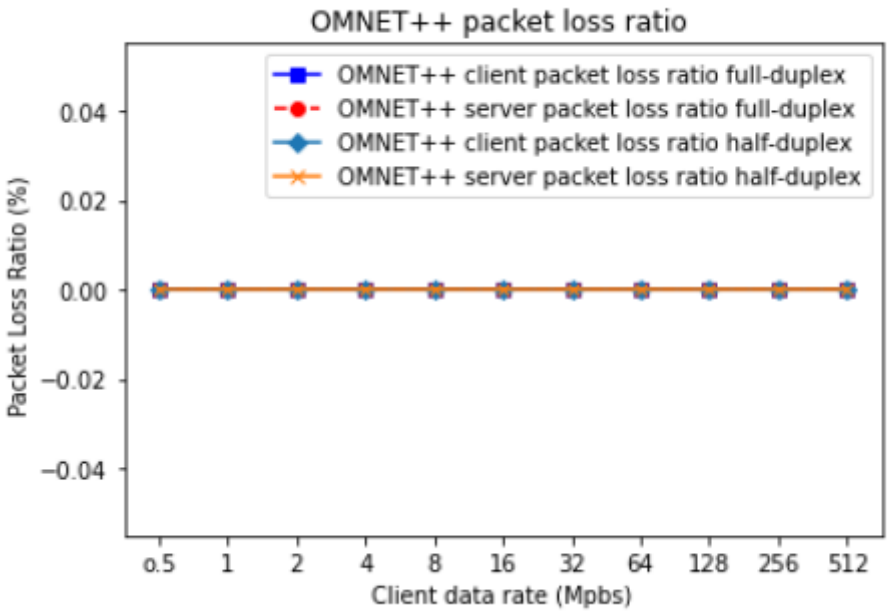
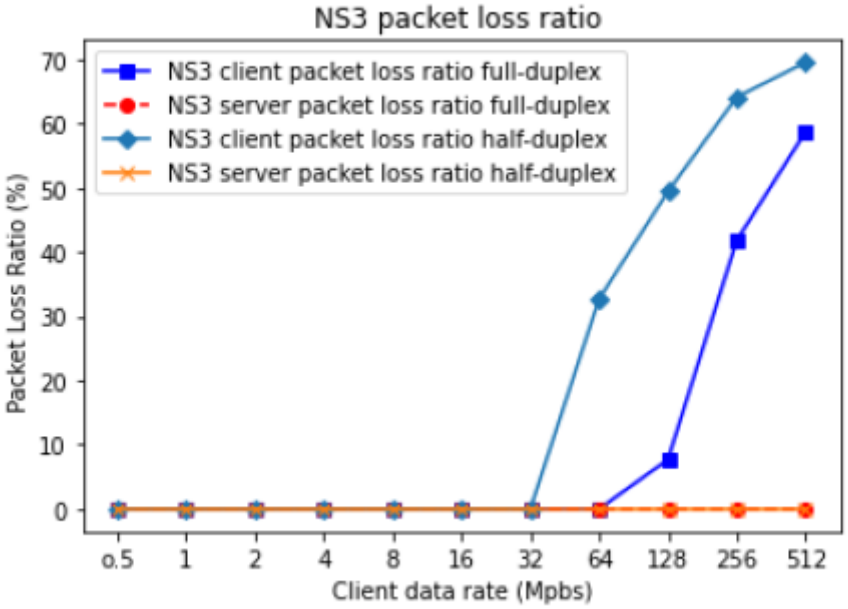
In the following you find a detailed explanation of each graph. The graphs contain their title and under each you find a paragraph of how to interpret the results.

