Fog Computing in IOT

### By- Abhishek Tayal (IIT(ISM) Dhanbad)

#### Software Used:

NetSim Standard v10.2, Visual Studio 2017, Wireshark.

#### Introduction:

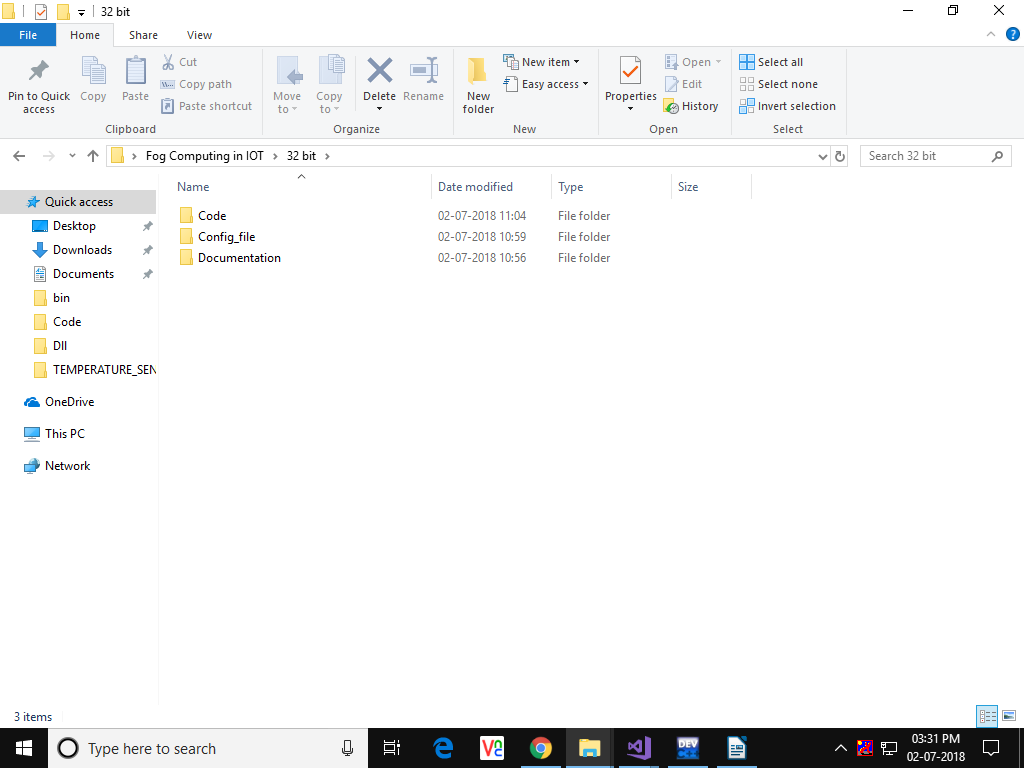
Fog computing refers to extending cloud computing to the edge of an enterprise's network. In terms of simulation fog computing essentially translated to have compute capability at the edge (or) at the intermediate devices. By handling these services that make up the Internet of Things (IoT) at the network edge, data can in many cases be processed more efficiently than if it needed to be sent to the cloud for processing.

