MAC Protocols

Dr. G. Omprakash

Assistant Professor, ECE, KLEF

Aim of the session

To familiarize students with the concepts and mechanisms of multiple access protocols

Learning Outcomes

At the end of this session, you should be able to:

- Demonstrate a comprehensive understanding of multiple access protocols and their applications.
- Critically evaluate the advantages and limitations of different multiple access techniques

Overview

1. Data link layer
2. MEDIUM ACCESS CONTROL (MAC) SUBLAYER
3. Random-Access
   - ALOHA
   - Carrier Sense Multiple Access (CSMA)

Recap

- **Well-defined service**
  - Unacknowledged connectionless service, Acknowledged connectionless service, Acknowledged connectionless service
- **Framing**
  - Byte count, Byte stuffing, Bit-Oriented framing (Bit stuffing), Physical layer coding violations
- **Error control**:
  - Detection - Block coding, Parity check, CRC, Checksum
  - Correction -Hamming Code
- **Flow control**:
  - Stop-and-Wait ARQ, Sliding window protocols (Go-Back-N, Selective repeat)

DLL Division

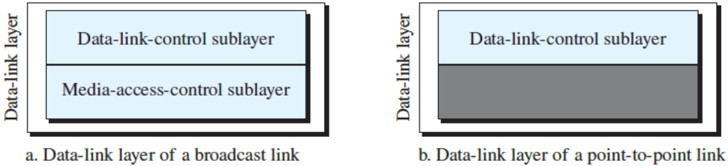


Figure: Dividing the data-link layer into two sublayers

- Data-link Control: Framing, error control and flow control
- Media Access Control: Channel allocation

Multiple-access Protocols

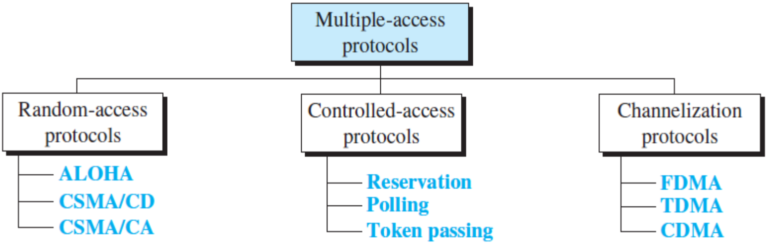


Figure: Taxonomy of multiple-access protocols

When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a **multiple-access protocol** to coordinate access to the link.

Random-Access

- Also called **Contention Methods**
- No station is superior to another station
- A station that has data to send makes a decision depending on the state of the medium
- There is no scheduled time for a station to transmit
- No rules to specify which station should send next
- Stations compete with one another to access the medium (*Contention* methods)
- ALOHA
  - Pure ALOHA, Slotted ALOHA
- CSMA
- CSMA/CD, CSMA/CA

ALOHA

- Advocates of Linux Open-source Hawaii Association (ALOHA)
- Developed at the University of Hawaii in early 1970
- It was designed for a radio (wireless) LAN
- Pure ALOHA: Time is continuous
- Slotted ALOHA: Time is divided into discrete slots into which all frames must fit

Pure ALOHA

- The original ALOHA protocol is called **pure ALOHA**
- Let users transmit whenever they have data to be sent.
- There will be collisions, of course, and the colliding frames will be damaged
- Even if 1 bit of a frame collides with 1 bit from another frame-Both will be destroyed
- The pure ALOHA protocol relies on acknowledgments from the receiver
- If the acknowledgment does not arrive after a time-out period, the station assumes that the frame has been destroyed and resends the frame.
- If the frame is destroyed, the sender just **waits a random amount of time** and sends it again.
- This time is called *backoff* time T_B.

Pure ALOHA

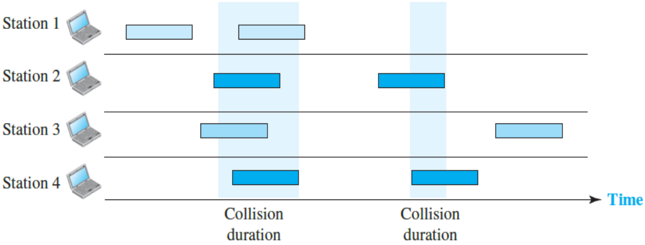


Figure: Frames in a pure ALOHA network

- The time it takes for a station to emit all of the bits of a frame onto the medium is the **transmission time**
- The **propagation time** is the time it takes for a bit to traverse the link between source and destination
- **Vulnerable time**: the length of time in which there is a possibility of collision $= 2 \times T_{fr}$ (T_{fr}: Time taken to send each frame)