**Packet loss:**



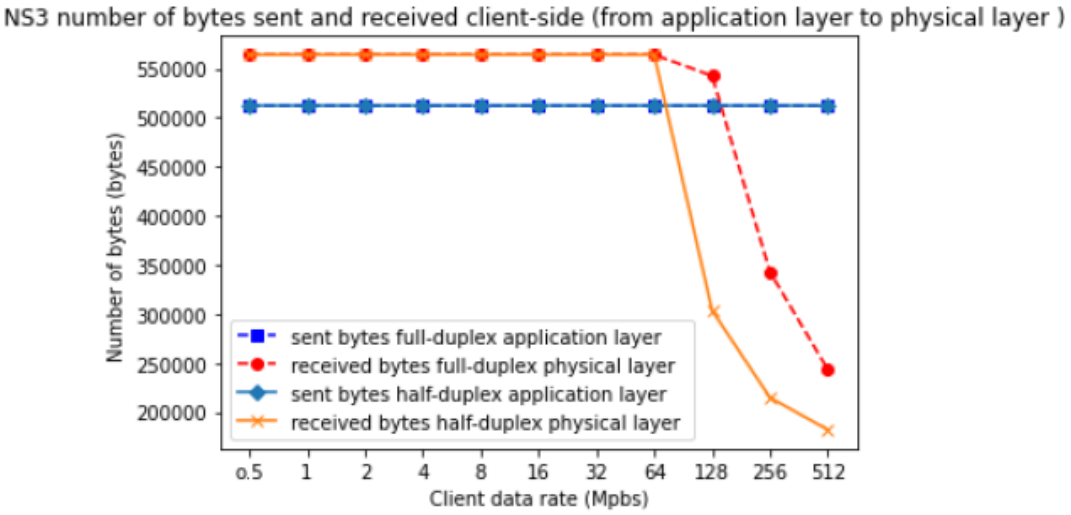

In omnet ++, there was no packet loss.
All the packets that were sent by the client were received by the server and vice versa.
So there is no packet loss in the application layer nor in the physical layer, this is why we see a flat line on 0% packet loss.

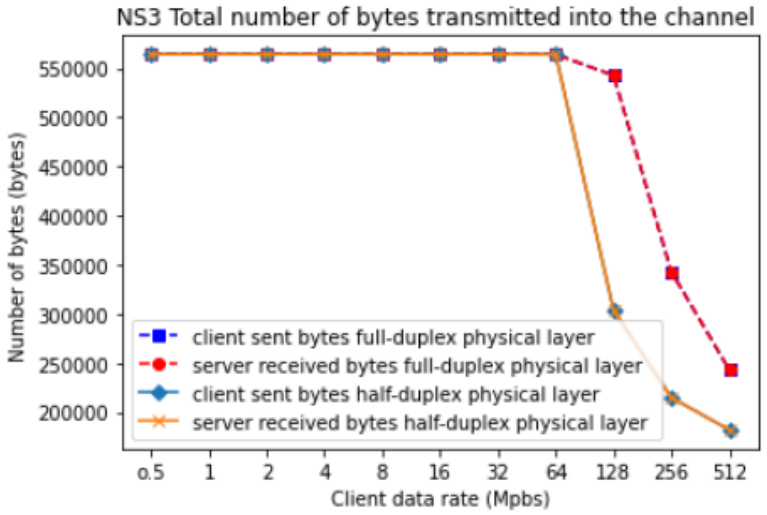



As we can see in the graph above packet loss occurs in NS3 on the client side. Both the full duplex and the half duplex modes have packet loss .

In order to understand where exactly the loss happened, we inspected the guts of the client machine, we monitored :Number of bytes sent from the application layer , the number of bytes received by the physical layer besides the Number of bytes transmitted into the channel. the next graph shows the results that we got.

We inspected the number of bytes and not packets count because the number of packets on the application layer differs from the number of packets in the physical layer due to packet fragmentation, and the addition of extra packets caused by the handshake...




From the graph we can see that for high datarates  not all packets emited by the application are put successfully into the channel in NS3

the packet loss of the client starts in half duplex at lower data rates than full duplex mode.

The duplex mode affects the packet loss because it changes the rate at which the channel can handle packets. If the channel is full duplex, it can handle a higher datarate because it is never occupied by both machines simultaneously .