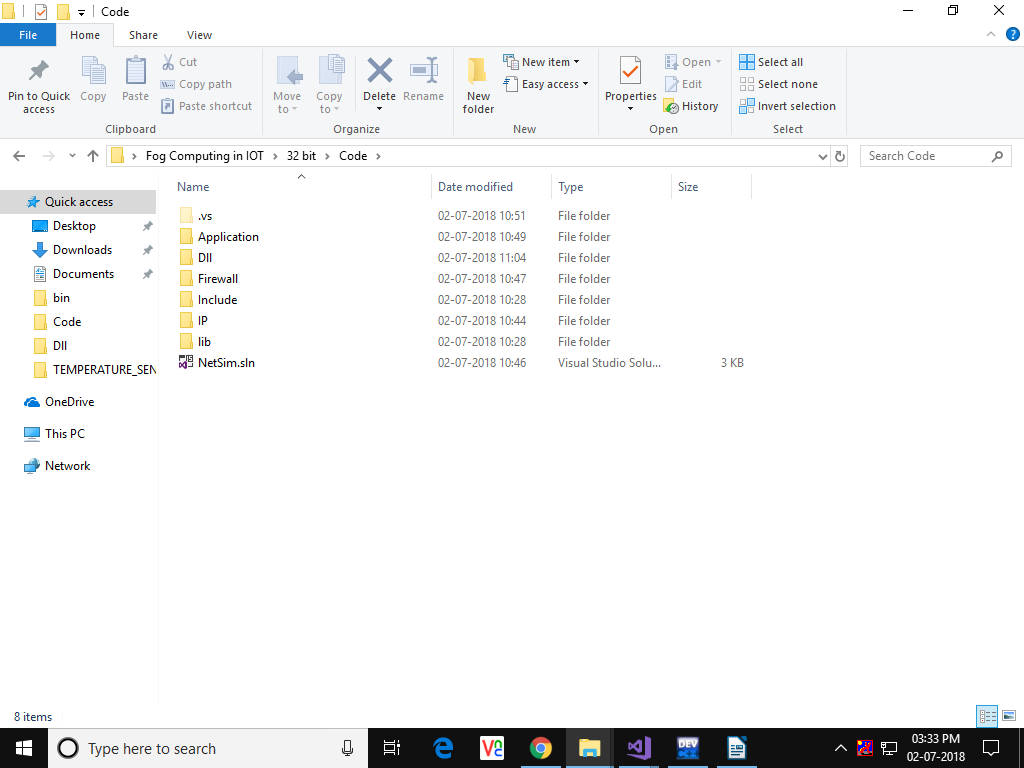
#### Project Goal:

To implement Fog Computing in NetSim IOT module. The Basic idea here is that sensors sends reading of the temperature (may be something else example pressure) through the LoWPAN Gateway, the Gateway will forward it if that value (reading by Sensor) is greater than a specific threshold value otherwise it will drop it. And when destination Wired Node (Server) receives a packet from the sensor then it will send a response.

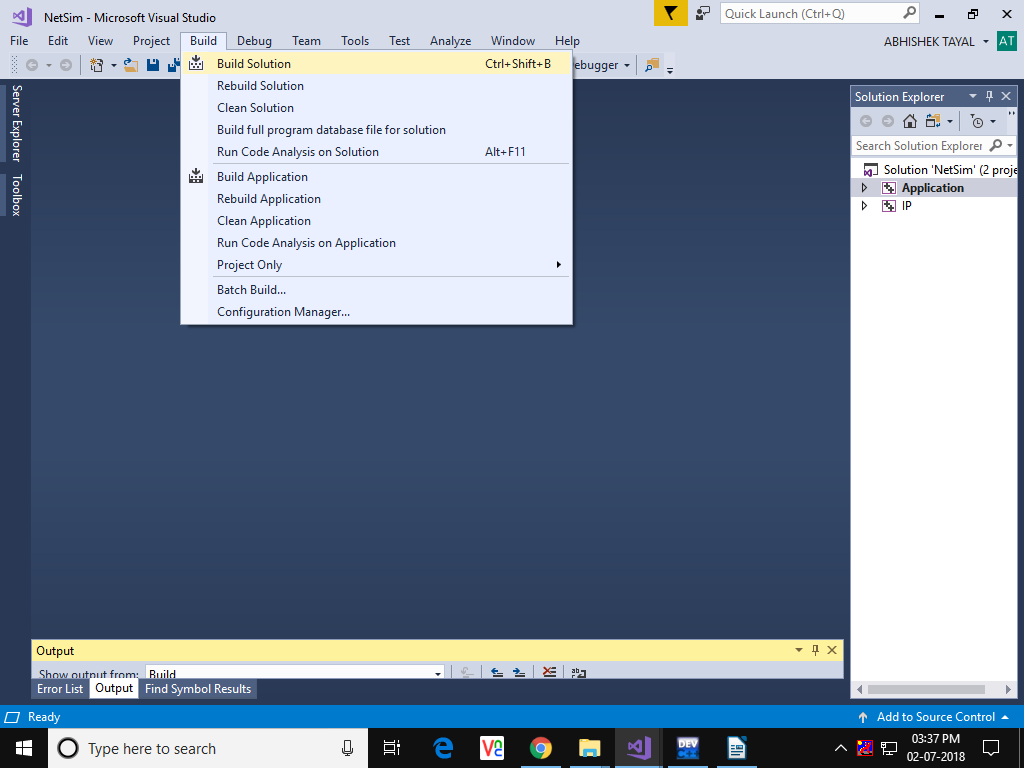
#### Steps to build and run the project:

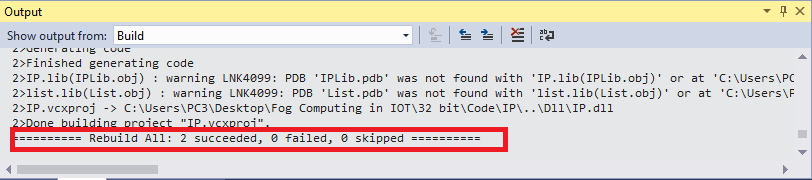
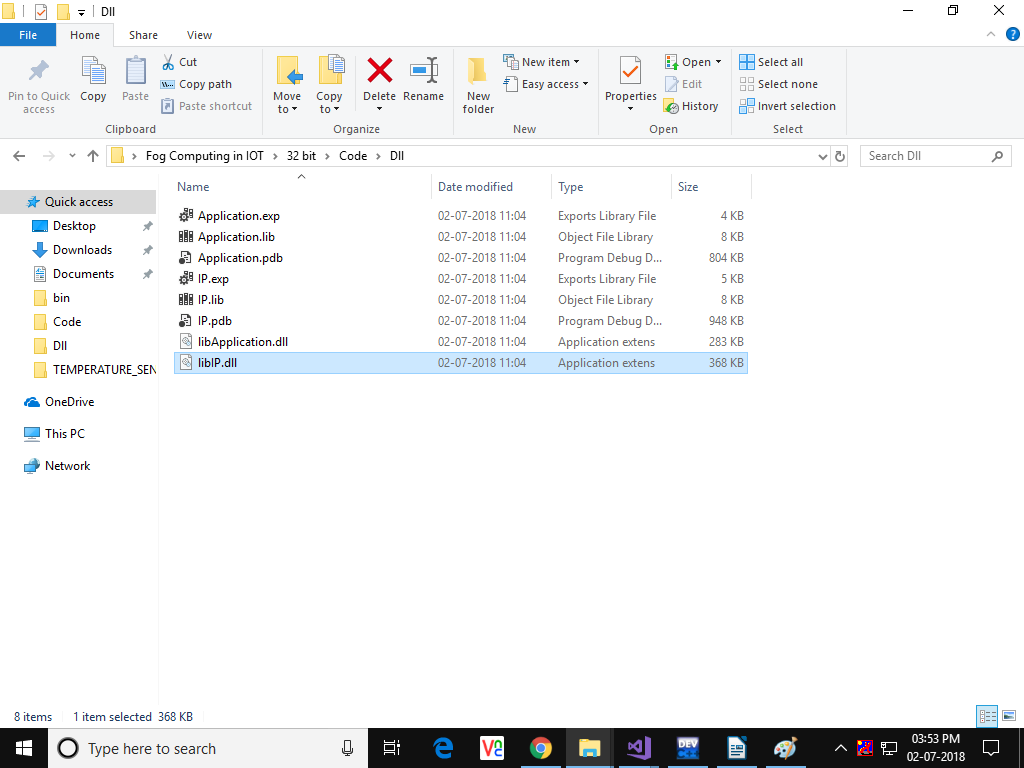
1. Open the NetSim.sln using Visual Studio from Code folder that is present inside the extracted Fog Computing in IOT folder.

