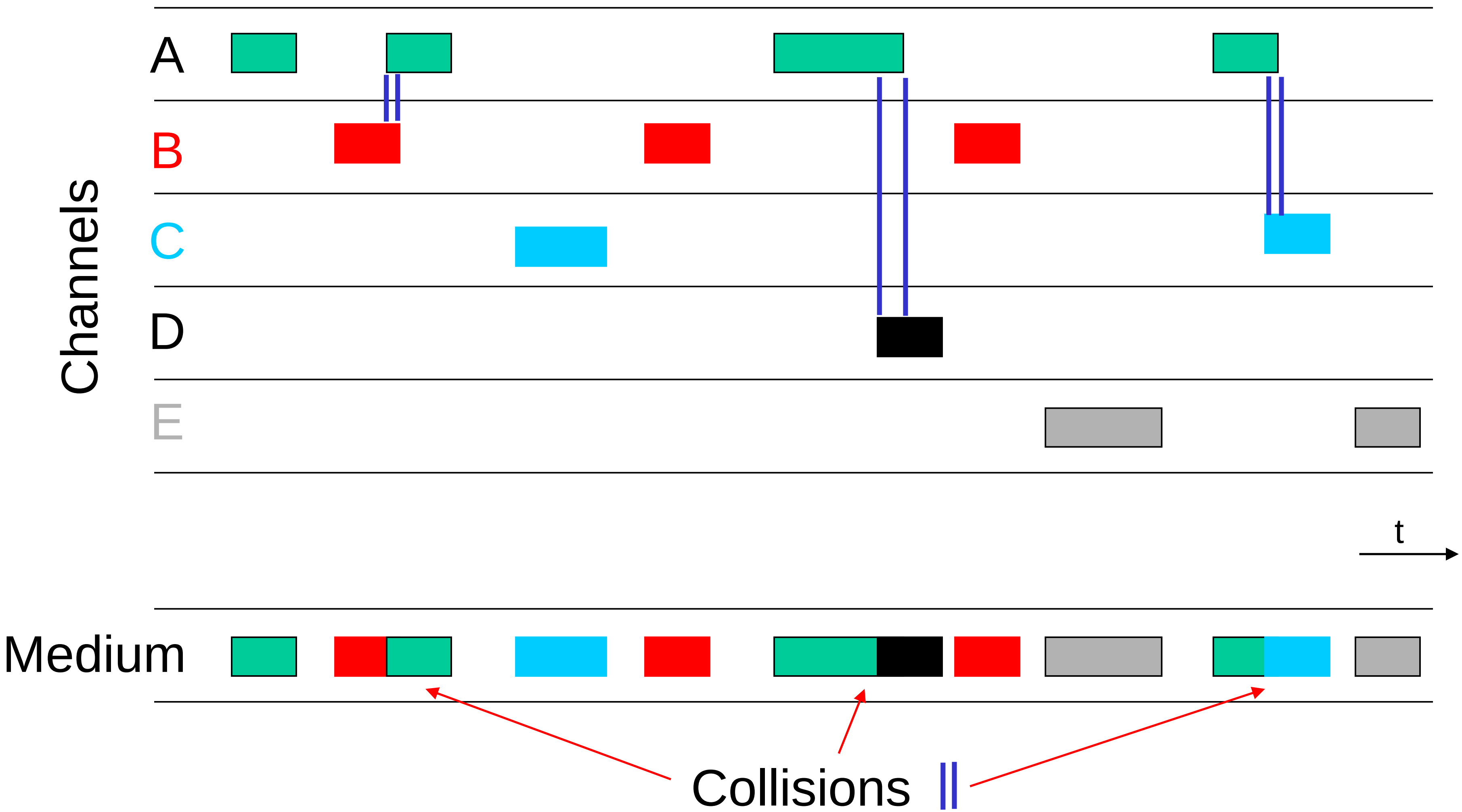
- Frames to be transmitted in a time-based multiplexing scheme on a physical medium:
  - At time *t* only a single station must be allowed to send
  - If a collision occurs, it must be detected
- Co-ordination of accessing the passive medium without any centralized control:
  - Media states: free, busy, contended
    - Collision (contention) is an interference during the sending operation
  - Sending time selected is based on local information only
  - Local observations cannot determine a media state, since electromagnetic waves travel with limited speed