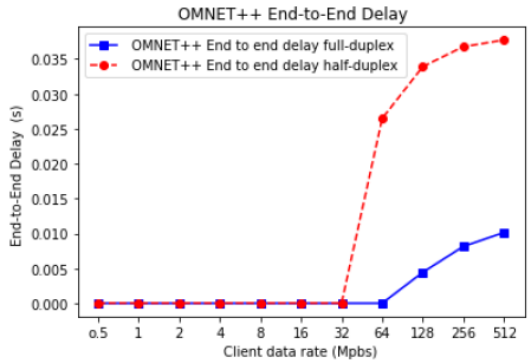
Here is a hit on why we have packet loss in NS3: In both duplex modes, packets that are emitted by the client will have to wait for extra time because the server is using the channel. if the waiting time passes a certain threshold and extra packets are being emitted by the

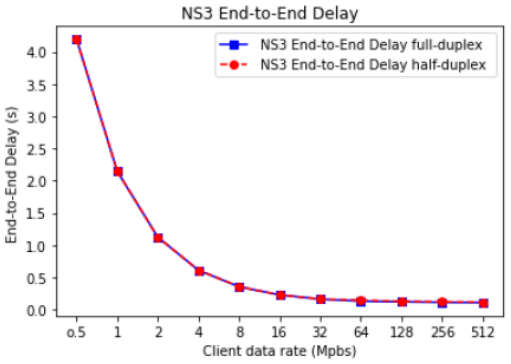
client's application there would be packet loss because the storage that is used to make the packets wait in pending state ( the buffer ) can become become full ( buffer overflow ),
.

It's easier to overflow the buffer on the in half-duplex mode, because in this mode the channel can't handle high datarates when both machines are sending packets, so some packets will have to wait in the machine before being put into the channel , and with more packet waiting in the buffer it can overflow.

This overflow occurs only in NS3. In omnet++ on the other hand, the channel of both modes can handle a huge number of packets. This difference in results is due to a difference in the implementation details of the two protocols on both simulators.

**End to End delay:**




The results show an opposite behavior for both duplex modes . NS3 has a decreasing trend. As the client data rate increases, we get less
end-to-end delay. In OMNET++, on the other hand, the end-to-end delay remains unchanged until the client data rate reaches 32 Mbps for half-duplex mode and
64 Mbps for full-duplex mode, at that point the time taken by TCP packets to be transferred increases significantly.

In comparison, NS3 shows a decreasing trend when the value of the client data rate increases. Therefore, the first impression
is that NS3 has a better end-to-end delay for a higher client data rates than Omnet++.

However, when we look into the exact values of the graphs we realize that there is more to see before we conclude, as the highest end-to-end delay in OMNET++ was
almost 0.035 seconds at a data rate of 512 Mbps. Meanwhile, the end-to-end delay
of NS3 at a low data rate was almost 4 seconds.

From the same figure, it can also be noticed that the NS3 end-to-end delay in both

modes are the same. On the other hand, TCP packets in OMNET++ consume more time in half-duplex mode than in the full-duplex mode.so this is yet another difference between the tow simulator.

The end to end delay values in NS3 decrease linearly . And this is in part due to the fact that there is increasing data loss with higher data rates.
The data loss occurs in the client machine before the packets reach the channel, which means less channel congestion and less packets to be echoed , and with less congestion comes better traffic and less delay.

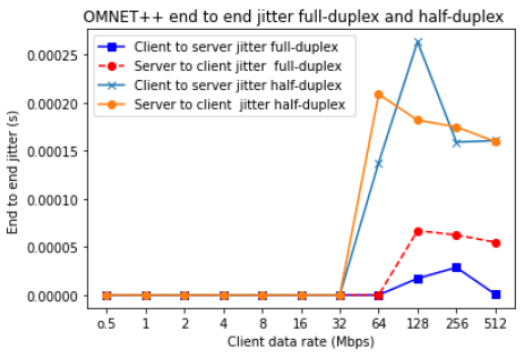
On the other hand, for Omnet++ there is no data loss so there are never less packets in traffic. and with increasing data rate, Collisions start to happen more so all packets need to wait to be sent and with more packets waiting in the client there is more delay especially in half duplex mode.