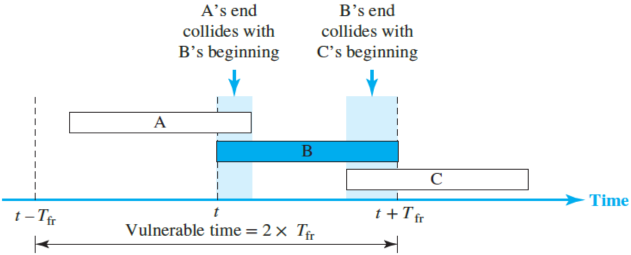


Figure: Vulnerable time for pure ALOHA protocol

Pure ALOHA

Throughput is a measure of how fast we can actually send data through a network.
The average number of successfully transmitted frames for pure ALOHA is

$$S = G \times e^{-2G}$$

where G the average number of frames generated by the system during one frame transmission time.
S_{max} is 0.184, for $G = 1/2$

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. Find the throughput for each of the following frames per second produced by the system

- 1000
- 500
- 250

The frame transmission time is 200/200 kbps or 1 ms

- 1000 frames/sec=1 frame per millisecond, G=1, $S = G \times e^{-2G} = 0.135$;
  Throughput$= 1000 \times 0.135 = 135$ frames
- G=1/2; 1/2 frame per millisecond
- S=0.184; Throughput$=500 \times 0.184 = 92$
- G=1/4; 1/4 frame per millisecond
- S=0.152; Throughput$=250 \times 0.152 = 38$

Slotted ALOHA

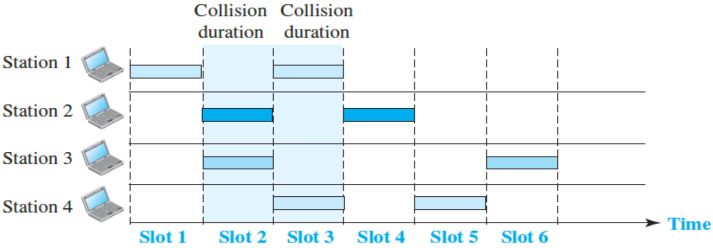


Figure: Frames in a Slotted ALOHA network

- Slotted ALOHA was invented to improve the efficiency of pure ALOHA.
- Divide the time into slots of T_{fr} seconds
- Force the station to send only at the beginning of the time slot.

Slotted ALOHA

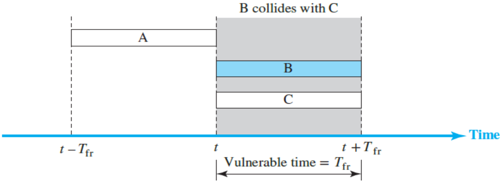


Figure: Vulnerable time for Slotted ALOHA network

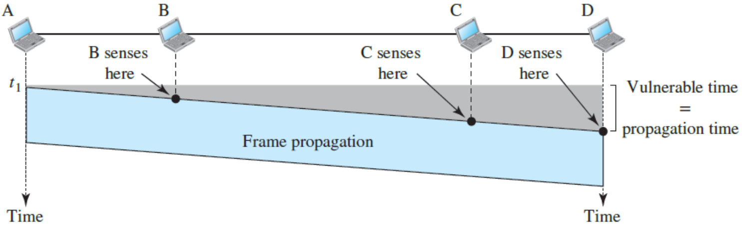
- If a station misses this moment, it must wait until the beginning of the next time slot
- Vulnerable time is T_{fr}
- Throughput $S = G \times e^{-G}$; $S_{MAX} = 0.368$ at G=1

Pure Vs Slotted ALOHA

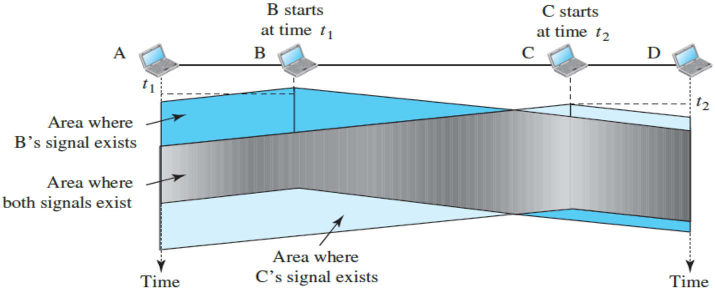
Pure Aloha	Slotted Aloha
Any station can transmit the data at any time.	Any station can transmit the data at the beginning of any time slot.
The time is continuous and not globally synchronized.	The time is discrete and globally synchronized.
Vulnerable time in which collision may occur $= 2 \times T_t$	Vulnerable time in which collision may occur $= T_t$
Probability of successful transmission of data packet $= G \times e^{-2G}$	Probability of successful transmission of data packet $= G \times e^{-G}$
Maximum efficiency = 18.4% (Occurs at G = 1/2)	Maximum efficiency = 36.8% (Occurs at G = 1)
The main advantage of pure aloha is its simplicity in implementation.	The main advantage of slotted aloha is that it reduces the number of collisions to half and doubles the efficiency of pure aloha.