Aloha

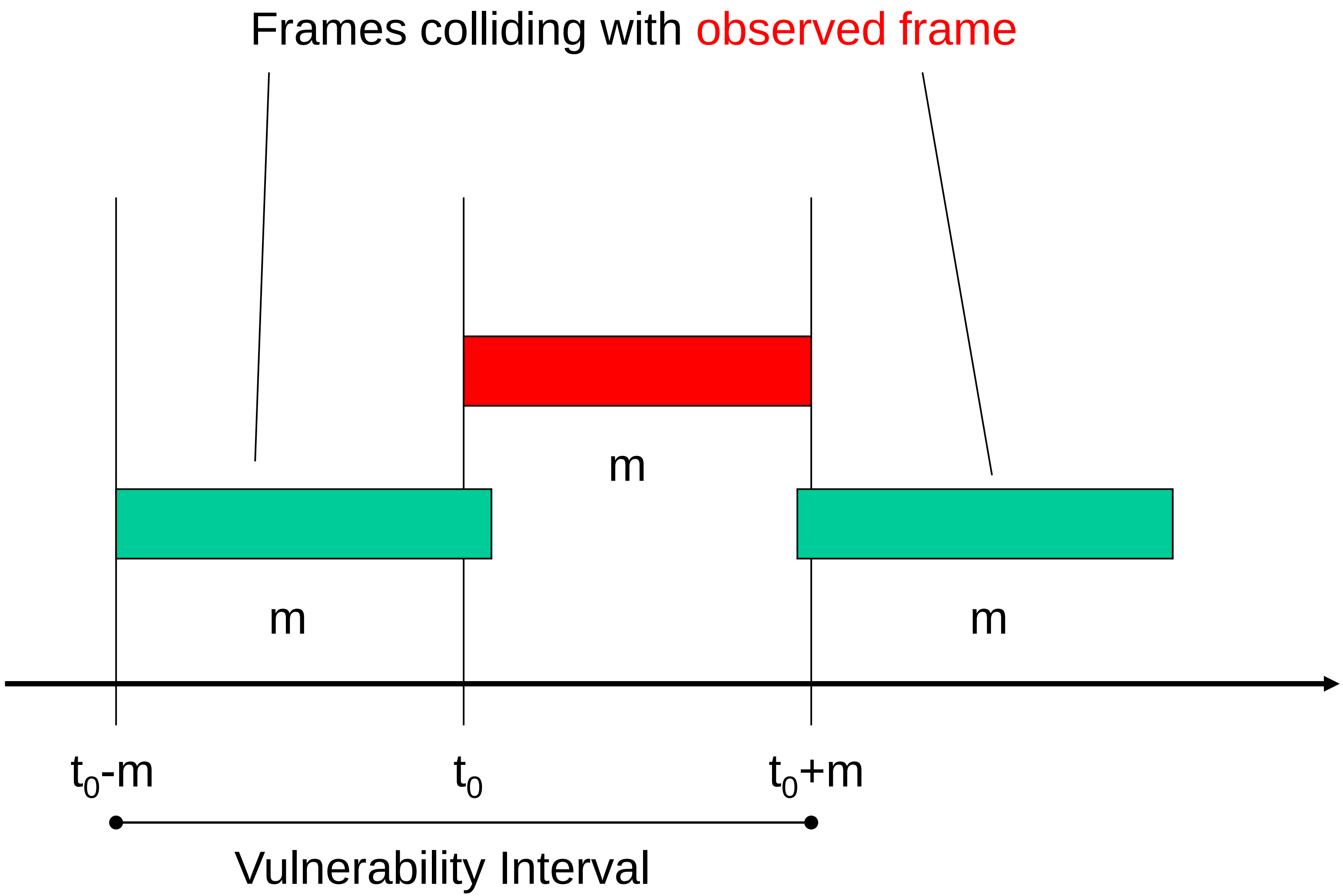
- Simplest case:
  - No state of medium considered
  - No co-ordination
- Frame with a checksum:
  - Error correction possible
- Receiver acknowledges and ignores erroneous frames
- Sender retransmits if no ACK shows up in time t (t > RTT)
- AlohaNet: Experimental and productive radio net of University of Hawaii, 1970