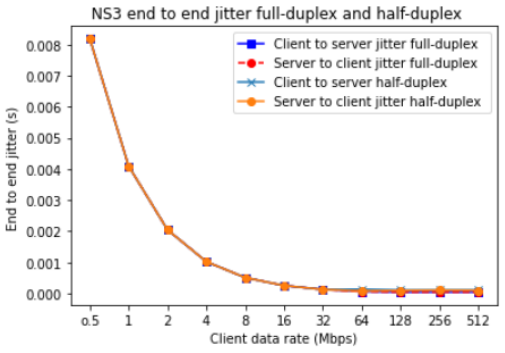
The NS3 decreasing delay can be attributed to other factors such as possibly, a faster packet forwarding through the channel when other packets are coming faster.

In Oment++  full duplex there is a very small increase in the end-to-end delay for the full-duplex at higher data rates. And we can explain this with the fact that when the client is using a higher data rate the server is overwhelmed with packets and the server needs time to send back the packets. So there's extra used time for the server to process and send them which increases the delay.

**Jitter:**

the jitter in ns3 starts at 0.008s and in omnet++ it starts at 0s
the jitter in omnet++ ends at 0.00025 wich is less than the jitter that ns3 ends with this indicates a big difference in the implementation of the two softwares, part of this difference in jitter can be explained with the difference in the number of collisions, see the collisions graphs, in NS3 there is never zero total number collisions, unlike Omnet++.




As seen in the graph above, the jitter in NS3 for both modes decreases as the client data rate increases. On the other hand, OMNET++ remains at significantly low

value, which is practically 0 s, this continues until the data rate of the client surpasses the value of 32 Mbps
in half-duplex mode and the value of 64 Mbps in full-duplex mode.

Ns3 plot starts at 0.008s and ends at 0.000114s whereas omnet++ starts at zero at ends at different values between zero and 0.00025s, this shows that the jitter values of the two simulators are more similar for higher data rates.

The increase in jitter is more significant in half-duplex mode , this is due to packet collisions, when collisions occur they cause the throughput to fluctuate which causes some packets to wait longer in the queue of the machines , which causes more delay of packets , so more collisions means more delay but collisions are not consistent that's why the delay also fluctuates resulting in higher jitter ( jitter = fluctuation in delay ) but in half duplex mode there is no collisions so there lower jitter values , this is true only for NS3 only , which indicates a difference in the implementation of the CSMA/CA algorithm in both softwares.

NS3 starts with higher jitter which means it has the least stable network with the most fluctuating end to end delay.

The jitter decreases exponentially for ns3 with increasing the rate of the client. This is because the network becomes more stable in terms of the variation of the delay . There is less variation of delay between successive packets with higher data rates in ns3.

The delay decreases consistently with higher data rates. With low delay values, there is a smaller range for potential fluctuation in the delay, thus reducing the likelihood of high jitter. Although it is possible to have high jitter with low delay, jitter values cannot be higher than the delay as jitter is the difference between two successive delays. As a result, lower delay values are associated with lower jitter. Additionally, when delay values are very low, most packets have almost the same delay, leading to low jitter. This relationship between delay and jitter is further reinforced by the fact that jitter is an indicator of network health, and networks with short delays tend to be healthier and have high and stable throughput , and a low jitter. While high jitter can occur in networks with low delays, the general trend is for jitter to decrease as delay decreases.

the jitter in omnet++ is higher for higher data rates,also it is specifically higher in half duplex mode, this is because in half duplex mode both machines are using the same channel and the network is less predictable due to collisions and the delay varies a lot between successive packets this is why the jitter is high .

