INDIVIDUAL ASSIGNMENT

EE028-3-2-MCS

Modern Communication Engineering

STUDENT: PEDRO FABIAN OWONO ONDO MANGUE

ID: TP063251

INTAKE CODE: APD3F2308CE

LECTURER: ASSOC. PROF. TS. DR. RAHED MOHAMMED ABDULLAH

HANDOUT DATE: 30TH OCTOBER

HAND IN DATE: 27TH NOVEMBER 2023

WEIGHTAGE: 30%

Contents

LIST OF FIGURES:

LIST OF TABLES

INTRODUCTION

In the assignment, I had the opportunity to implement a very useful IoT application which is denominated as Smart Campus application using Netsim software.

<u>IoT SMART CAMPUS APPLICATION:</u>

Smart Campus Application is an IoT application created with the purpose of improving life in school or university. That application is being sustained through its sensors or wireless sensors that have been implemented in the schematic in Figure 1 and tested accordingly.

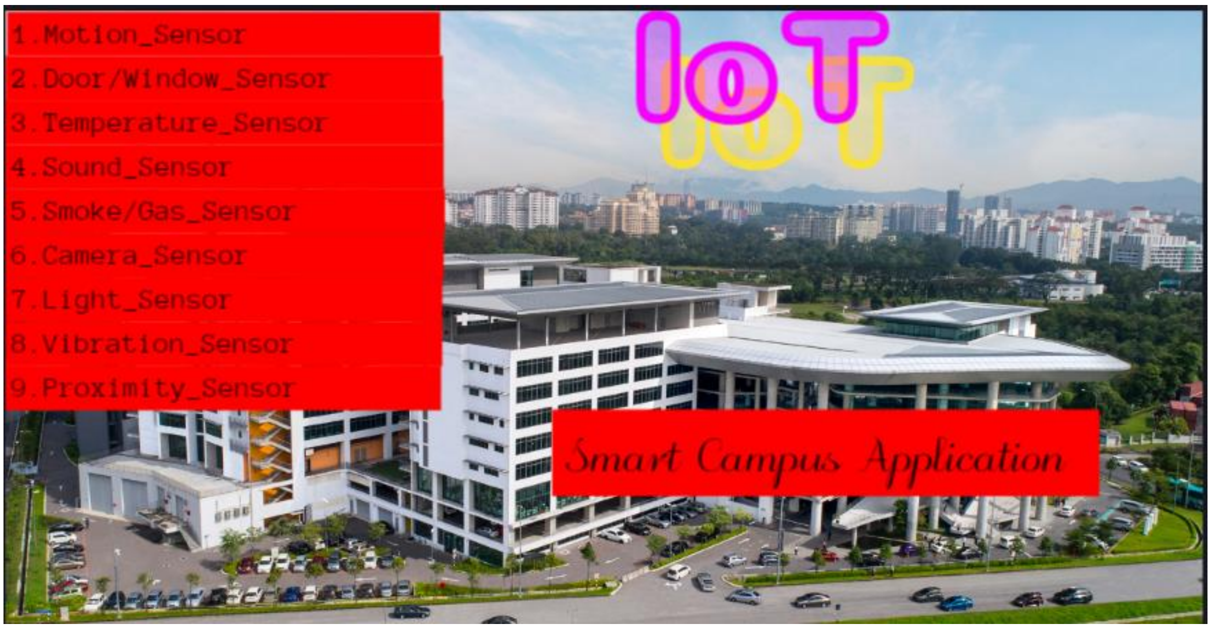


Figure 1: IoT Smart Campus application

As you can visualize in Figure 1, we have implemented 9 wireless sensors into the application and each sensor has a different function. The sensors implemented in the application are:

- Motion Sensor: It is the first sensor that I think of for the IoT Smart Campus application and its function is to detect the movement of people in a specific area of the university, which can be for example the offices or the hall area.

- Door or Window Sensor: For security purposes in any university, it is crucial to be aware of the movements of either the doors or the windows but most importantly the doors. This sensor will mainly control the status of the doors to indicate whether they are open or closed. On the other hand, it can also be used to manage or regulate the energy within the campus.

- Temperature Sensor: The temperature sensor has the function of adjusting or measuring the ambient temperature within the building, specifically in the classrooms and offices.

- Sound Sensor: It detects the level of noise within a particular area of the building such as offices, libraries, and classrooms. It is used to regulate the noise in such a way that for example, students can have classes without loud sounds.

- Camera Sensor. This sensor converts light into a digital image. It is mainly for surveillance of the whole campus.

- Smoke/Gas Sensor: This sensor can detect the presence of smoke or gases that can be harmful or toxic within the building in any area.

- Light Sensor: It is used to measure the intensity of the light depending on the time and it is correlated to the motion sensor in such a way that the light will turn on as it detects movement in that area.

- Vibration Sensor: The vibration sensor detects vibration across the building to Maintain its structure.

- Proximity Sensor: it detects the presence of objects or people within a certain proximity which means that for example if the proximity sensor detects that people are getting close to a door, it will facilitate the number of those people coming towards the door.

All these sensors are going to be connected to a central system that is called Ad-hoc which is going to be connected to a 6_Lowpan gateway that is going to assign an Internet Protocol address from version 6 to each device or wireless sensor. After that, it will be connected to a router and a wired node based on the assignment requirements.

AIMS AND OBJECTIVES

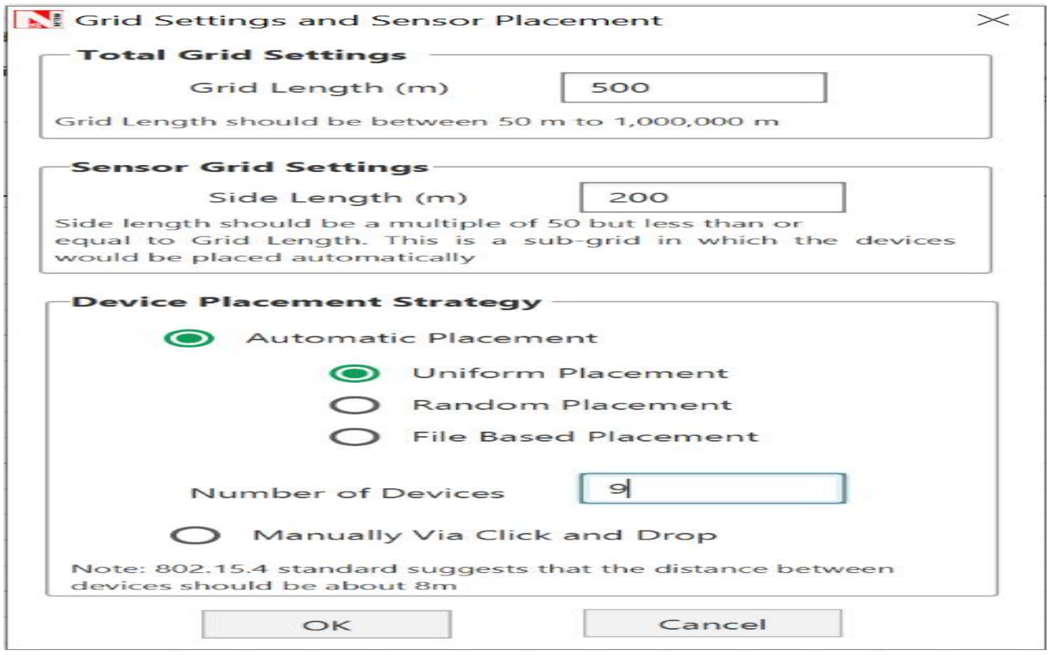
The aim of this assignment is to be able to construct an IoT application using good tools and techniques demonstrating our understanding of Modern Communication Systems.

The objectives of this assignment are :

To build an IoT network with sensors, 6_LowPAN Gateway, routers, and wired nodes.
To perform the configuration setting in the documentation or report to analyze any variation of Internet Protocol address from the source to the destination by adjusting some tables that we will get from simulation such as IP metric, UDP metric table, Application metric table, etc. Finally, we should be able to explain the graphs.

SCHEMATIC/DESIGN IMPLEMENTATION

Basically, we must open the software then after that there is one section that will ask us for the type of application we would like to implement. In that section I chose IoT application we will get this new section in Figure 2 where we might need to set the grid length of the schematic. I set the length as 500 and 200 as you can see in Figure 2. After that, I had the option of choosing the number of devices that I wanted to implement in the IoT application which are 9 devices.


Grid Settings and Sensor Placement

Total Grid Settings

Grid Length (m): 500

Grid Length should be between 50 m to 1,000,000 m

Sensor Grid Settings

Side Length (m): 200

Side length should be a multiple of 50 but less than or equal to Grid Length. This is a sub-grid in which the devices would be placed automatically

Device Placement Strategy

[✓ Automatic Placement]

[✓ Uniform Placement]

Random Placement

File Based Placement

Number of Devices: 9

Manually Via Click and Drop

Note: 802.15.4 standard suggests that the distance between devices should be about 8m

OK Cancel

Figure 2: Grid and Sensor placement settings

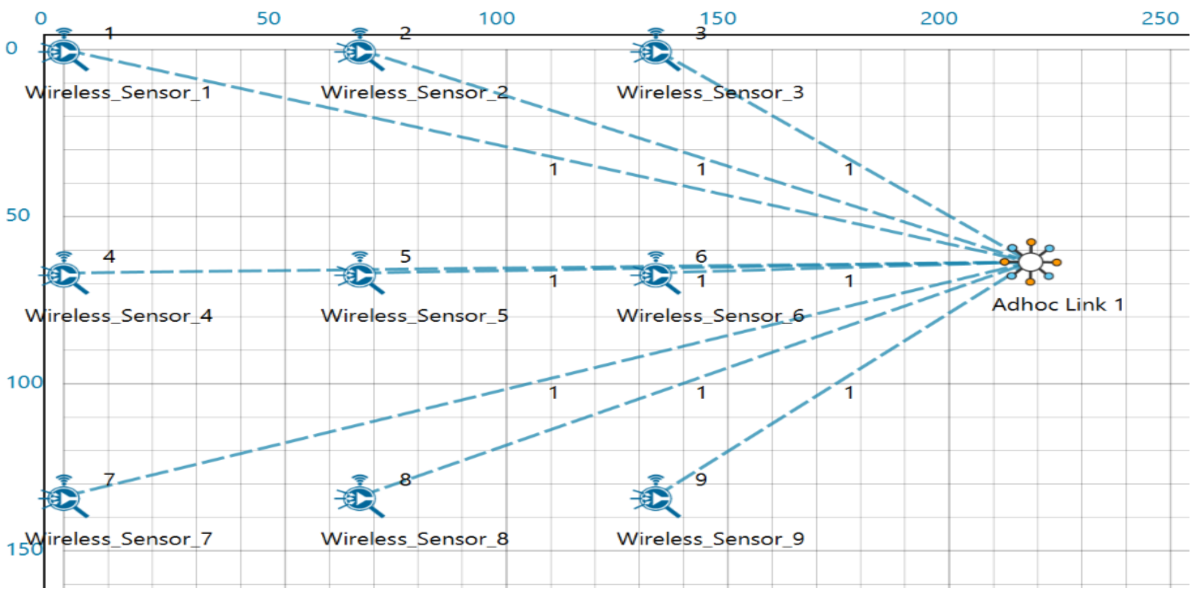


Figure 3: IoT Basic Design

Since my objective in this project is to simulate the Smart Campus IoT application, I could identify the sensor's names for my application, and I changed them accordingly based on the functionality of the Smart Campus IoT application. Now in the next Figure 4, you will have a clear picture and understanding of the application.