Aloha — Access Example

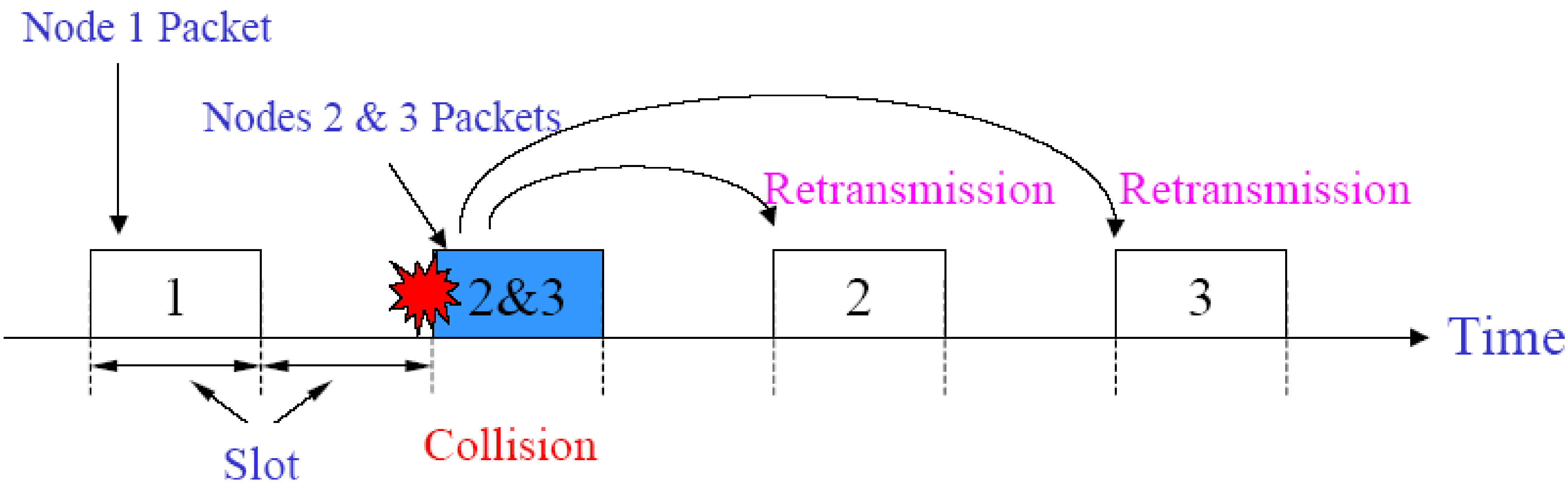


Vulnerability



Slotted Aloha

- Access scheme as within Pure Aloha, however, sending permissions only available for slot starts