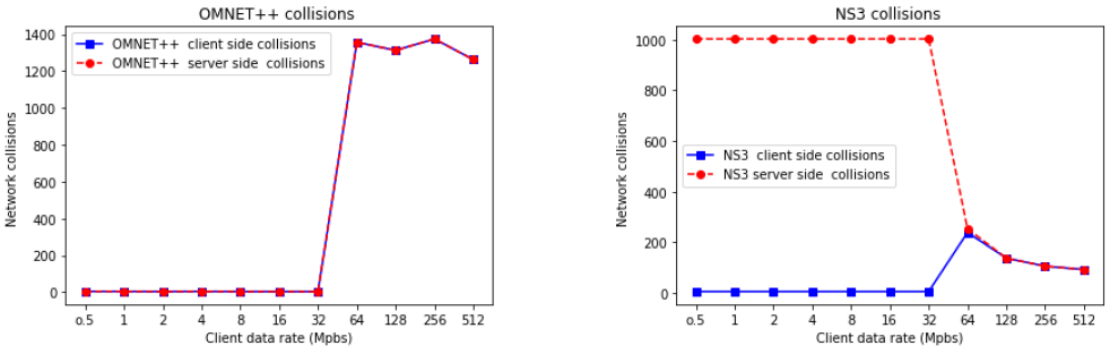
so the jitter increases exponentially for the higher data rates for half  duplex mode due to the network not being stable because of collisions and doesn't increase much for full  duplex mode because there are no collisions in that case.

In omnet++, for half duplex mode jitter  starts to increase “earlier “ than full duplex mode ( in lower data rates ). This is because in general half duplex networks are less stable due to collisions.

In full duplex mode, the jitter of the client is typically lower than that of the server. This is because the client immediately sends packets without computing, while the server must wait for packets before sending them back, thus the server's speed varies a lot. Moreover, the server's jitter may increase if the machine is busy receiving packets and can't forward previously sent packets into the channel resulting in an increase in their delay . This fluctuation in delay may be more prevalent at higher data rates. Additionally, the client's jitter may increase if it is busy receiving the echo from the server, which occurs more frequently at very high data rates. The availability of the machines is affected by the interruption of the sending process by the receiving mode.

In half duplex mode, the client experiences higher jitter in omnet++ due to constant interruption by the server. As a result, the server has a lower jitter. As the data rate increases, the jitter for both the server and client increases in omnet++. However, initially, the server's jitter increases more rapidly. This is because the server experiences more difficulty in sending data compared to the client. However, at higher data rates, the client's jitter begins to surpass that of the server. This variability in jitter is attributed to the dynamics of the network, which is sensitive to changes in configuration such as echo delay or packet size.

## collision:




During our experiments, network collisions were observed when the client and server simultaneously attempted to send packets in half duplex mode

It should be noted that, unlike OMNET++, NS3 does not include the simulation of signal jamming. As a result, collisions detected by the client machine are not visible to the server machine, and vice versa. This leads to a discrepancy in the number of collisions displayed by the client and server machines in NS3, in contrast to OMNET++, where both machines display the same number of collisions.

The number of collisions in OMNET++ starts at a specific data rate, which we can call the critical data rate. At this data rate, everything changes as the server and client start to interrupt one another leading to collisions.

In contrast, the number of collisions in NS3 is different between client-side or server-side, we are talking about the number of collisions that the machine was able to detect in its own physical layer. Collisions don't necessarily occur on the wire. When a machine is trying to access a channel and finds it busy that's also detected as collision.

In NS3, Initially, the number of collisions in the client is zero because the client is taking control over the channel, the server tries to send and finds the channel busy, it detects the collision. So every time the server tries to echo, it finds the channel busy. This is why the server detects all the collisions for low data rates.

However, starting from a critical data rate , the client and the server start to detect the same collision simultaneously. This may be because at high data rates the dynamics of the traffic changes such that both machines fail to send and both detect collisions. This can be explained by the details of the backoff algorithm, because with continuously re-occurring collisions, and after a certain threshold, the algorithm's behavior will change and this is what happens with high data rates in NS3.

It is important to note that in NS3, since it does not implement signal jamming, collisions detected by the client machine are not visible to the server machine and vice versa. This is why the client and server machines in NS3 display different collisions, unlike OMNET++, where both machines display the same number of collisions.

In omnet++ CSMA algorithm works better for lower data rates. That's why its behavior changes at extreme data rates. And the opposite is true for NS3.