Carrier Sense Multiple Access

- The chance of collision can be reduced if a station senses the medium before trying to use it.
- CSMA requires that each station first listen to the medium before sending
- CSMA can reduce the possibility of collision, but it cannot eliminate it.



CSMA



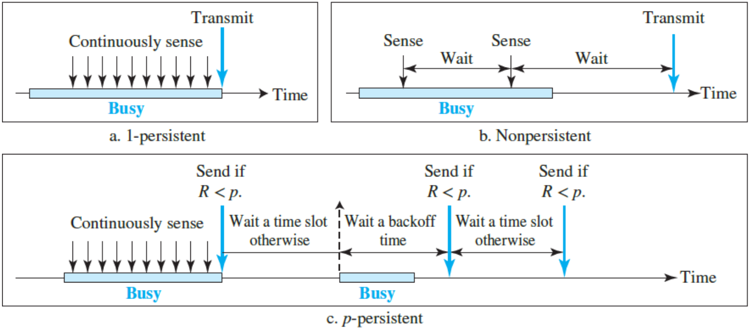
- At time t_1, station B senses the medium and finds it idle, so it sends a frame
- At time t_2 ($t_2 > t_1$), station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C.
- Station C also sends a frame. The two signals collide, and both frames are destroyed.

Persistance Methods

What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions:

- 1-persistent method
- nonpersistent method
- p-persistent method

Persistance Methods



Persistance Methods

- **1-Persistent**
  - When a station has data to send, it first senses the line
    - If the channel is idle, the station sends its data.
    - if the channel is busy, the station waits until it becomes idle.
  - If a collision occurs, the station waits a random amount of time and starts all over again
  - The protocol is called 1-persistent because the station transmits with a probability of 1 when it finds the channel idle.
  - If two stations become ready in the middle of a third station's transmission, both will wait until the transmission ends, and then both will begin transmitting exactly simultaneously, resulting in a collision

Persistance Methods

- **Nonpersistent**
  - Station that has a frame to send senses the line.
    - If the line is idle, it sends immediately
    - If the line is not idle, it waits a random amount of time and then senses the line again
  - The nonpersistent approach reduces the chance of collision
    - It is unlikely that two or more stations will wait the same amount of time to send simultaneously.
  - This method reduces the efficiency of the network
    - Because the medium remains idle when there may be stations with frames to send.

Persistance Methods

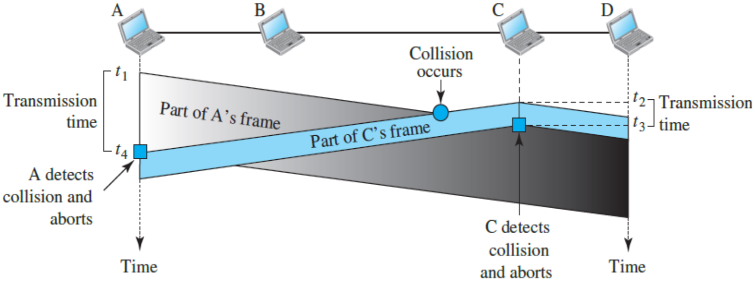
- **p-persistent**
  - This method is used if the channel has time slots
  - This approach combines the advantages of the other two strategies
    - It reduces the chance of collision and improves efficiency.
  - In this method, after the station finds the line idle
    - Generate a random number R (0-1); Send frame if $R < p$
    - With a probability $q = 1 - p$, it defers until the next slot
    - If that slot is also idle, it either transmits or defers again

CSMA with Collision Detection

If two stations sense the channel to be idle and begin transmitting simultaneously, their signals will still collide

- Another improvement is for the stations to quickly detect the collision and abruptly stop transmitting
- This strategy saves time and bandwidth.
- **CSMA/CD** is the basis of the classic Ethernet LAN

CSMA/CD


A
B
C
D
Collision occurs
t_1
Transmission time
t_4
Part of A's frame
Part of C's frame
t_2
t_3
Transmission time
A detects collision and aborts
C detects collision and aborts
Time
Time

Minimum Frame size

- For CSMA/CD to work, we need a restriction on the frame size
- Once the entire frame is sent, station may not keep a copy of the frame and does not monitor the line for collision detection
- $T_{fr} \geq 2T_p$

Acknowledge various sources for the images.

Thankyou