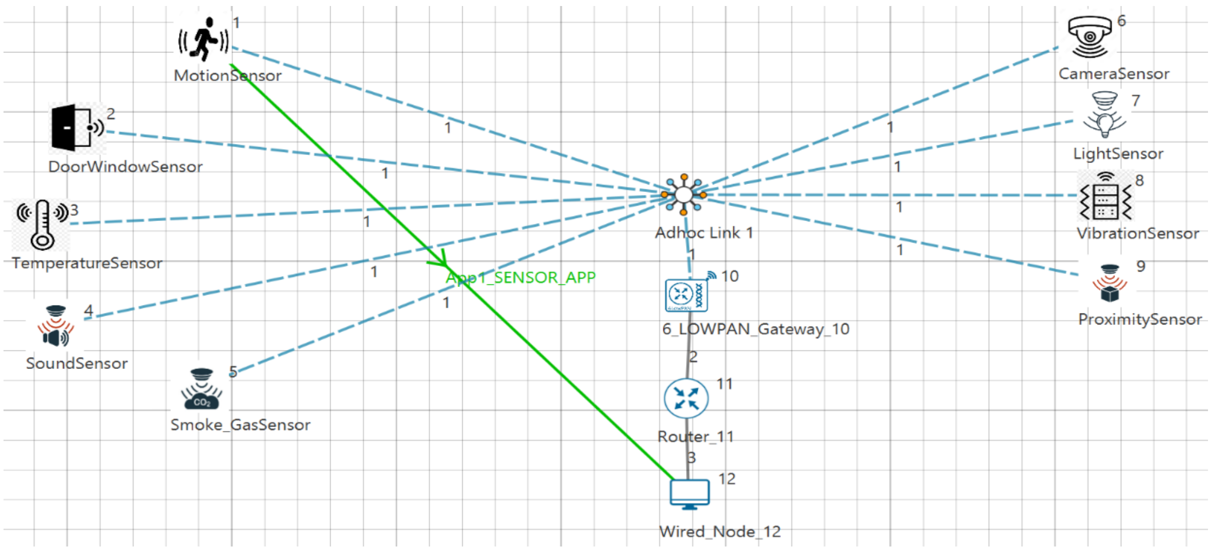


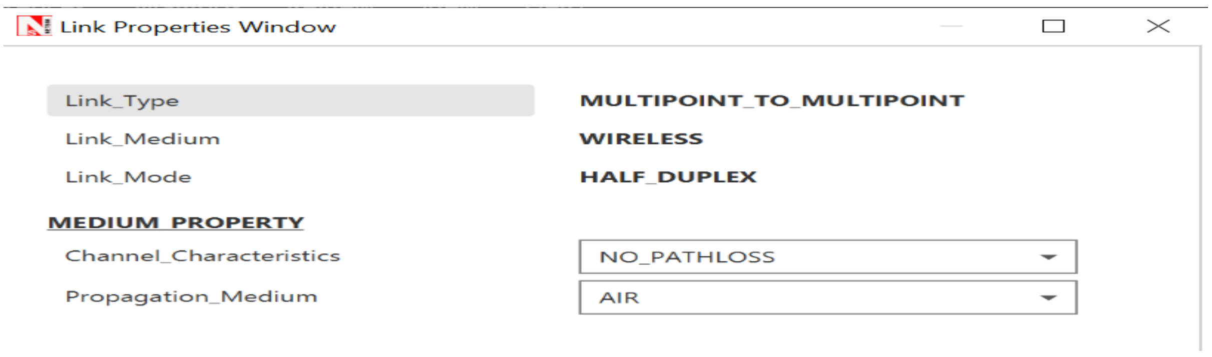
Figure 4: Smart Campus IoT Design

In Figure 4, you can observe the full representation or design of the Smart Campus IoT application with their respective icons. The nine sensors are connected to Ad-hoc which is a device-to-device communication where all the wireless sensors are connected. After that, it will go through the 6_Lowpan Gateway that contains the IPV6 that is going to be assigned to the data that has been sent that is going to go to the router and then finally to the wired node component. Moreover, you can observe that the connection between the 6_Lowpan Gateway to the Router and the Wired Node is through a wired whereas the connection of the devices towards the Ad-hoc link is wireless as well as the connection from the Ad-hoc to the 6_Lowpan Gateway.

Next, I am going to brief certain components such as the Ad-hoc link, 6_Lowpan Gateway, Router, and the Wired Node.

CONFIGURATION OF THE COMPONENTS

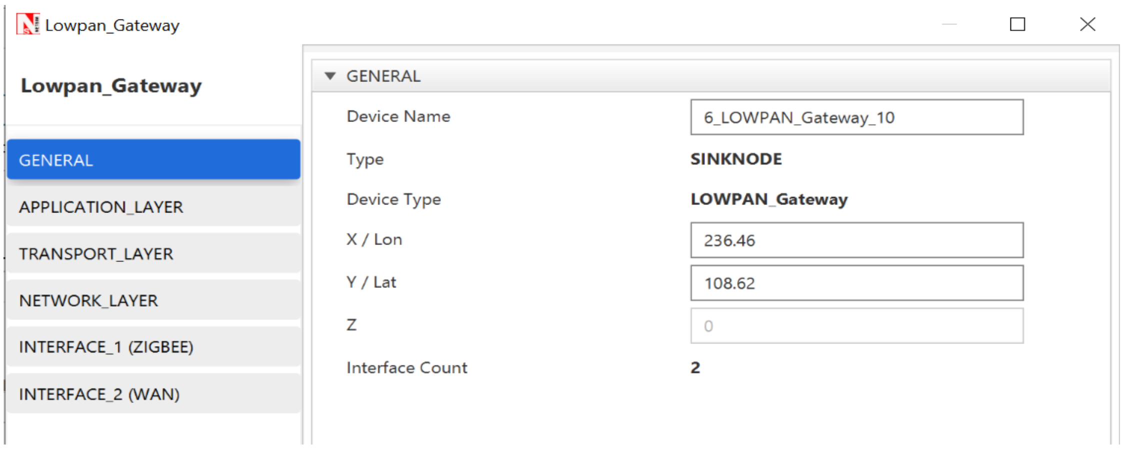
Let us observe the configuration or parameters for the Ad-hoc component.


Link Properties Window

Link_Type	MULTIPOINT_TO_MULTIPOINT
Link_Medium	WIRELESS
Link_Mode	HALF_DUPLEX
MEDIUM_PROPERTY	
Channel_Characteristics	NO_PATHLOSS
Propagation_Medium	AIR

Figure 5: Ad-hoc parameters.

As you can see in Figure 5, we have two parameters for the Ad-hoc link which are the channel characteristics and propagation medium that were set as No pathloss for channel measurement and the air as the propagation medium.


Lowpan_Gateway

Lowpan_Gateway

GENERAL

Device Name	6_LOWPAN_Gateway_10
Type	SINKNODE
Device Type	LOWPAN_Gateway
X / Lon	236.46
Y / Lat	108.62
Z	0
Interface Count	2

Figure 6: 6_Lowpan parameters

In the 6_Lowpan Gateway, we have four application layers and two interfaces which we have maintained by default in the program as well as in the router and the wired node with the difference that in the wired node we only have one interface(Ethernet).

ADVANTAGES AND DISADVANTAGES OF WIRED NODES

ADVANTAGES:

- They are more secure, and it is hard to hack.
- It facilitates the transmission of larger amounts of data due to its higher Bandwidth.
- More stable and consistent.

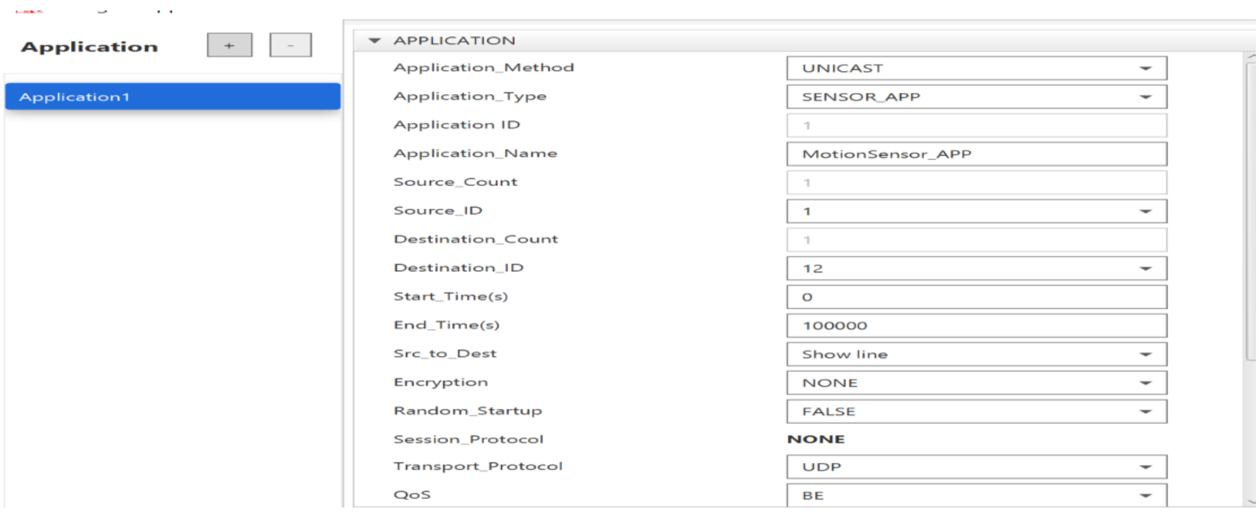
DISADVANTAGES

- They tend to be connected through a wire and that will limit their mobility.
- It is more complex in terms of installation and is very expensive.
- It requires maintenance to make sure the cables are working perfectly.