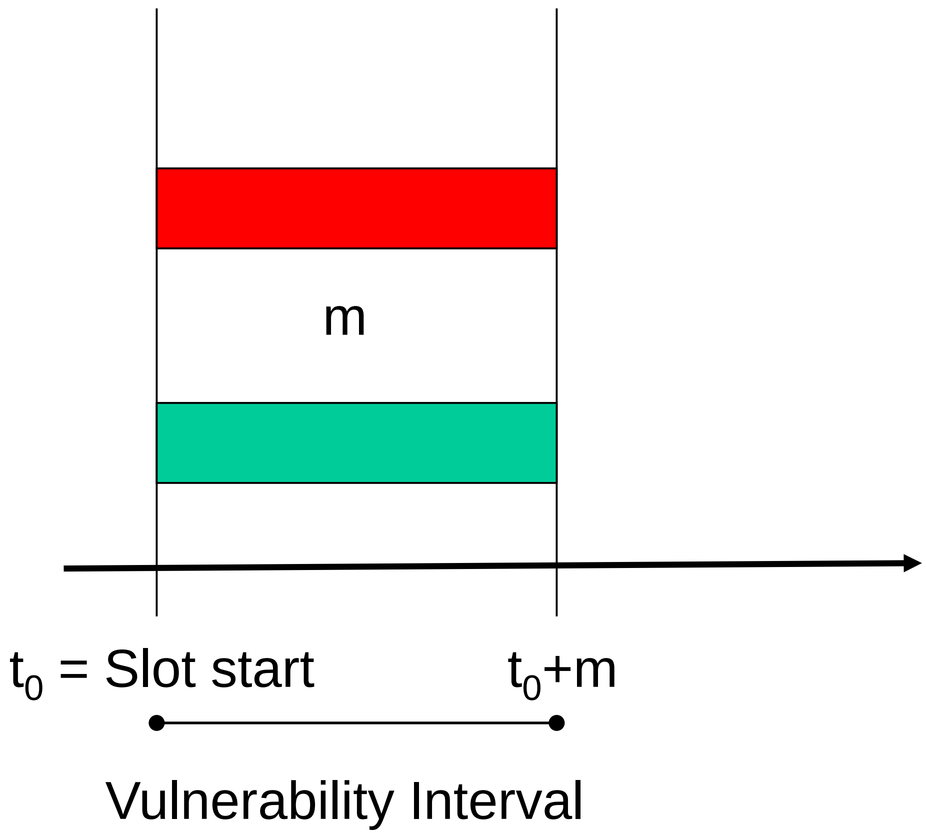
Important Issues and Access Algorithm

- Performance:
  - Function of ***f***(number of stations, traffic load, traffic characteristics, parameters of medium)
- Fairness:
  - Average/maximum time to access the medium
  - Depends on configuration and traffic
- Aloha medium access algorithm:

```
/* send one frame */
repeat
    send(data)
    r=receive()
    if r <> ack then
        wait_random_time
    end
until r=ack
```

Vulnerability

Frames colliding with observed frame



Carrier Sense Multiple Access (CSMA)

- Solving the problem of medium access:
  - Check availability first before sending data

```
/* send one frame */
Repeat
    wait_channel_available()
    send(data)
    r=receive()
    if r <> ack then
        wait_random_time
    end
until r=ack
```

Variants of CSMA

p: Probability of sending upon finding the channel idle.

Monitor the medium before accessing it

1-persistent CSMA
- free: send (probability 1)
- occ.: wait until free, then send

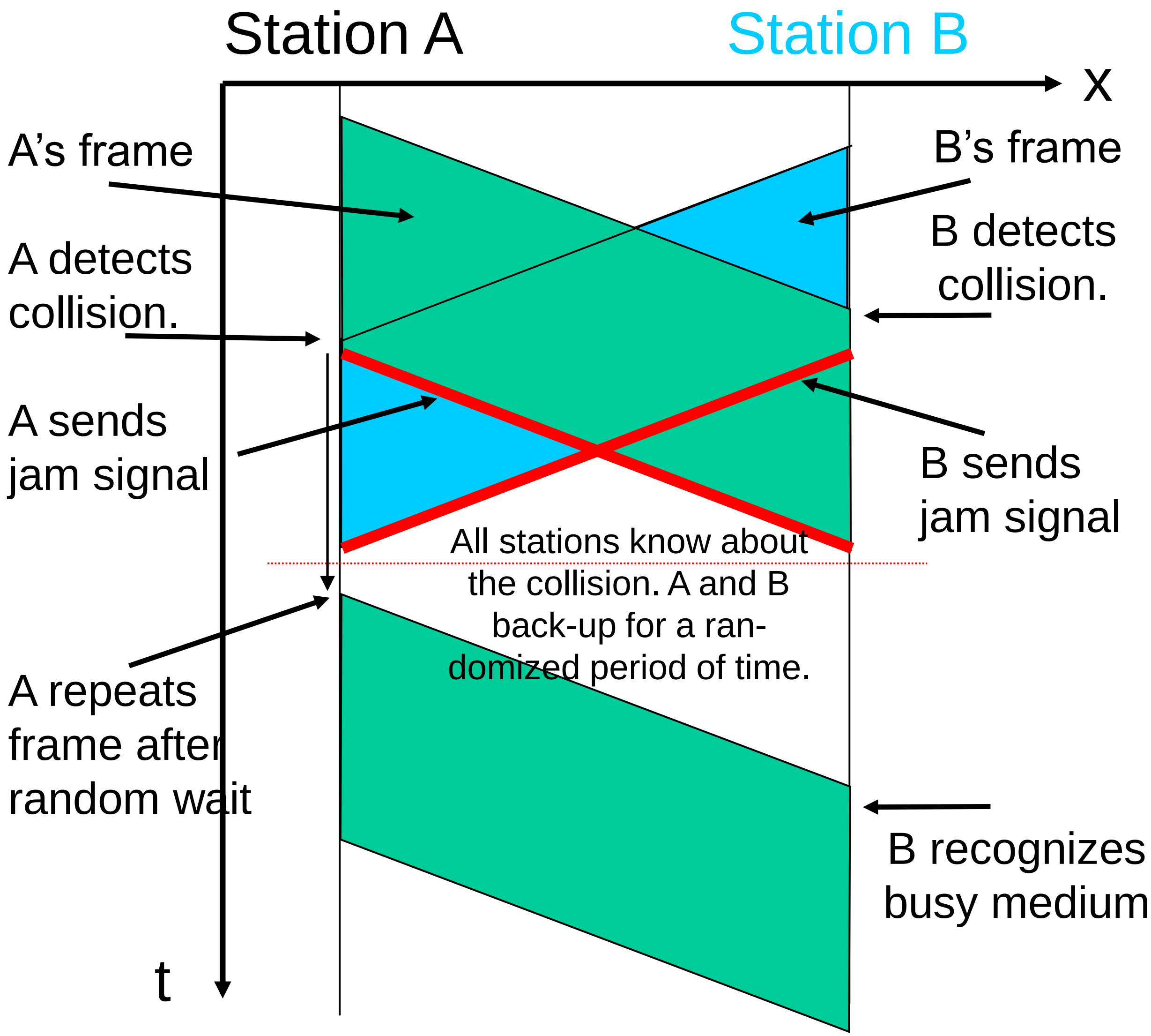
non-persistent
- free: send
- occ: wait random time, then re-test medium; send if free

