

# Medium Access Control: Stochastic Methods - ALOHA

In this lesson, we'll study ALOHAnet!

## We'll cover the following ^

- Additive Links On-line Hawaii Area (ALOHA)
  - ALOHAnet
  - Pseudocode
  - Slotted ALOHA
- Quick Quiz!

**Stochastic or Optimistic MAC algorithms** assume that collisions are part of the normal operation of a Local Area Network. They **aim to minimize the number of collisions**, but they **don't try to avoid all collisions**.

Stochastic algorithms are usually easier to implement than deterministic ones.

## Additive Links On-line Hawaii Area (ALOHA) #

In the 1960s, computer networks consisted of mainframes with a few dozen terminals attached to them using the telephone network or physical cables. The University of Hawaii chose a different organization though. Instead of using telephone lines to connect the terminals to the mainframes, they developed the **first packet radio technology**.

## ALOHAnet #

ALOHAnet showed that it was **possible to use radio signals to interconnect computers**. The first version of ALOHAnet, operated as follows:

- First, the terminals and the mainframe exchanged fixed-length frames composed of 704 bits. Each frame contained 80 8-bit characters, some control bits and parity information to detect transmission errors.
- Two channels in the 400 MHz range were reserved for the operation of

1. The first channel was used by the mainframe to send frames to all terminals.
  2. The second channel was shared among all terminals to send frames to the mainframe.
- As all terminals shared the same transmission channel, there was a risk of collision. To deal with this problem as well as transmission errors, the mainframe verified the parity bits of the received frame and sent an acknowledgment on its channel for each correctly received frame. The terminals, on the other hand, had to retransmit the unacknowledged frames.

## Pseudocode #

The pseudo-code below shows the **operation of an ALOHANet terminal**. We use this python syntax for all Medium Access Control algorithms described in this chapter. The algorithm is applied to each new frame that needs to be transmitted. It attempts to transmit a frame at most **max** times (while loop). Each transmission attempt is performed as follows:

1. The frame is sent. Each frame is protected by a timeout.
2. Then, the terminal waits for either a valid acknowledgment frame or the expiration of its timeout.
3. If the terminal receives an acknowledgment, the frame has been delivered correctly and the algorithm terminates. Otherwise, the terminal waits for a random time and attempts to retransmit the frame.

```
# ALOHA
N=1
while N <= max:
    send(frame)
    wait(ack_on_return_channel or timeout)
    if (ack_on_return_channel):
        break # transmission was successful
    else if(timeout):
        # timeout
        wait(random_time)
        N=N+1
    else:
        # Too many transmission attempts
```



## Slotted ALOHA #

Many improvements to ALOHANet have been proposed, and this technique, or some of its variants, are still found in wireless networks today. **The slotted technique** proposed in Roberts' 1975 paper titled "[ALOHA packet system with and without slots and capture](#)" is important because it **shows that a simple modification can significantly improve channel utilization**.

It works like so:

- Instead of allowing all terminals to transmit at any time, divide the time into slots and allow terminals to transmit only at the beginning of each slot.
- Each slot corresponds to the time required to transmit one fixed size frame.
- In practice, these slots can be imposed by a single clock that is received by all terminals. In ALOHANet, it could have been located on the central mainframe.

The analysis in Roberts's paper reveals that this simple modification **improves the channel utilization by a factor of two**.

## Quick Quiz! #

1

Slotted ALOHA improves channel utilization

☐ A) True

☐ B) False

COMPLETED 0%

1 of 2



In the next lesson, we'll look at another stochastic MAC protocol called carrier

sense multiple access.