STEPS FOR RUNNING THE APPLICATION

There is one button on top of the design that is called application which is where we set all the parameters and get ready to simulate but before that, we should check the plot and packet tracer, and select all the boxes, so the simulation will be starting.

Let us observe the application configuration in Figure 7 below:



Application

Application1

APPLICATION

Application_Method	UNICAST
Application_Type	SENSOR_APP
Application ID	1
Application_Name	MotionSensor_APP
Source_Count	1
Source_ID	1
Destination_Count	1
Destination_ID	12
Start_Time(s)	0
End_Time(s)	100000
Src_to_Dest	Show line
Encryption	NONE
Random_Startup	FALSE
Session_Protocol	NONE
Transport_Protocol	UDP
QoS	BE

Figure 7: Application configuration

Based on Figure 7 I have set the application method as Unicast which means that the data frame or sensor app is going to be sensed through Motion Sensor 1 which is the only device responsible for transmitting the data frame to the wired node that is device 12. Most importantly, you can understand it by looking at this table in Figure 7 because I have selected the source ID as device 1 which is the motion sensor that is going toward the destination to device 12 which is the wired node. For a better illustration, you should refer to Figure 4.

We have the option to select more applications in such a way that we can have multiple Source IDs that are sending the information simultaneously, but we will not apply that approach as I will not set the transmission as broadcast because it is not required for this assignment.

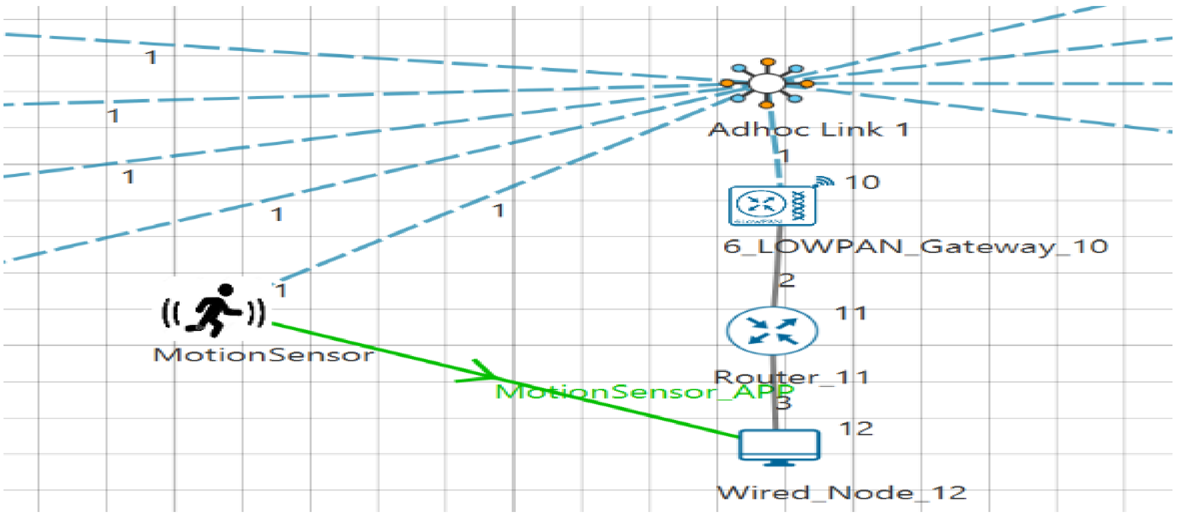
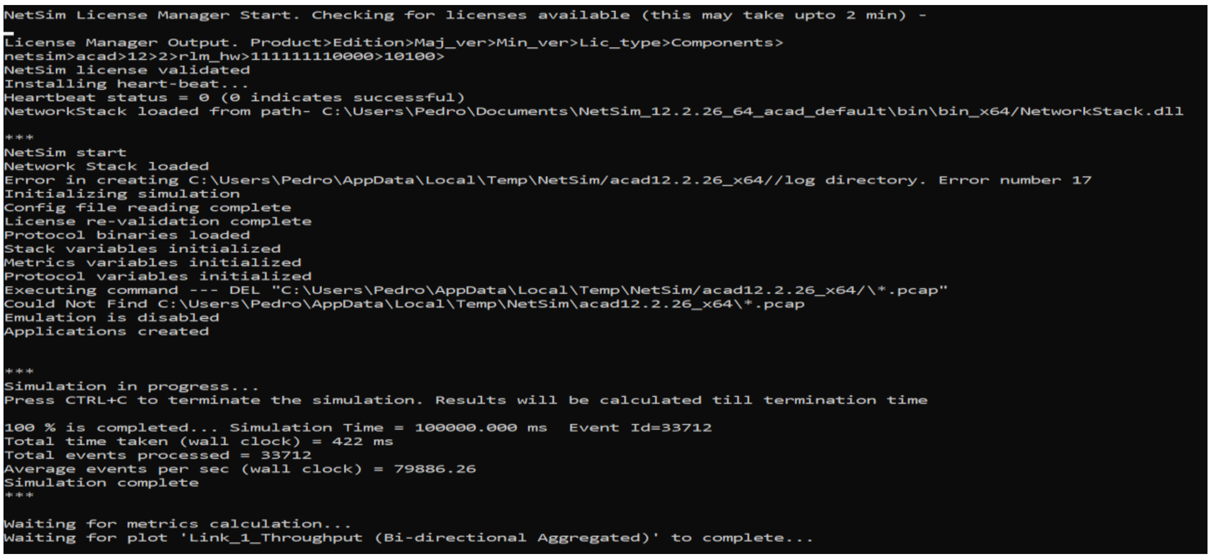


Figure 8: Application Motion Sensor

The arrow in the schematic, the green one is a communication Point-to-Point or a one-direction data transmission. In between the Motion Sensor to the wired nod

SIMULATION SETUP

For simulation, I only need to click run the program after the application component has been set for Unicast, the Source, and the Destination. In the end, we will have to choose whether we want to run the program in UDP or TCP.



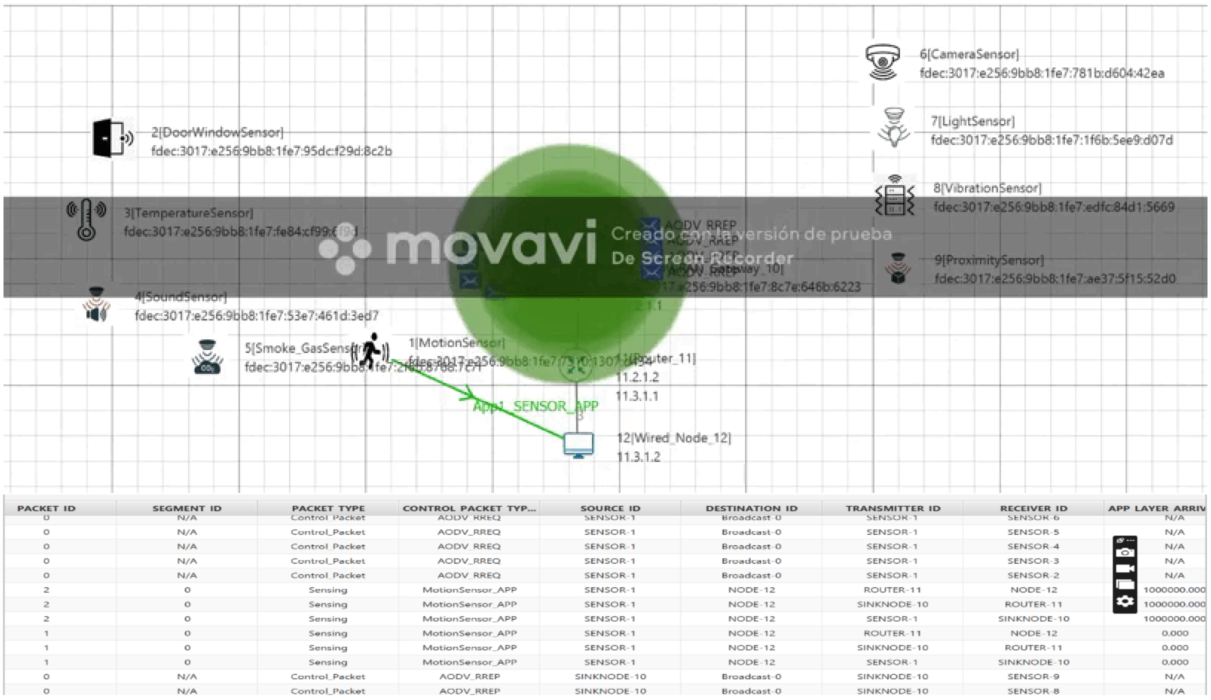
```
NetSim License Manager Start. Checking for licenses available (this may take upto 2 min) -

License Manager Output. Product>Edition>Maj_ver>Min_ver>Lic_type>Components>
netsim>acad>12>2>rlm_hw>111111110000>10100>
NetSim license validated
Installing heart-beat...
Heartbeat status = 0 (0 indicates successful)
NetworkStack loaded from path- C:\Users\Pedro\Documents\NetSim_12.2.26_64_acad_default\bin\bin_x64/NetworkStack.dll

***
NetSim start
Network Stack loaded
Error in creating C:\Users\Pedro\AppData\Local\Temp\NetSim/acad12.2.26_x64//log directory. Error number 17
Initializing simulation
Config file reading complete
License re-validation complete
Protocol binaries loaded
Stack variables initialized
Metrics variables initialized
Protocol variables initialized
Executing command --- DEL "C:\Users\Pedro\AppData\Local\Temp\NetSim/acad12.2.26_x64/\*.pcap"
Could Not Find C:\Users\Pedro\AppData\Local\Temp\NetSim\acad12.2.26_x64\*.pcap
Emulation is disabled
Applications created


***
Simulation in progress...
Press CTRL+C to terminate the simulation. Results will be calculated till termination time

100 % is completed... Simulation Time = 100000.000 ms  Event Id=33712
Total time taken (wall clock) = 422 ms
Total events processed = 33712
Average events per sec (wall clock) = 79886.26
Simulation complete
***

Waiting for metrics calculation...
Waiting for plot 'Link_1_Throughput (Bi-directional Aggregated)' to complete...
```

Figure 9: Loading Process.

