Medium Access Control: Stochastic Methods - CSMA

In this lesson, we'll study the carrier sense multiple access protocol.

We'll cover the following

Carrier Sense Multiple Access (CSMA)

Why CSMA?

- **ALOHA and slotted ALOHA** can easily be implemented, but using them in anything but a lightly loaded network will be extremely inefficient. Designing a network for a very low utilization is possible, but it clearly increases the cost of the network.
- To overcome the problems of ALOHA, many Medium Access Control mechanisms have been proposed which **improve channel utilization**.
- Carrier Sense Multiple Access (CSMA) is one of these and is a significant improvement compared to ALOHA.

How It Works

CSMA requires all nodes to listen to the transmission channel to verify that it's free before transmitting a frame. When a node senses the channel to be busy, it defers its transmission until the channel becomes free again.

Pseudocode

Pseudocode

The pseudocode below provides a more detailed description of the operation of CSMA.

```
# persistent CSMA
N=1
while N <= max:
  wait(channel_becomes_free)
  send(frame)
  wait(ack or timeout)
  if ack:
    break  # transmission was successful
  else :
    # timeout
    N=N+1
# end of while loop
# Too many transmission attempts
```

Pseudocode: operation of CSMA terminal

The above pseudocode is often called **persistent CSMA** as the terminal will **continuously listen** to the channel and transmit its frame as soon as the channel becomes free.

Non-persistent CSMA

Another important variant of CSMA is the **non-persistent CSMA**. The main difference between persistent and non-persistent CSMA described in the pseudocode below is that a non-persistent CSMA node **does not continuously listen to the channel** to determine when it becomes free. When a non-persistent **CSMA terminal senses the transmission channel to be busy, it waits for a random time before sensing the channel again**. This improves channel utilization compared to persistent CSMA. With persistent CSMA, when two terminals sense the channel to be busy, they will both transmit (and thus cause a collision) as soon as the channel becomes free.

However, the higher channel utilization achieved by non-persistent CSMA comes at the expense of **slightly higher waiting time** in the terminals when the network is lightly loaded.

```
# Non persistent CSMA
```

```
N=1
while N <= max:
  listen(channel)

  if free(channel):
    send(frame)
    wait(ack or timeout)
    if received(ack):
      break  # transmission was successful
    else:
      # timeout
      N=N+1
  else:
    wait(random_time)
  # end of while loop
# Too many transmission attempts
```

Performance of CSMA Variants

Kleinrock and Tobagi analyzed the performance of several CSMA variants in detail. Under some assumptions about the transmission channel and the traffic, here's a table of the channel utilization of each protocol we've looked at so far.

Protocol	Channel Utilization
ALOHA	18.4%
Slotted ALOHA	36.6%
Persistent CSMA	52.9%
non-persistent CSMA	81.5%

Quick Quiz!

1 What's the difference between persistent and non-persistent CSMA?

A) Persistent CSMA constantly monitors the channel to

determine when it is free, whereas non-persistent CSMA checks the channel at random intervals.

B) Non-persistent CSMA constantly monitors the channel to determine when it's free, whereas persistent CSMA checks the channel at random intervals.

C) Persistent CSMA doesn't monitor the channel to determine when it's free whereas non-persistent CSMA checks the channel at random intervals.

In the next lesson, we'll look at an incredibly popular variant of the carrier sense multiple access protocol.