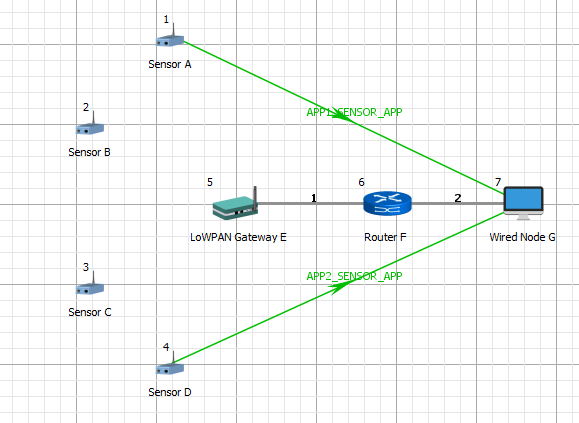




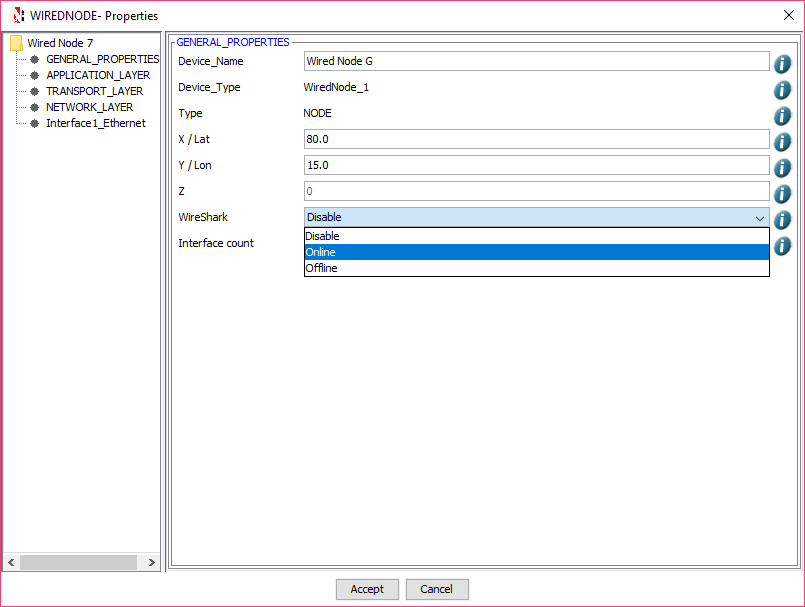
1. In Visual Studio go to Build and Rebuild Solution



1. You should see a message in the Output window as shown in the following figure:
2. Inside the extracted Fog Computing in IOT folder, go to DLL folder and copy the file libApplication.dll and libIP.dll 
3. Go to **<NetSim\_Install\_Directory>/bin** (It will be something like “C:\Program Files\NetSim Standard\bin”)
4. Rename libApplication.dll and libIP.dll files which are already present there to something like libApplication\_original.dll and libIP\_original.dll as a backup.
5. Paste the new libApplication.dll and libIP.dll files that you copied from the DLL folder.
6. A sample Configuration.netsim file is provided inside the Config\_file folder. Open the sample Configuration file or create a network scenario in NetSim IoT module as shown below and configure traffic between sensors and Wired Node (Server) :



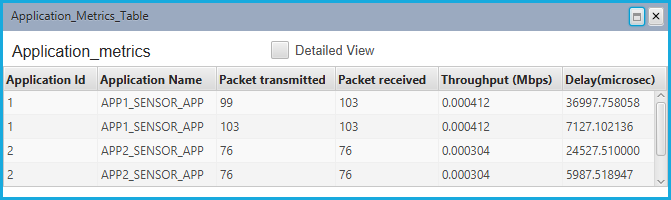
1. Packets can be captured using Wireshark to analyse the payload. Set the Wireshark to online in the Destination wired node where you can observe the packet payload.