In Figure 9 we can observe the loading process for the application. After this process we will be able to get the tables and the graph of the application which is going to be shown accordingly.

SIMULATION RESULTS

PACKET ANIMATION

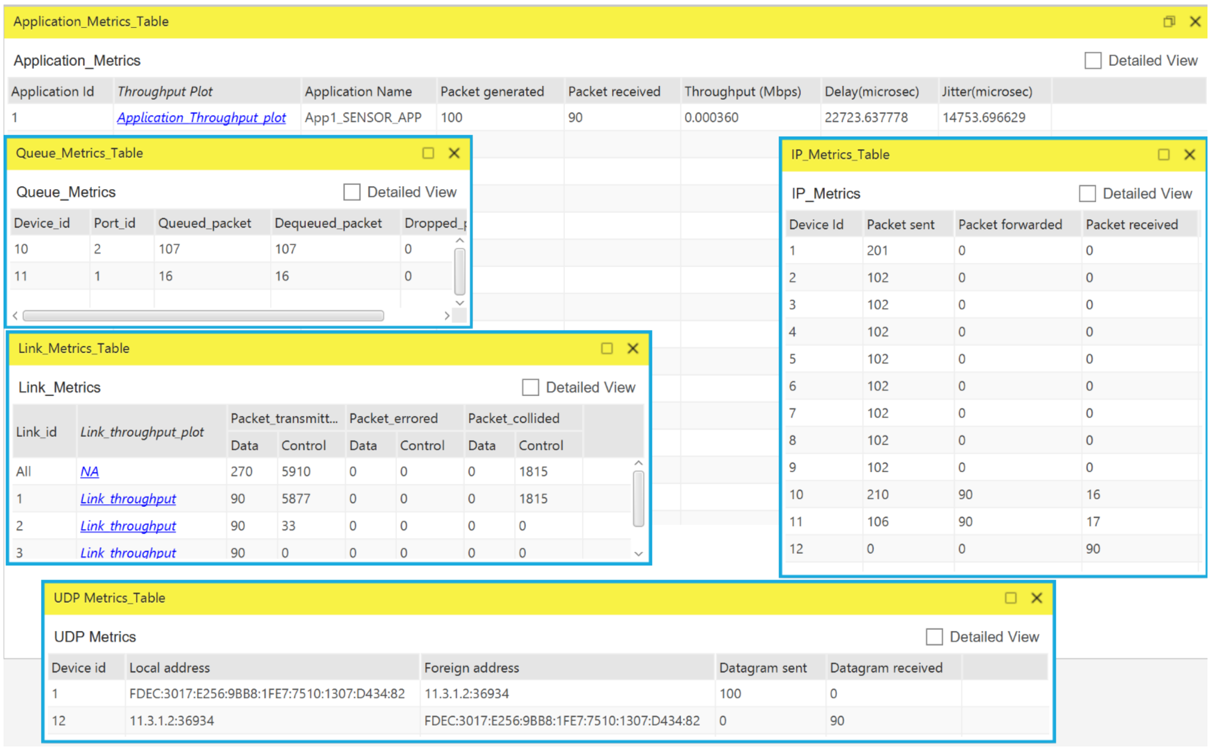

6[CameraSensor]
fdec:3017:e256:9bb8:1fe7:781b:d604:42ea
7[LightSensor]
fdec:3017:e256:9bb8:1fe7:1f6b:5ee9:d07d
2[DoorWindowSensor]
fdec:3017:e256:9bb8:1fe7:95dc:f29d:8c2b
8[VibrationSensor]
fdec:3017:e256:9bb8:1fe7:edfc:84d1:5669
3[TemperatureSensor]
9[ProximitySensor]
fdec:3017:e256:9bb8:1fe7:ae37:5f15:52d0
AODV_RREP
4[SoundSensor]
fdec:3017:e256:9bb8:1fe7:53e7:461d:3ed7
5[Smoke_GasSensor]
1[MotionSensor]
11[Router_11]
11.2.1.2
11.3.1.1
App1_SENSOR_APP
12[Wired_Node_12]
11.3.1.2

PACKET ID	SEGMENT ID	PACKET TYPE	CONTROL PACKET TYP...	SOURCE ID	DESTINATION ID	TRANSMITTER ID	RECEIVER ID	APP LAYER ARRIV
0	N/A	Control_Packet	AODV_RREQ	SENSOR-1	Broadcast-0	SENSOR-1	SENSOR-6	N/A
0	N/A	Control_Packet	AODV_RREQ	SENSOR-1	Broadcast-0	SENSOR-1	SENSOR-5	N/A
0	N/A	Control_Packet	AODV_RREQ	SENSOR-1	Broadcast-0	SENSOR-1	SENSOR-4	N/A
0	N/A	Control_Packet	AODV_RREQ	SENSOR-1	Broadcast-0	SENSOR-1	SENSOR-3	N/A
0	N/A	Control_Packet	AODV_RREQ	SENSOR-1	Broadcast-0	SENSOR-1	SENSOR-2	N/A
2	0	Sensing	MotionSensor_APP	SENSOR-1	NODE-12	ROUTER-11	NODE-12	1000000.000
2	0	Sensing	MotionSensor_APP	SENSOR-1	NODE-12	SINKNODE-10	ROUTER-11	1000000.000
2	0	Sensing	MotionSensor_APP	SENSOR-1	NODE-12	SENSOR-1	SINKNODE-10	1000000.000
1	0	Sensing	MotionSensor_APP	SENSOR-1	NODE-12	ROUTER-11	NODE-12	0.000
1	0	Sensing	MotionSensor_APP	SENSOR-1	NODE-12	SINKNODE-10	ROUTER-11	0.000
1	0	Sensing	MotionSensor_APP	SENSOR-1	NODE-12	SENSOR-1	SINKNODE-10	0.000
0	N/A	Control_Packet	AODV_RREP	SINKNODE-10	Broadcast-0	SINKNODE-10	SENSOR-9	N/A
0	N/A	Control_Packet	AODV_RREP	SINKNODE-10	Broadcast-0	SINKNODE-10	SENSOR-8	N/A

Figure 10: Packet Animation

As you can see in the animation in Figure 10 basically the Motion Sensor as the source ID is sending the data or information to the destination which is the Wired Node. However, the information goes to the 6_Lowpan, and after that, it goes to the router, and at the end, it reaches its destination which is the wired node. In Figure 10, you can see how successfully the information reaches its destination.

TABLES FROM SIMULATION (UDP & TCP)



Application_Metrics_Table

Application_Metrics

Application Id	Throughput Plot	Application Name	Packet generated	Packet received	Throughput (Mbps)	Delay(microsec)	Jitter(microsec)
1	Application_Throughput_plot	App1_SENSOR_APP	100	90	0.000360	22723.637778	14753.696629

Queue_Metrics_Table

Queue_Metrics

Device_id	Port_id	Queued_packet	Dequeued_packet	Dropped_
10	2	107	107	0
11	1	16	16	0

IP_Metrics_Table

IP_Metrics

Device Id	Packet sent	Packet forwarded	Packet received
1	201	0	0
2	102	0	0
3	102	0	0
4	102	0	0
5	102	0	0
6	102	0	0
7	102	0	0
8	102	0	0
9	102	0	0
10	210	90	16
11	106	90	17
12	0	0	90

Link_Metrics_Table

Link_Metrics

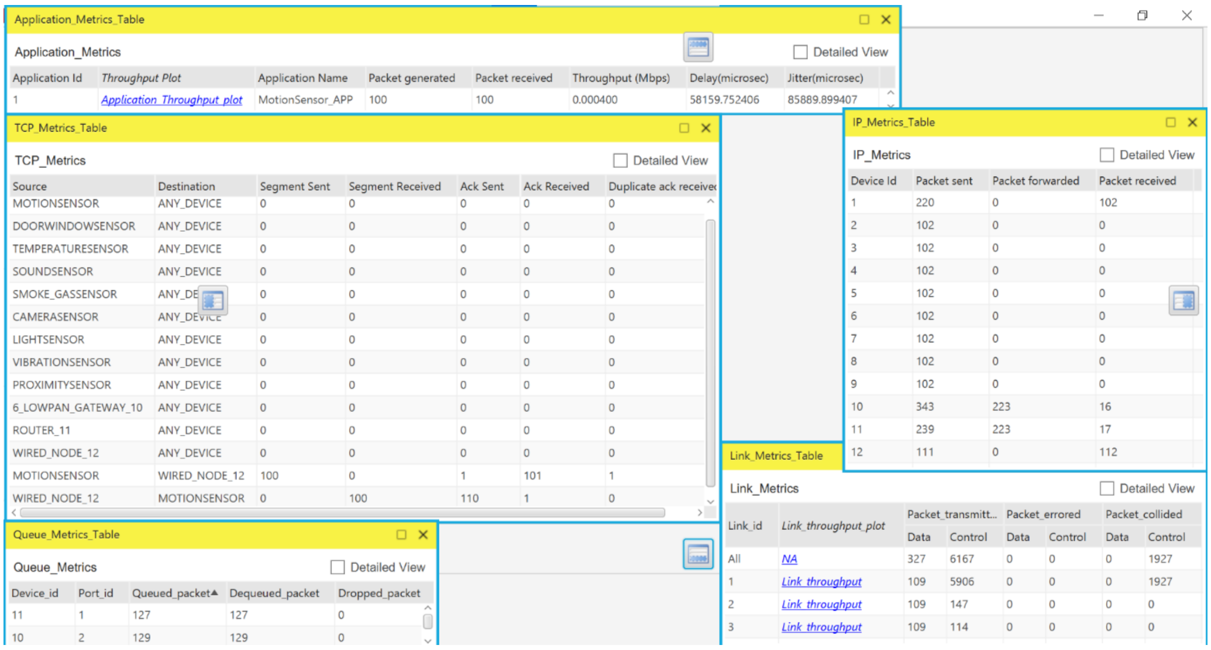
Link_id	Link_throughput_plot	Packet_transmitt...		Packet_errored		Packet_collided	
		Data	Control	Data	Control	Data	Control
All	NA	270	5910	0	0	0	1815
1	Link_throughput	90	5877	0	0	0	1815
2	Link_throughput	90	33	0	0	0	0
3	Link_throughput	90	0	0	0	0	0