**Throughput:**

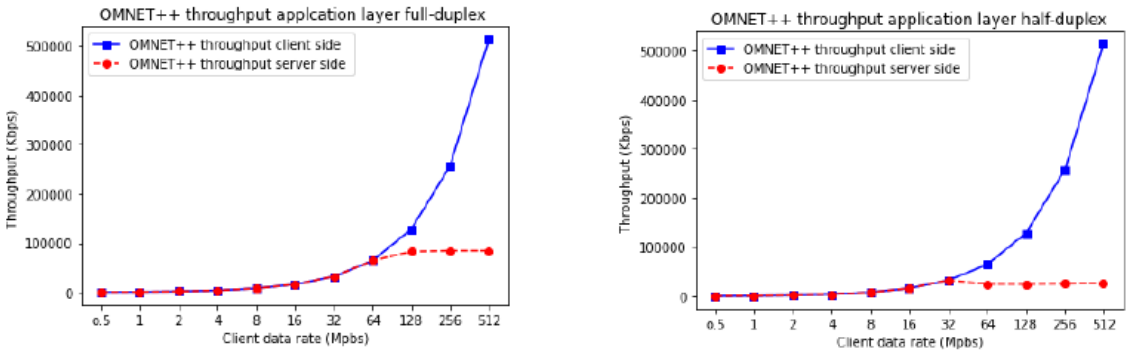



Figure 4.7: The application layer throughput of the client and server in full-duplex mode running PPP protocol and half-duplex mode running CSMA/CD protocol in OMNET++

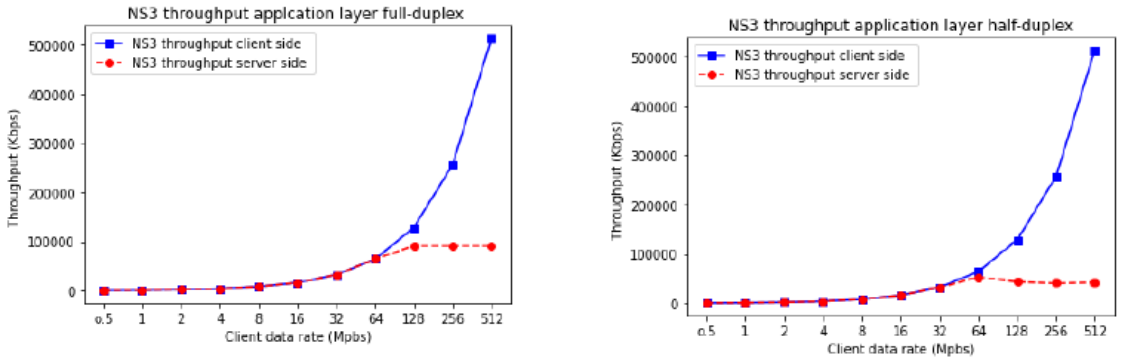



Figure 4.8: The application layer throughput of the client and server in full-duplex mode running PPP protocol and half-duplex mode running CSMA/CD protocol in NS3.

In the diagrams above , OMNET++ and NS3 application layer throughput for the client and server in both duplex modes are all proportionally similar.

The client's application layer throughput is linearly proportional to the client data rate.

On the other hand, the server application layer throughput increases along with the client application layer throughput. However, it does not increase at the same rate due to the delay between the client and the server.

The application layer throughput of the server in full-duplex mode is higher than the half-duplex mode in OMNET++ and NS3.