1. Run simulation for say, 100 seconds with packet trace enabled and Wireshark enabled in the destination nodes.

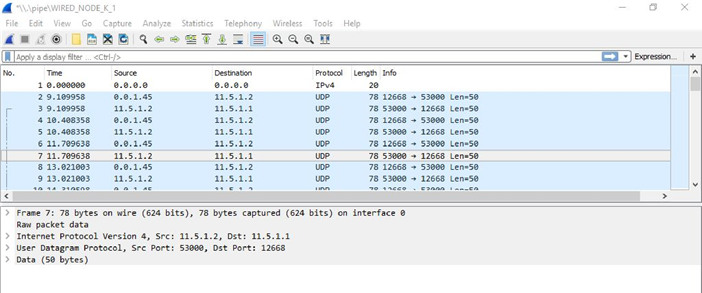
#### Results and Analysis:

1. At simulation end, observe the Application metrics table present in the Simulation Results window as shown below:

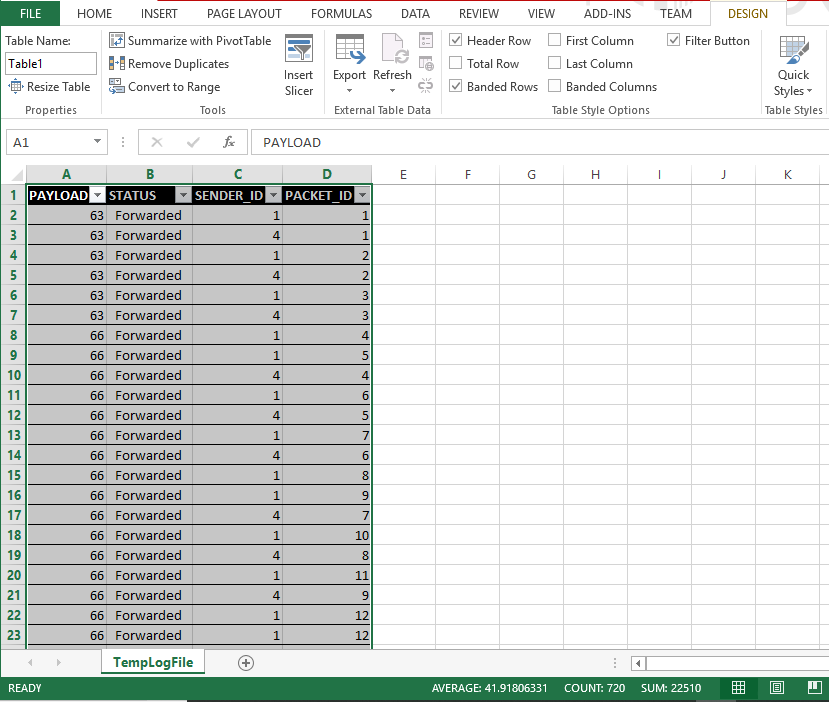


As you can see in this example, for Application 1 for 28 packet were received at the destination out of total 99 packets that were sent. This is because packets are filtered in the Gateway device based on the threshold that was set. For each packet that was successfully received, there is a response packet that is generated from the destination. You can see 28 response packet that were sent from destination to source for Application 1.