UDP Metrics_Table

UDP Metrics

Device id	Local address	Foreign address	Datagram sent	Datagram received
1	FDEC:3017:E256:9BB8:1FE7:7510:1307:D434:82	11.3.1.2:36934	100	0
12	11.3.1.2:36934	FDEC:3017:E256:9BB8:1FE7:7510:1307:D434:82	0	90

Table 1: UDP simulation



Application_Metrics_Table

Application_Metrics

Application Id	Throughput Plot	Application Name	Packet generated	Packet received	Throughput (Mbps)	Delay(microsec)	Jitter(microsec)
1	Application_Throughput_plot	MotionSensor_APP	100	100	0.000400	58159.752406	85889.899407

TCP_Metrics_Table

TCP_Metrics

Source	Destination	Segment Sent	Segment Received	Ack Sent	Ack Received	Duplicate ack receive
MOTIONSENSOR	ANY_DEVICE	0	0	0	0	0
DOORWINDOWSENSOR	ANY_DEVICE	0	0	0	0	0
TEMPERATURESENSOR	ANY_DEVICE	0	0	0	0	0
SOUNDSENSOR	ANY_DEVICE	0	0	0	0	0
SMOKE_GASSENSOR	ANY_DE	0	0	0	0	0
CAMERASENSOR	ANY_DEVICE	0	0	0	0	0
LIGHTSENSOR	ANY_DEVICE	0	0	0	0	0
VIBRATIONSENSOR	ANY_DEVICE	0	0	0	0	0
PROXIMITYSENSOR	ANY_DEVICE	0	0	0	0	0
6_LOWPAN_GATEWAY_10	ANY_DEVICE	0	0	0	0	0
ROUTER_11	ANY_DEVICE	0	0	0	0	0
WIRED_NODE_12	ANY_DEVICE	0	0	0	0	0
MOTIONSENSOR	WIRED_NODE_12	100	0	1	101	1
WIRED_NODE_12	MOTIONSENSOR	0	100	110	1	0

IP_Metrics_Table

IP_Metrics

Device Id	Packet sent	Packet forwarded	Packet received
1	220	0	102
2	102	0	0
3	102	0	0
4	102	0	0
5	102	0	0
6	102	0	0
7	102	0	0
8	102	0	0
9	102	0	0
10	343	223	16
11	239	223	17
12	111	0	112

Link_Metrics_Table

Link_Metrics

Link_id	Link_throughput_plot	Packet_transmitt...		Packet_errored		Packet_collided	
		Data	Control	Data	Control	Data	Control
All	NA	327	6167	0	0	0	1927
1	Link_throughput	109	5906	0	0	0	1927
2	Link_throughput	109	147	0	0	0	0
3	Link_throughput	109	114	0	0	0	0

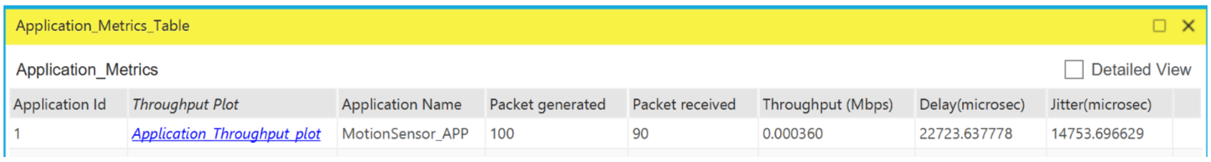
Queue_Metrics_Table

Queue_Metrics

Device_id	Port_id	Queued_packet▲	Dequeued_packet	Dropped_packet
11	1	127	127	0
10	2	129	129	0

Table 2: TCP Simulation

Let us start analyzing the values we got from the table 1 UDP simulation:



Application_Metrics_Table

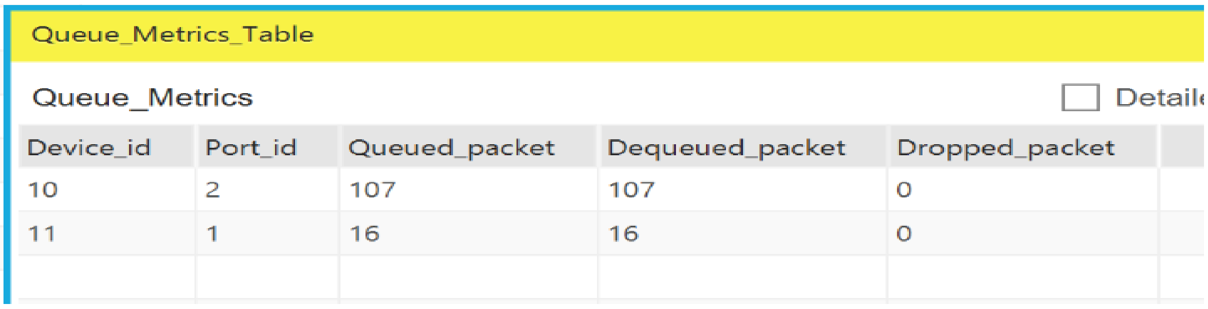
Application_Metrics

Detailed View

Application Id	Throughput Plot	Application Name	Packet generated	Packet received	Throughput (Mbps)	Delay(microsec)	Jitter(microsec)
1	Application_Throughput_plot	MotionSensor_APP	100	90	0.000360	22723.637778	14753.696629

Figure 11: Application Metrics Table UDP

In Figure 11, we can observe that the total number of packets generated for Motion Sensor is 100 packets in which only 90 packets have been successfully received through the application within the network called Motion Sensor App. We can also observe that the packets have been measured at 0.000360 Mbps, which is low throughput, and we can observe that there is a loss of packets within the network.



Queue_Metrics_Table

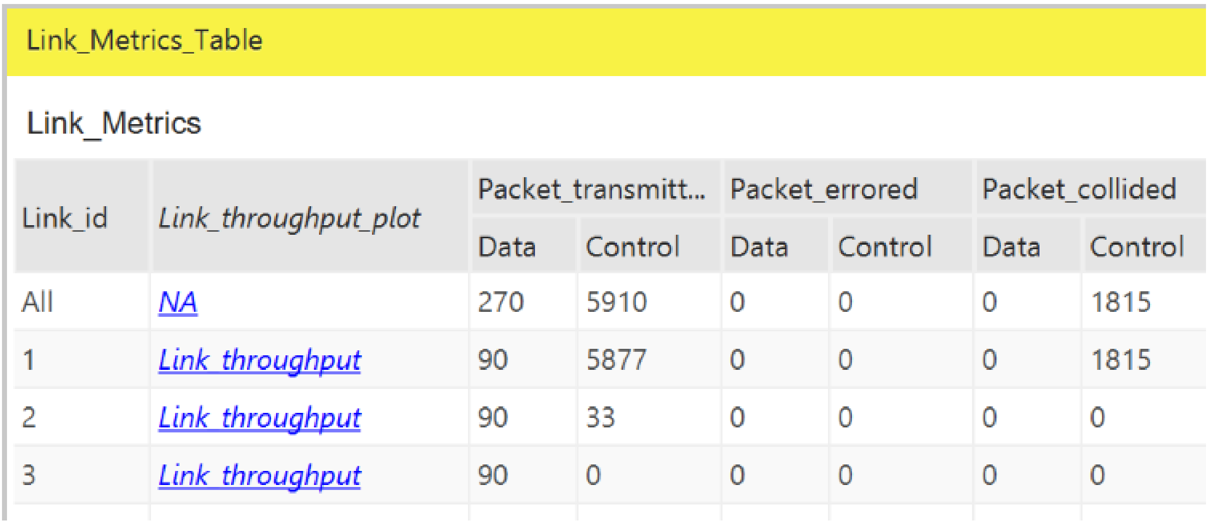
Queue_Metrics

Detaile

Device_id	Port_id	Queued_packet	Dequeued_packet	Dropped_packet
10	2	107	107	0
11	1	16	16	0

Figure 12: Queue Metric Table UDP

Device id 10 and 11 in my network were set as the 6_Lowpan Gateway and the router that got the port id 2(6_Lowpan) and 1(router) and we can see in the graph that there are zero dropped packets which means that no packets loss was found around these devices. Besides that, for the 6_Lowpan gateway, we can observe that 107 packets were queued, and for the router, only 16 were queued. Moreover, the number of packets that have been processed and removed or dequeued for the 6_Lowpan gateway is 107 packets while for the router is 16 packets which indicates that the network is handling the traffic in a very effective way.