p-persistent
- free: send with probability p
- occ: wait till free, then send with probability p

If transmission unsuccessful:
→ Collision
Repeat after random wait

CSMA with Collision Detection (CSMA/CD)

- Stations can detect collisions only reliably, if frame has minimum length
- Jamming signal makes sure that every station knows that the channel is in a collision state
- Random wait time increases exponentially with repeated collisions
  → *Binary Exponential Backoff*



CSMA/CD Algorithm (1)

- If line is idle:
  - Send immediately
  - Upper bound message size of 1500 byte
  - Must wait 51.2 μs between back-to-back frames
    - Numbers: Ethernet
- If line is busy:
  - Wait until idle and transmit immediately
  - Called 1-persistent (special case of p-persistent)

```
/* send one frame */
t=1
Repeat
    wait_channel_available()
    r=monitor_while_sending(data)
    if r = collision then
        wait_random_time(t)
        t=2*t
    end
until r=success
```

CSMA/CD Algorithm (2)

- Collision handling:
  - Jam for 512 bits, then stop transmitting frame
  - Minimum frame is of 64 byte length:
    - Header plus 46 byte of data

- Delay & try again in slot times (each of which 51.2 μs):
  - 1st time: uniformly distributed between 0 and 1 slots
  - 2nd time: uniformly distributed between 0 and 3 slots
  - 3rd time: uniformly distributed between 0 and 7 slots
  - Give up after several tries (usually maximum of 16)
  - Scheme termed "exponential back-off"

Performance Tuning of CSMA/CD

- The maximum throughput of CSMA-based systems is roughly indirectly proportional to β:

  $\beta = \tau / m = (\tau * C) / L$

  τ: Propagation delay [s]