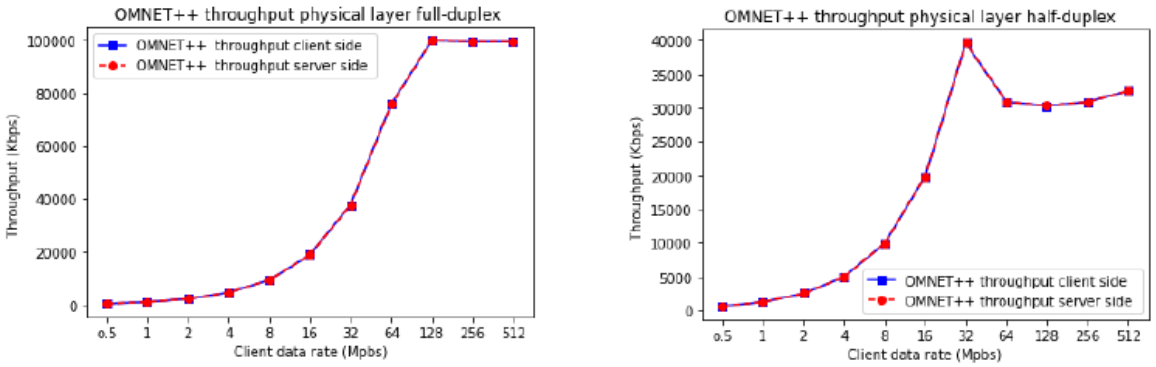



Figure 4.9: The physical layer throughput of the client and server in full-duplex mode running PPP protocol and half-duplex mode running CSMA/CD protocol in OMNET++

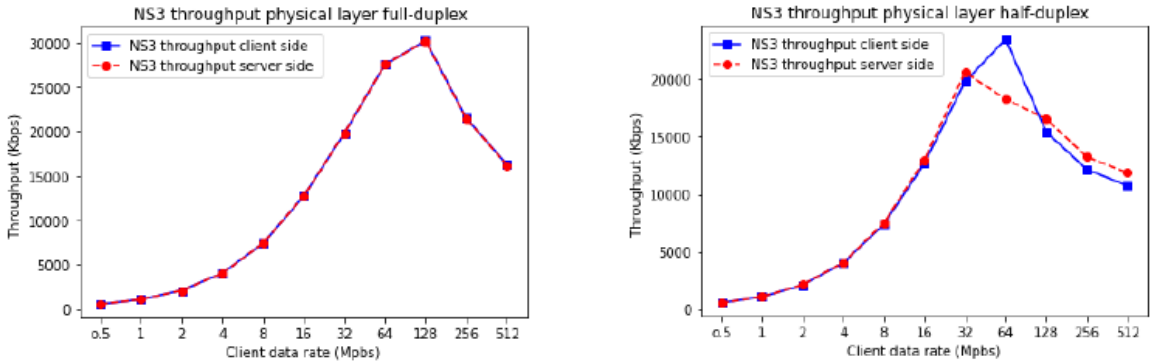



Figure 4.10: The physical layer throughput of the client and server in full-duplex mode running PPP protocol and half-duplex mode running CSMA/CD protocol in NS3.

In comparison, the throughput of the physical layer of OMNET++ is higher than NS3 at both the client and server sides in full-duplex and half-duplex modes. Therefore OMNET++ provides higher efficiency in data transmission and smooth communication at the physical layer compared to NS3.

---

throughput of the application layer for for all software's for all configuration is the same as the client is emitting more data than the server because the client doesn't have to wait for anything to emit the next packet.

But the server has to wait for  a packet to echo it. So the server is waiting for every packet to send the packet back . But the client doesn't have to do anything to start sending, this is why the client is emitting more data in the time unit. This is why throughput is high.

So for the physical layer in all the configurations on both software the physical layer of both machines has an increasing data rate for when the client data rate  is increasing.

but at a specific critical data rate  it starts decreasing because of the collision they increase too much and the  throughput is decreasing because the network lost its efficiency and this happens on both machines.
OMNET throughput  on the physical layer reaches a maximum in sometimes starts declining and sometimes starts to be stable . But in ns3 it always declines up to a certain maximum. We are talking about the throughput of the physical layer when the data rate of the client is increasing the throughput  in omnet++ reach the maximum and is almost stable but ns3 it decreases . This is because they simulated two simulators simulate the network in the physical layer differently. This is the difference between two simulators. They simulated differently they handled the algorithm of collision differently. And from that curve we see the difference.