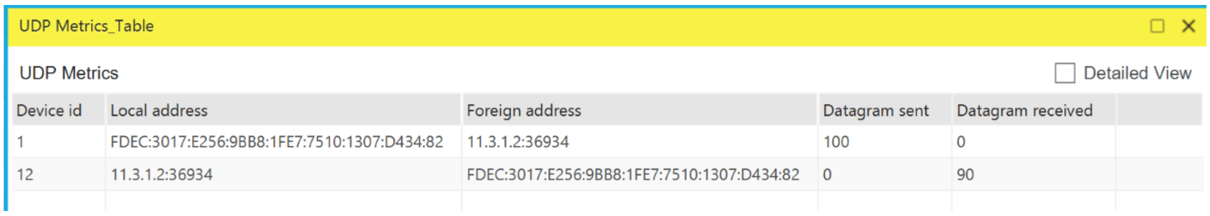
Link_Metrics_Table

Link_Metrics

Link_id	Link_throughput_plot	Packet_transmitt...		Packet_errored		Packet_collided	
		Data	Control	Data	Control	Data	Control
All	NA	270	5910	0	0	0	1815
1	Link_throughput	90	5877	0	0	0	1815
2	Link_throughput	90	33	0	0	0	0
3	Link_throughput	90	0	0	0	0	0

Figure 13: Link Metric Table UDP

So basically, in the table above Figure 13, we can observe that 1815 packet has been collided for all the links which it possible because there are around 9 devices send their information simultaneously on the same Ad-hoc or channel. Besides that, we can observe that links 2 and 3 had no collision which is a good sign and means that they were able to handle the data they have received without any disruption or interference. The main link is link 1 which comes from the motion sensor device that is handling the control traffic and it suggests that all packet collisions have happened in that link.



UDP Metrics_Table

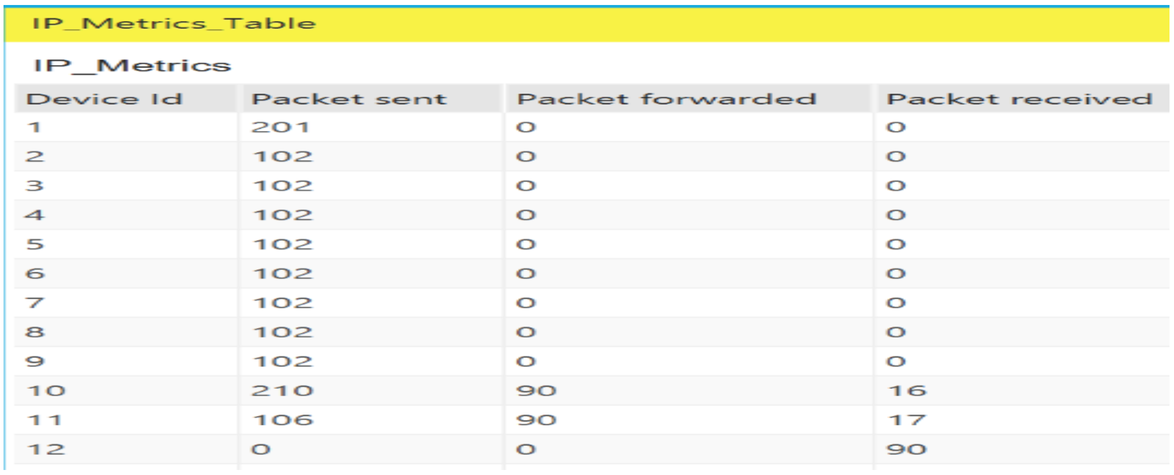
UDP Metrics

Detailed View

Device id	Local address	Foreign address	Datagram sent	Datagram received
1	FDEC:3017:E256:9BB8:1FE7:7510:1307:D434:82	11.3.1.2:36934	100	0
12	11.3.1.2:36934	FDEC:3017:E256:9BB8:1FE7:7510:1307:D434:82	0	90

Figure 14: UDP Metric Table UDP

In the UDP Metric Table, we can appreciate the IPV6 address that is assigned to device 1 or the sensor that has transmitted the data, which is the Motion Sensor, and the IPV6 assigned to the wired node or device 12. Lastly, in the table, we can clearly verify that 100 packets have been sent from the motion sensor while the wired node receives 90 packets.



IP_Metrics_Table

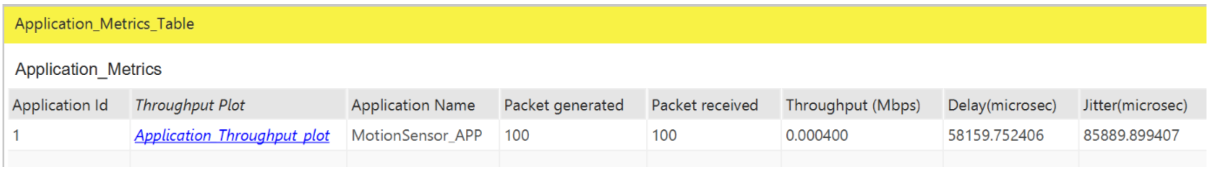
IP_Metrics

Device Id	Packet sent	Packet forwarded	Packet received
1	201	0	0
2	102	0	0
3	102	0	0
4	102	0	0
5	102	0	0
6	102	0	0
7	102	0	0
8	102	0	0
9	102	0	0
10	210	90	16
11	106	90	17
12	0	0	90

Figure 15: IP Metric Table UDP

In Figure 14, we have the information on the Internet Protocol packet. Since we are taking the motion sensor as a reference, we can analyze it by saying that device 1 or the Motion Sensor has sent 201 packets which go to the Ad-hoc component, then it goes to the 6_Lowpan gateway(first intermediary) which is sending 210 packets and finally going to the router(second intermediary) that is sending 106 packets. Secondly from 201 packets from the Motion sensors or device 1 that went across the 6_Lowpan gateway and the router, 90 packets were forwarded by them, and 16 packets were received for the 6_Lowpan gateway at one side whereas 17 packets were received by the router at another side. Finally, 90 packets were received by the wired nodes.

Now let us compare what we got from the UDP(User Datagram Protocol) table with the TCP(Transmission Control Protocol) table:


Application_Metrics_Table

Application_Metrics

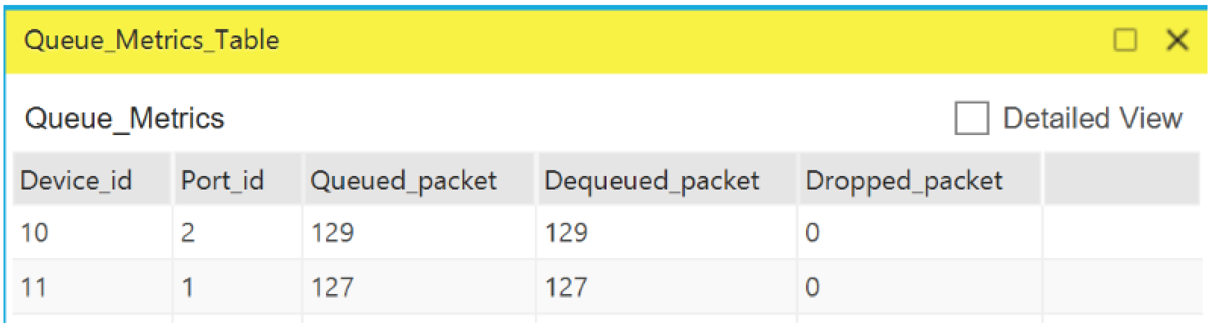
Application Id	Throughput Plot	Application Name	Packet generated	Packet received	Throughput (Mbps)	Delay(microsec)	Jitter(microsec)
1	Application_Throughput_plot	MotionSensor_APP	100	100	0.000400	58159.752406	85889.899407

Figure 15: Application Metric Table TCP

For Transmission Control Protocol table in Figure 15 we can observe how we got the 100 packets generated which we have recovered from the received packet which indicates that we did not lose any packet from the Motion Sensor to the wired nodes whereas in the UDP Application Metric Table in Figure 11 we had a loss of 10 packets.

We can also make a difference in both TCP and UDP which can be observed in the Throughput. The throughput of UDP is 0.000360 Mbps whereas the throughput for TCP is 0.000400 Mbps which is the highest one.

The delay for UDP is less than the TCP delay because UDP lacks an acknowledgment process like TCP.


Queue_Metrics_Table

Queue_Metrics

Detailed View

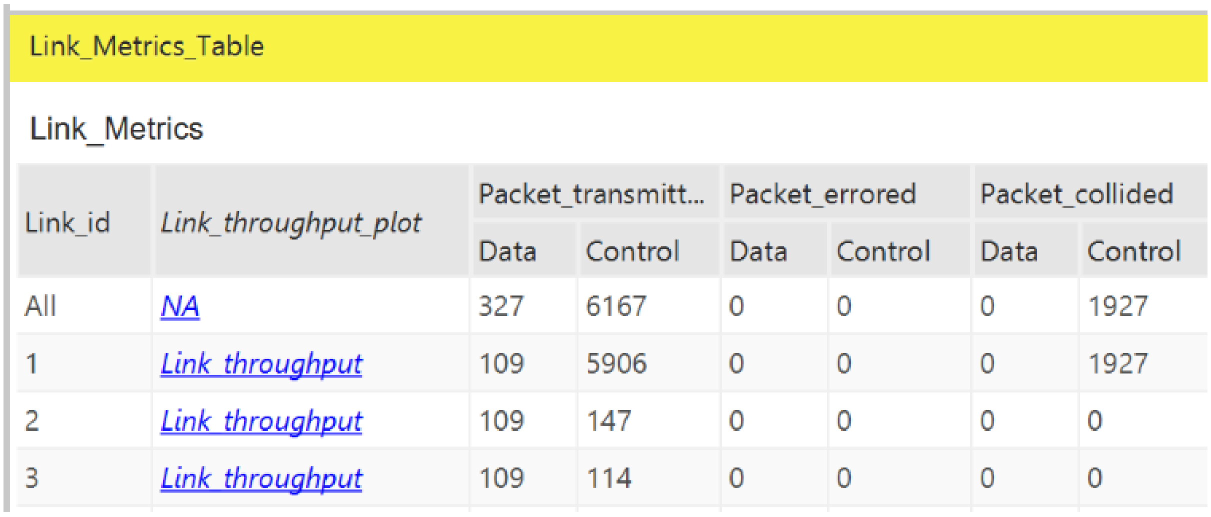
Device_id	Port_id	Queued_packet	Dequeued_packet	Dropped_packet
10	2	129	129	0
11	1	127	127	0

Figure 16: Queue Metric Table TCP

Now comparing the UDP Queue Metric Table in Figure 12 and the TCP Queue Metric Table in Figure 16 we can observe that the TCP table shows some significant numbers for device 10 (6_Lowpan) and device 11(router) which got a high number of queued and dequeued packets, and that is because of higher traffic occurring in TCP as well as the capacity of acknowledgments.