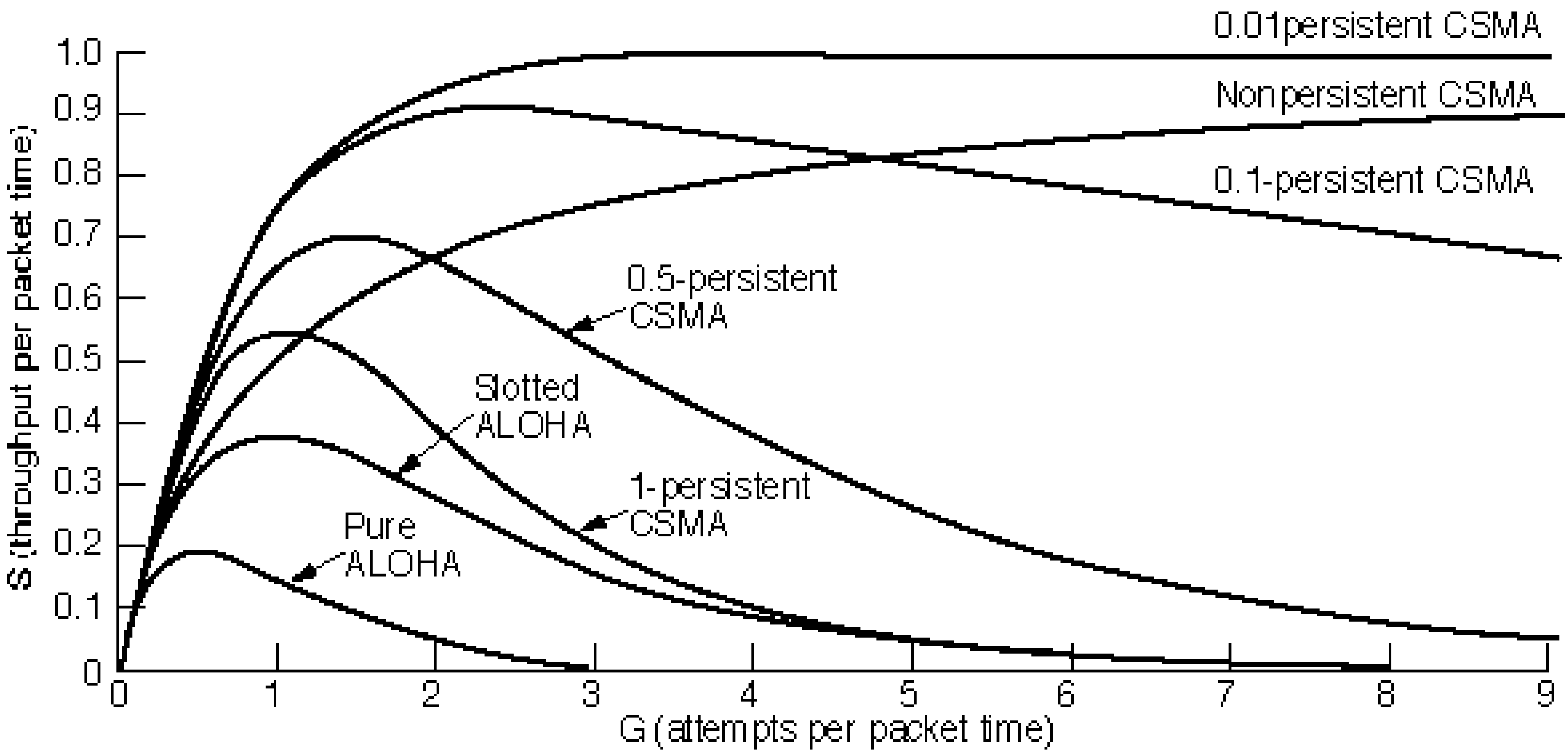
  m: Frame length [s]

  L: Frame length [bit]