**number of events :**

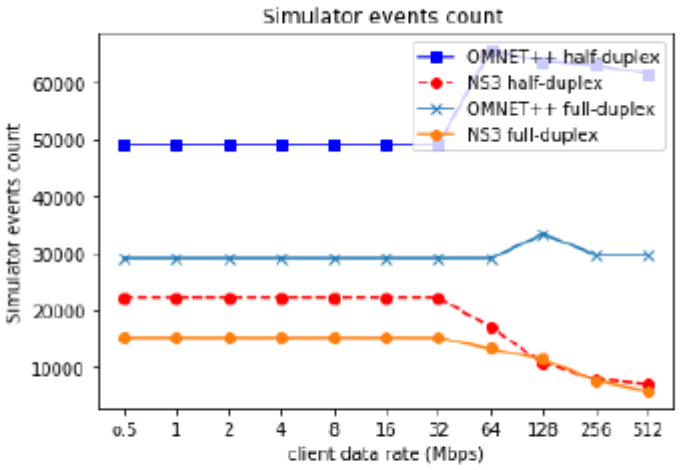



Figure 4.11: Simulator events count

The figure shows interesting compatibility between the two simulators, as the number simulation events for both are in the same order of magnitude and a narrow range.
The highest number of simulation events is for OMNET++ simulators in half-duplex mode due to the increasing number of collisions which trigger more events such as signal interference and jamming. Next is OMNET++ full duplex, which indicates that OMNET++ divides the simulations into smaller steps, which can lead to more accurate results because each step is treated separately. In comparison, NS3 performs the same scenario using lesser events.
In NS3, the number of events for half and full duplex is almost the same at higher client data rate values, which is mainly related to the decrease in packet delivery, causing lower traffic load and fewer events from the loss of packets.

On the other hand, the number of events increases in OMNET++ when the client data rate value surpasses the throughput of the client's physical layer. However, this is not the case in NS3, where the decrease in the number of collisions, the absence of jamming signals, and the decrease in traffic load at a higher client data rate which all result in less events generated.

Omnet++ generates more events to create the simulation. But ns3 generate less events in both simulators half duplex mode has more events because it causes collisions and collisions are events and they result in more action in the network by triggering the algorithm, the collision detection and condition CSMA CA CD system or CI CD algorithm and all that are extra events in both simulators,

has to flex more has more events, because it causes collisions and collisions are events and they result in even more action in the network by triggering the algorithm the collision detection and collision CSMA CA, C D, CS mercy sed algorithm and all that are extra events. This is why half-duplex mode has more events in both simulators but for net plus plus. This is a difference between the two simulators that owner plus plus generates more than

**simulation times** :

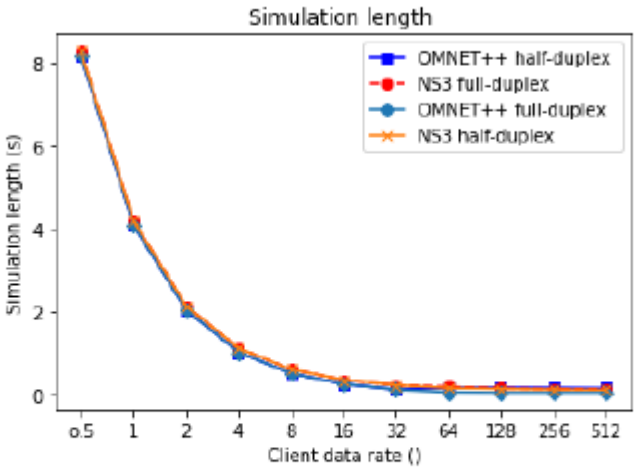



Figure 4.12: OMNeT++ Architecture

In this experiment, the time taken by the connection to close is computed in OMNET++ and NS3.
The previous figure shows the simulation length of peer-to-peer network topology in full-duplex mode running PPP protocol and half-duplex mode running CSMA/CD protocol in NS3 and OMNET++.

The most significant similarity between the two simulators is the simulation time.

The simulation time is cut in half for both simulators when the client data rate value doubles. However, except at high data rates from 16 Mbps, the simulation time does not decrease as quickly and reach a minimum value. This minimum here is mainly caused by the packet propagation delay or the time it takes for the data to travel, which cannot be eliminated by a high client data rate.

The simulation time decreases linearly as the client rate increases. This is because there is less time needed for the data transfer to occur.