Both UDP and TCP got zero dropped packets which indicates that there is no congestion within the network.

For UDP and TCP the queued and dequeued packets are operating efficiently without any loss of packet because the queued packet is being recovered in dequeued packet



Link_Metrics_Table

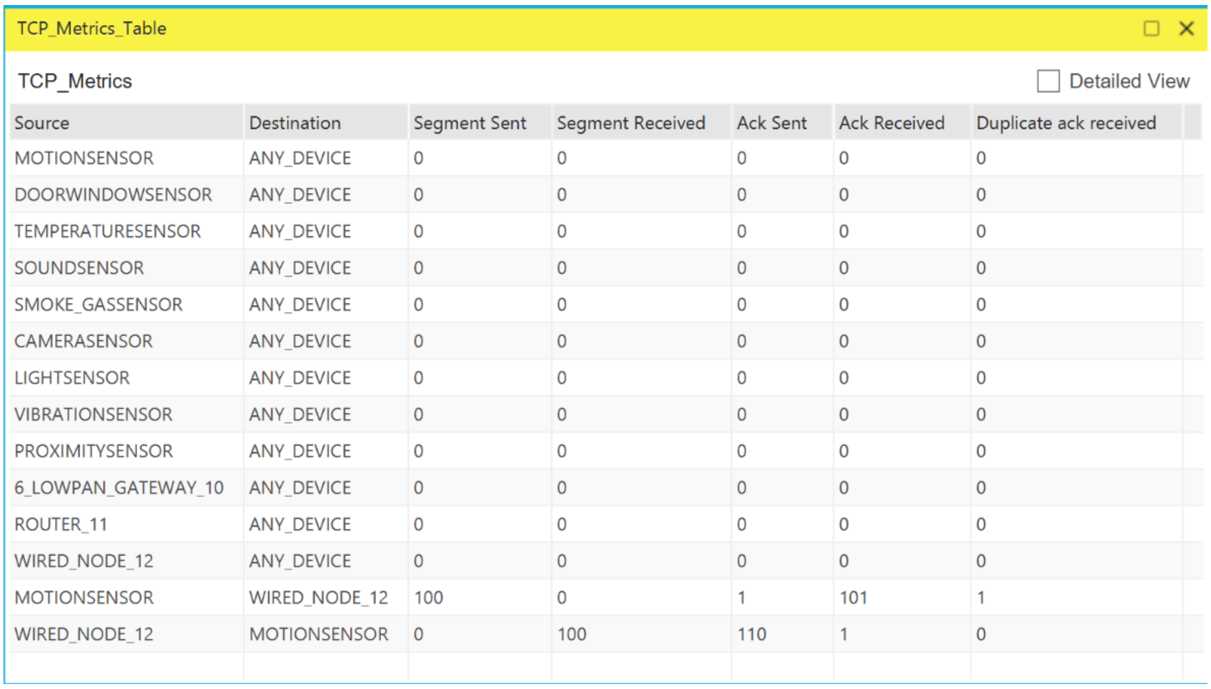
Link_Metrics

Link_id	Link_throughput_plot	Packet_transmitt...		Packet_errored		Packet_collided	
		Data	Control	Data	Control	Data	Control
All	NA	327	6167	0	0	0	1927
1	Link_throughput	109	5906	0	0	0	1927
2	Link_throughput	109	147	0	0	0	0
3	Link_throughput	109	114	0	0	0	0

Figure 17: Link Metric Table TCP

In TCP there is a total of 327 data packets and 6167 control packets that are transmitted across all the links. Basically, the link where we are sensing the information for motion sensor has 5906 control packet which is the highest one in the network with 1927 control packets colliding. On the other, in Figure 13 for the UDP Link Metric Table, we have less total number of data packet transmissions and control packets compared to the TCP. We also have a slightly smaller number of control packets from the motion sensor or link 1.

UDP indicates less overhead because it could get potentially faster because of its fewer or smaller control packets and collisions while TCP might lead to a good regulation of the control traffic because of its high control packets and collisions.



TCP_Metrics_Table

TCP_Metrics

Detailed View

Source	Destination	Segment Sent	Segment Received	Ack Sent	Ack Received	Duplicate ack received
MOTIONSENSOR	ANY_DEVICE	0	0	0	0	0
DOORWINDOWSENSOR	ANY_DEVICE	0	0	0	0	0
TEMPERATURESENSOR	ANY_DEVICE	0	0	0	0	0
SOUNDSENSOR	ANY_DEVICE	0	0	0	0	0
SMOKE_GASSENSOR	ANY_DEVICE	0	0	0	0	0
CAMERASENSOR	ANY_DEVICE	0	0	0	0	0
LIGHTSENSOR	ANY_DEVICE	0	0	0	0	0
VIBRATIONSENSOR	ANY_DEVICE	0	0	0	0	0
PROXIMITYSENSOR	ANY_DEVICE	0	0	0	0	0
6_LOWPAN_GATEWAY_10	ANY_DEVICE	0	0	0	0	0
ROUTER_11	ANY_DEVICE	0	0	0	0	0
WIRED_NODE_12	ANY_DEVICE	0	0	0	0	0
MOTIONSENSOR	WIRED_NODE_12	100	0	1	101	1
WIRED_NODE_12	MOTIONSENSOR	0	100	110	1	0

Figure 18: TCP Metric Table

In this Figure 19, you can notice that there is one-way communication or a one-to-one relationship happening from the source ID which is MotionSensor to the destination ID

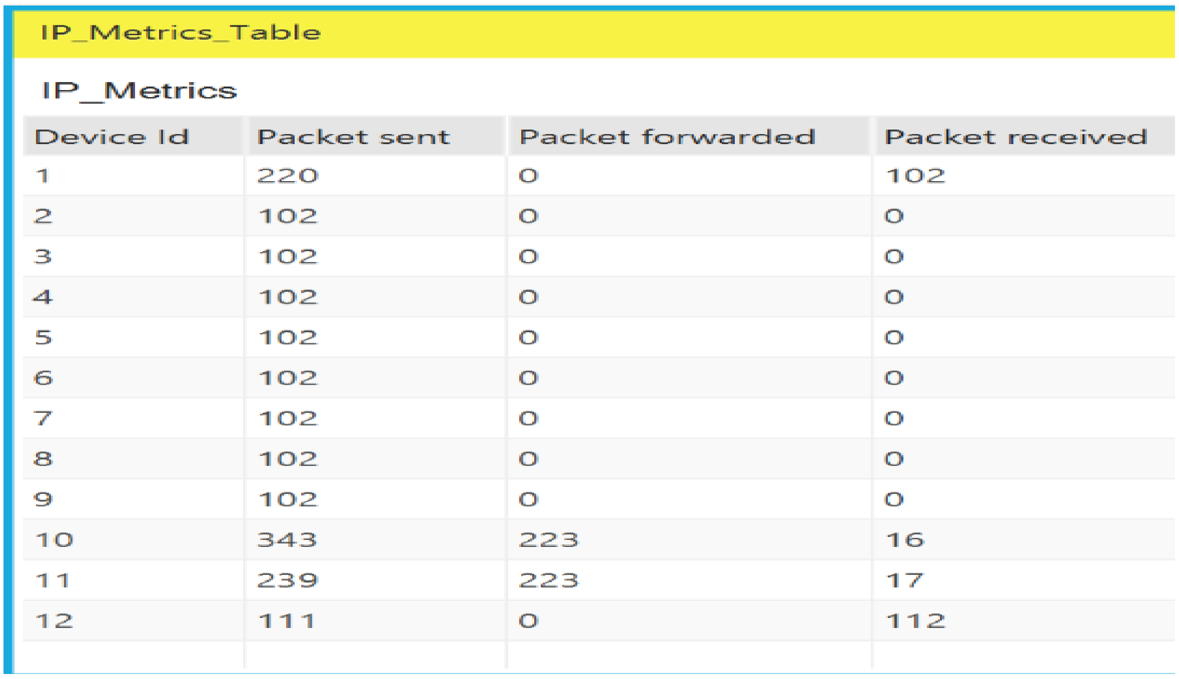
Wired Node with a successful acknowledgement that was sent back. You can also realize that the other sensors are zero which indicates that they are not communicating within the network as I did not activate them for data transmission.

MOTION SENSOR TO WIRED NODE

The motion sensor sent 100 packets or segments to the wired node but did not receive any data or segment, which is why it is zero in the table in Figure 18. This could suggest that we are dealing with a one-to-one relationship. After that, the motion sensor received one acknowledgment, ACK from the wired node which is like a confirmation that the 100 segments sent by the motion sensor were received. The wired node sent 101 acknowledgments back to the motion sensor. Finally, we can observe that there was a duplicate acknowledgment received by the motion sensor which means that one of the segments might have been lost and retransmitted again.

WIRED NODE TO MOTION SENSOR

The wired node sent 100 segments to the motion sensor. Then the motion sensor received 110 acknowledgments in response which is many acknowledgments received by the motion sensor and this indicates that some segments might have been retransmitted by the wire node. No duplicate acknowledgment was received by the wired node which means that there is no error sent by the motion sensor.


IP_Metrics_Table

IP_Metrics

Device Id	Packet sent	Packet forwarded	Packet received
1	220	0	102
2	102	0	0
3	102	0	0
4	102	0	0
5	102	0	0
6	102	0	0
7	102	0	0
8	102	0	0
9	102	0	0
10	343	223	16
11	239	223	17
12	111	0	112

Figure 19: IP Metrics Table TCP

In Figure 14, we have the information on the Internet Protocol packet for UDP where a packet from MotionSensor is sent and we end up receiving 112 packets from the wired node whereas in Figure 19 for TCP a packet of 220 has been sent and it first went to the 6_Lopan gateway which has transmitted 343 data packet and secondly the data packet sent 239 from the router and thirdly 111 data packet transmitted from the wired node. Besides that, the packet forwarded from the 6_Lowpan gateway and the router is 223 data of which we got 16 packets received from the 6_Lowpan gateway and 17 packets from the router. Lastly, we only got 112 packets received in the destination ID which is the wired node.