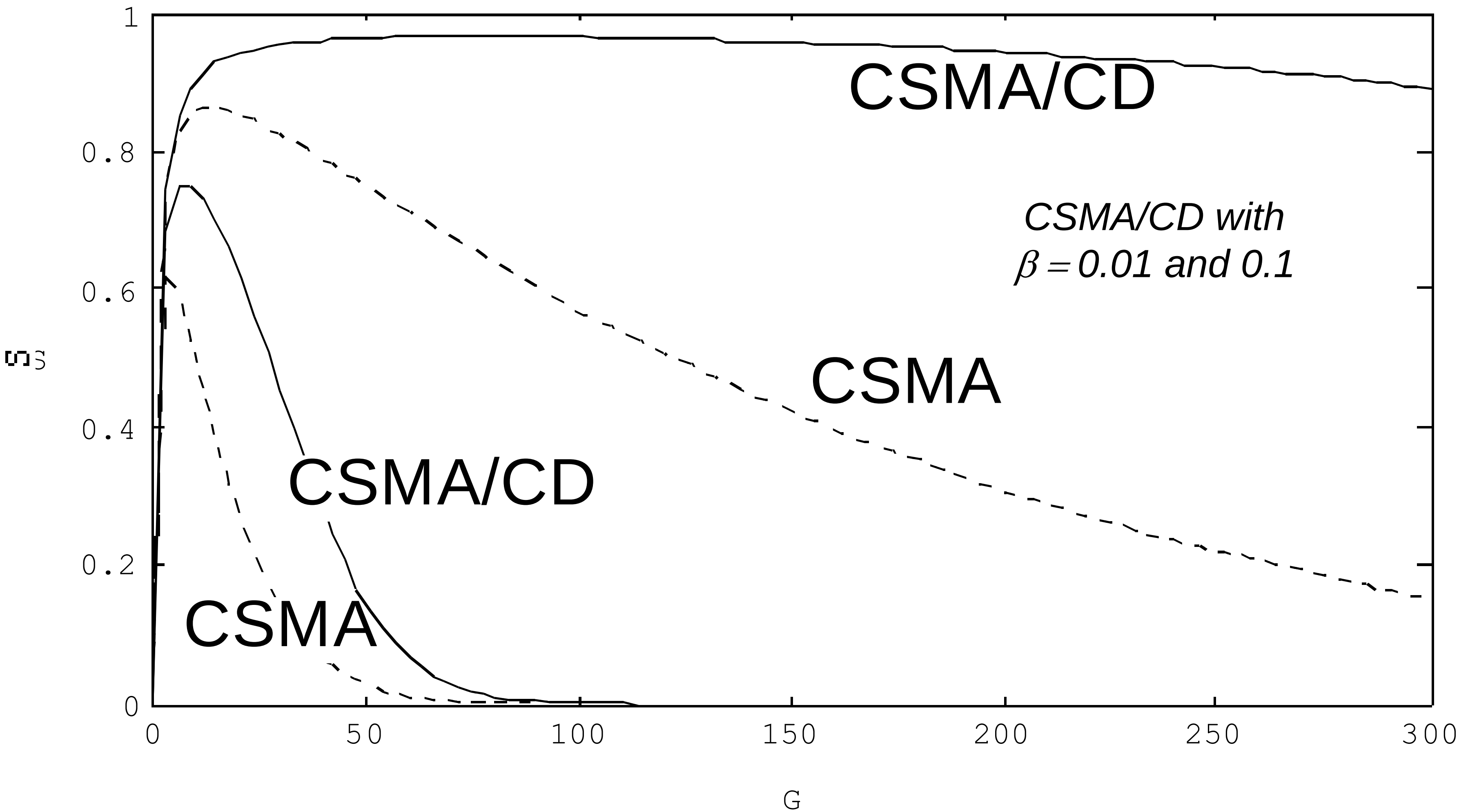
  C: Transmission rate [bit/s]

- For good performance, β should be <= 0.01.

Achievable Throughput



Throughput with Different β



CSMA/CD Frame Format

7	1	2/6	2/6	2	max. 1500	max. 46	4	(Byte)
Preamble	**SFD**	**DA**	**SA**	**Length**	**Payload**	**PAD**	**FCS**	

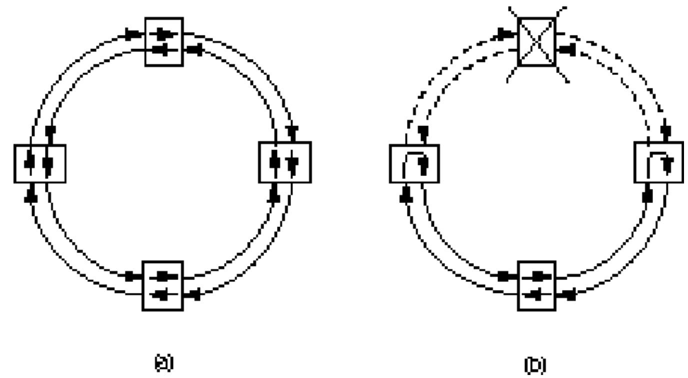
Preamble:	Bit synchronization
SFD:	Byte synchronization (Start of Frame Delimiter)
DA:	Destination address
SA:	Source address
Length:	Length of payload
Payload:	Upper layer frame
PAD:	To fill up a short frame (padding)
FCS:	32-Bit CRC for error detection (Frame Check Sequence)