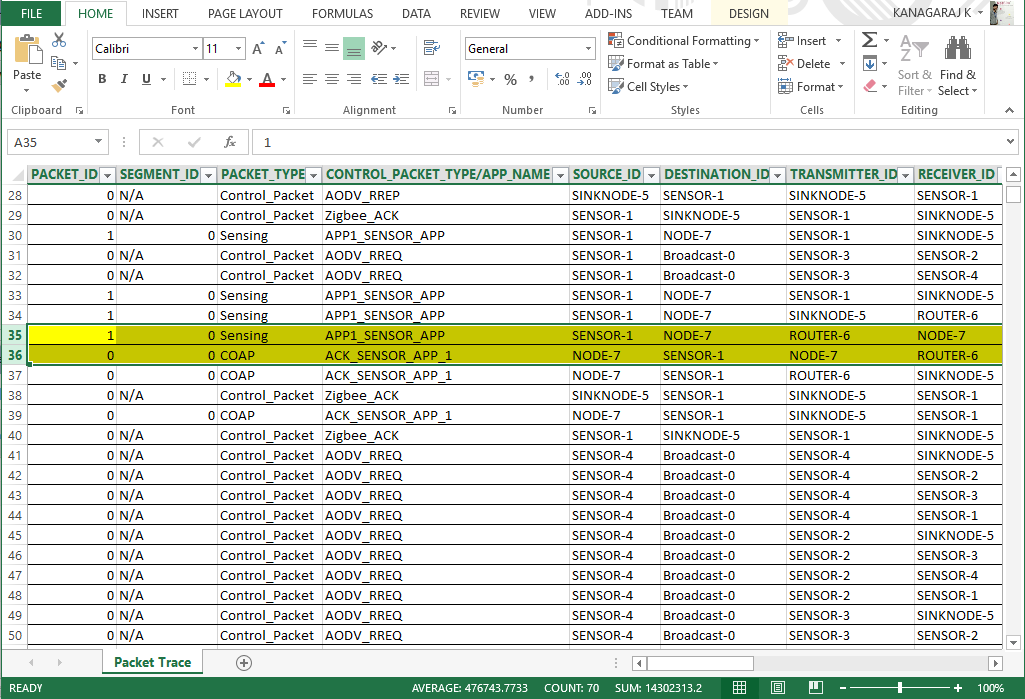
1. Wireshark if enabled in the destination wired or wireless nodes will give you a pcap file as shown below, where you can observe the sensor data and the response packet that was sent back:



1. The sensing packets sent by the sensors and the responses message sent by the destination nodes can clearly be analysed using NetSim's inbuilt packet trace log file which can be obtained by enabling packet trace before running the simulation. Shown below is a packet trace file where one can observe the packet flow in the network:



1. The sensing packets sent by the sensors and the responses message sent by the destination nodes can clearly be analysed using NetSim's inbuilt packet trace log file which can be obtained by enabling packet trace before running the simulation. Shown below is a packet trace file where one can observe the packet flow in the network:



#### Understanding Code modifications done:

**Following section explains the code modifications that were made for implementation of this project in NetSim:**

**Modifications are done to the various files of IP and Application source codes of Netsim.**

**1. Change the Payload value on the basis of Random number:**

The payload of the packet is added in the application layer while generating the packet. Assuming that the temperature varies from 0 to 99. So, we use a random value generating function giving the value between 0 and 99. We have to copy that value to our payload.

To change the payload the following steps can be followed:

**Step 1**: Open**NetSim.sln** file from**<NetSim Installation Directory>\src\Simulation**folder**.**Go to **copy\_payload()** function present in **Application.c** file inside Application project.