In TCP we got a higher number of packets sent from device 1 or the motion sensor compared to what we got in UDP.

TCP AND UDP GRAPHS:

TCP GRAPHS

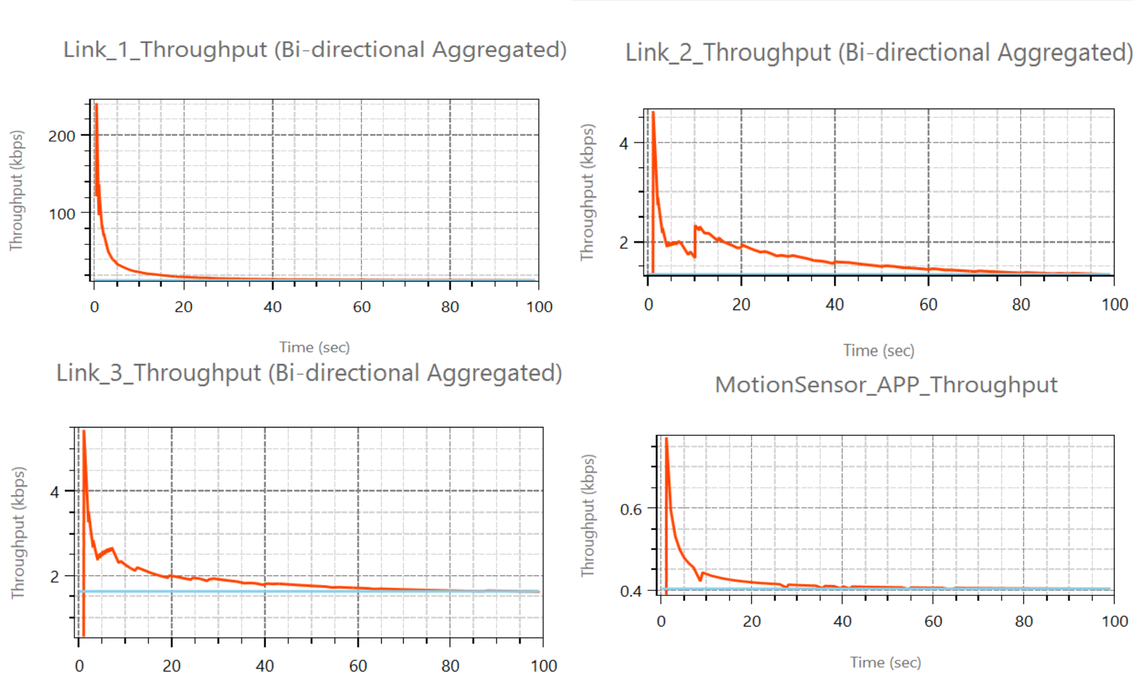


Figure 20: Link 1, Link 2, Link 3, MotionSensor graphs TCP

UDP GRAPHS

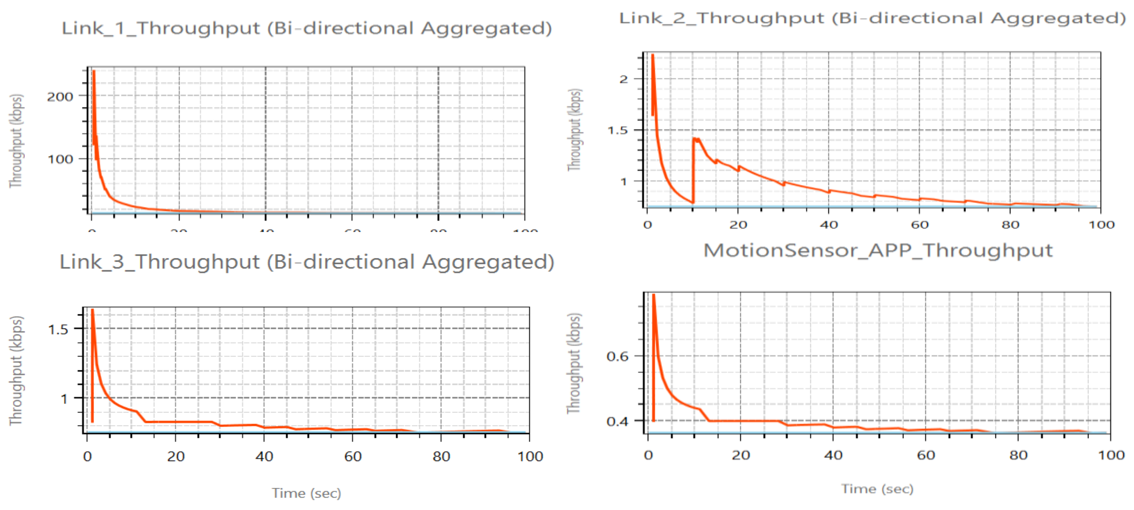


Figure 21: Figure 20: Link 1, Link 2, Link 3, Motion Sensor graphs UDP

TCP & UDP FOR LINK 1:

For TCP we have a maximum data throughput of 109 kbps and UDP shows 90 kbps (refer to Figures 17 and 13). In TCP graph we can observe a high initial value that is close to 200 kbps and then it gets stabilized to a steadier graph as it decreases. In other words, I can say that the starting of the phase in the graph represents the initial burst for TCP considering that the congestion is not noticeable which is why the throughput stabilizes. As per UDP, it follows the same pattern which is unusual considering that UDP does not handle certain mechanisms like TCP. An example of any mechanism could be congestion control or acknowledgment.

TCP & UDP FOR LINK 2:

For TCP, we have a throughput that goes up to 4 kbps which is low because the link is not taking a high considerable value for traffic. The range of the horizontal is from 0 to 100. We can observe an initial peak that drops immediately, and the peak is lower than 4 kbps. We can observe a sharp above 2kbps that occurs within the first 10 seconds and gets more stabilized after 20 seconds. Now for the UDP graph, we can say that first, the throughput or Y axis is above 2 kbps which indicates that the initial peak is above 2 kbps, and that as soon as the measurement started, we can see that the link began transmitting its maximum data immediately. Right after the peak, we have a decline that is heavily reaching 1.5 kbps.

TCP & UDP FOR LINK 3:

For TCP in link 3 there is an immediate spike at the beginning of the which is usual in TCP connections. Besides that, we can observe a steep decline that stabilized over time. The throughput is consistent in the graph for TCP. In the end, the throughput might remain constant after the initial sharp occurred around 10 seconds. For UDP, we have a time from 0 to 100 seconds, and the values of the throughput start from 0 and go up to 1.5 kbps. At the beginning of the graph, we have a sharp peak that goes up to 1.5 kbps which represents the start of a data transfer. Besides that, there is a quick decline immediately after the peak as the throughput decreases until it settles around 1 kbps. Finally, we can observe that right after the initial decline the throughput gets more stable.

TCP & UDP FOR MOTIONSENSOR_APP THROUGHPUT:

For TCP the initial peak is going up to 0.6 kbps. We can observe a sharp decline where we got a steep decline which gets stabilized over time. The throughput goes down at a low rate of 0.4 kbps. Now regarding UDP, we can visualize that the initial peak is the same as TCP which is around 0.6 kbps but there is a faster drop approximately around 0.4 kbps as well which is getting more stable over time. Both UDP and TCP have a similar initial peak and decline. They both stabilize at 0.4 kbps as well.

DISCUSSION

I have created an IoT application for Smart Campus application. The application contains 9 wireless sensors in which I selected only one wireless sensor to be transmitted to the wired node. The sensor is called a motion sensor which goes to the Ad-hoc link, after that it will move towards the 6_Lowpan Gateway and the router, and then finally it will reach the wire node component. Moreover, I have set the wireless sensor (motion sensor) as the source ID and the wired node as the destination ID. After that, I implemented a notable comparison of the network based on TCP and UDP where I got some significant insights and information.

Starting with TCP, which is Transmission Control Protocol, I have analyzed it and concluded that there is no packet loss because the packets that were generated within the application were recovered as well, so there is no loss packet. Besides that, I have noticed that there is high traffic in wireless devices such as 6_Lowpan and the router which is usual for TCP because it has more functionalities such as congestion control protocol, acknowledgments received or sent, etc. That is why also we have higher values in TCP compared to UDP.

For UDP I have a considerable number of packet losses which is only 10 segments from the source to the Destination. Besides that, since we only receive 90 segments in the destination from the motion sensor to the wired node, we must consider the data packet transmission, which is the same as the received packet, we also have the control packet transmission which is 5877 and the control packet collided which is 1815. All these parameters that we just mentioned now are considerably slower than TCP as they should be because in UDP we don’t have certain functionalities as in TCP.

The graphs for UDP and TCP are slightly similar, but we can still encounter or appreciate some differences such as the value of the throughput and the time for example the link 1 graph for both TCP and UDP follows the same pattern which is unusual. The difference is more noticeable in the peak, and you can observe that the peak for TCP is usually higher than the peak for UDP. In the end, we can say that the starting throughput for TCP is much more consistent whereas the one for UDP seems to follow the same pattern, which indicates that UDP should be influenced in the network to behave like TCP.

CONCLUSION

I have completed successfully the IoT application for a Smart Campus that consists of 9 wireless sensors devices connected to a 6_lowpan, router until the wired node. I have selected a sensor called motion sensor which will sense or detect any movement of people within the building or the campus. The IoT application created in this project consists of 9 devices connected to a Ad-hoc link as well as the 6_Lowpan, router and a wired node.

REFERENCES

Lathi, B. P., & Ding, Z. (2010). *Modern digital and analog communication systems* (International 4th ed.). Oxford University Press.

Gallager, R. G. (2007). *Principles of digital communication*. Cambridge University Press.

Agbo, S. O., & Sadiku, M. N. O. (2017). *Principles of modern communication systems*. Cambridge University Press. https://doi.org/10.1017/9781107107922