Token Ring Networks

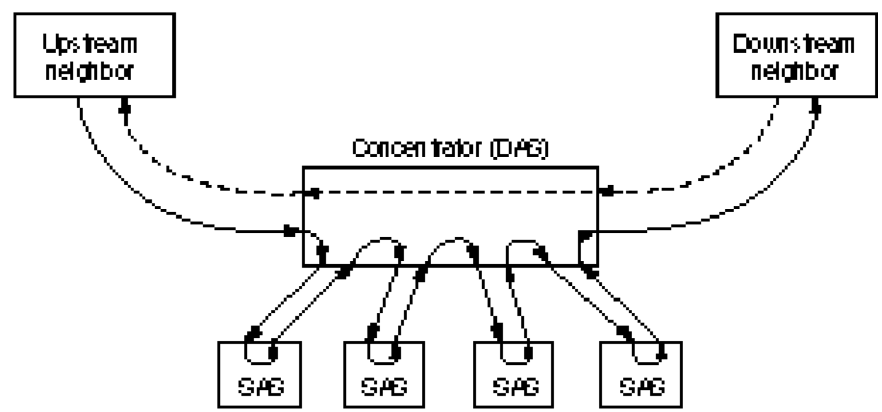
- Sharing concept:
  - Frames flow in one direction: upstream to downstream
  - Special bit pattern (token) rotates around ring
  - Hosts must capture token before allowed to transmit
  - Hosts release token after done transmitting:
    - Immediate release
    - Delayed release
  - Remove host's frame when it comes back
  - Stations get a round-robin type of service
- Examples:
  - IBM: 4 Mbit/s Token Ring
  - 16 Mbit/s IEEE 802.5 (Token Ring)
  - 100 Mbit/s Fiber Distributed Data Interface (FDDI)

FDDI

- Physical properties:
  - Dual rings
  - Single or dual attached hosts



- Each station imposes a delay (*e.g.*, 50 ns)
- Maximum of 500 stations
- Upper limit of 100 km
  - 200 km of fiber
- Uses 4B/5B encoding
- Can be implemented over copper (CDDI)



FDDI Access Algorithm

- Timed Token principle:
  - Token Holding Time (THT) determines upper limit on how long a station can hold the token
  - Token Rotation Time (TRT) determines how long it takes the token to traverse the ring:

    TRT <= Active_Nodes * THT + Ring_Latency
  - Target Token Rotation Time (TTRT) determines the agreed-upon upper bound on TRT:
    - Each node measures TRT between successive arrivals of the token
    - If measured TRT > TTRT, then token is late so don't send data
    - If measured TRT < TTRT, then token is early so OK to send data
  - Worst case: 2x TTRT between seeing the token

Token Management

- Cases of lost tokens:
  - No token when initializing ring
  - Bit error corrupts token pattern
  - Node holding token crashes
- Generating a new token (and agreeing on TTRT):
  - Execute when joining the ring or suspecting a failure
  - Each node sends a special claim frame that includes the node's bid for the TTRT
  - Upon reception of a claim frame, update bid and forward
  - If the claim frame makes it all the way around the ring:
    - This bid was the lowest
    - Everyone knows TTRT
    - This host inserts new token

FDDI Frame Format

SoF	Control	DA	SA	Payload	CRC	EoF	Status	
1	*1*	*6*	*6*	*N*	*4*	*1*	*6*	*(Byte)*

- Control Field:
  - 1st bit: asynchronous (0) versus synchronous (1) data
  - 2nd bit: 16-bit (0) versus 48-bit (1) addresses
  - Last 6 bits: de-multiplexing key:
    - Includes reserved patterns for token and claim frame
- Status Field:
  - From receiver back to sender
  - Error in frame
  - Address recognized
  - Frame accepted (flow control)