**Step 2:**Insidethe function **copy\_payload(),**make the modifications as given below (highlighted inred) :

**void copy\_payload(UINT8 real[], NetSim\_PACKET\* packet, unsigned int\* payload, APP\_INFO\* info)**  
**{**  
**srand(time(0));**  
**int r = rand() % 100;**  
**char s[2];**  
**s[1] = (char)((r % 10) + '0');**  
**s[0] = (char)((r / 10) % 10 + '0');**  
**int slength = strlen(s);**  
**u\_short i;**  
**uint32\_t key = 16;**  
**if (payload)**  
  
**{**  
**/\*for (i = 0; i < \*payload; i++)**  
**{**  
**if (info->encryption == Encryption\_XOR)**  
**real[i] = xor\_encrypt('a' + i % 26, 16);**  
**else**  
**real[i] = 'a' + i % 26;**  
**}\*/**  
**for (i = 0; i < \*payload; i++)**  
**{**  
**if (info->encryption == Encryption\_XOR)**  
**real[i] = xor\_encrypt(s[i % slength], 16);**  
**else**  
**real[i] = s[i%slength];**  
**}**  
**if (info->encryption == Encryption\_TEA)**  
**encryptBlock(real, payload, &key);**  
**else if (info->encryption == Encryption\_AES)**  
**aes256(real, payload);**  
**else if (info->encryption == Encryption\_DES)**  
**des(real, payload);**  
**}**  
**}**

**Step 3:**Nowmake modification highlighted inred in**fn\_NetSim\_Add\_DummyPayload()**in **Application.c file**

**while (packet)  
{  
if (!packet->szPayload &&  
packet->pstruAppData &&  
packet->pstruAppData->dPacketSize &&  
packet->pstruAppData->nAppType != TRAFFIC\_EMULATION /\* Don't set payload in  
emulation \*/ /\*&&  
wireshark\_flag\*/)  
{  
unsigned int size = (unsigned int)packet->pstruAppData->dPacketSize;**

**2. Find the LoWPAN Gateway id**

To ensure that the computing happens only in the Gateway, we use the device type and interface count parameters to differentiate Gateway node from the sensor nodes.

To find the Gateway Id the following steps can be followed:

**Step 1:**Go to **fn\_NetSim\_IP\_Run()** function present in**IP.c**file inside IPproject.

**Step 2:**Insidethe function **fn\_NetSim\_IP\_Run(),**make the modificationsas given below (highlighted in red) :

**declspec(dllexport) int fn\_NetSim\_IP\_Run()**

**{**

**int i = 0; int GatewayDeviceId = -1;**

**for (i = 0; i < NETWORK->nDeviceCount; i++)**

**{**

**if (NETWORK->ppstruDeviceList[i]->nDeviceType == 1 && NETWORK->ppstruDeviceList[i]->nNumOfInterface == 2)**

**GatewayDeviceId = NETWORK->ppstruDeviceList[i]->nDeviceId;**

**}**

**switch (pstruEventDetails->nEventType)**

**{**

**case NETWORK\_OUT\_EVENT:**

**3. Add condition to forward or drop the packet**

Now whether the packet is forwarded or dropped by the LoWPAN Gateway depends on the payload value (Temperature reading) and threshold value. In this case the threshold value is set to 50. So packet will be forwarded by the Gateway if payload value is greater than equal to 50, else will get dropped. These Modification are added in the IP module. It is also ensured that the payload is verified only for data packets and not for control packets. In NetSim all control packets have the id 0. So if packet Id is zero then there is no check done.

To add the condition following steps can be followed:

**Step 1:**Open**fn\_NetSim\_IP\_Run()** function present in**IP.c**file inside IPproject.

**Step 2:**Insidethe function **fn\_NetSim\_IP\_Run(),**make the modificationsas given below (highlighted in red) **\_declspec(dllexport) int fn\_NetSim\_IP\_Run()**

**{**

**switch (pstruEventDetails->nEventType)**  
**{**  
**case NETWORK\_OUT\_EVENT:**  
**{**  
**ptrIP\_FORWARD\_ROUTE route = NULL;**  
**NetSim\_PACKET\* packet = pstruEventDetails->pPacket;**  
**//Calculating the temperature value using the payload**  
**unsigned char str[3] = "";**  
**if (pstruEventDetails->pPacket->nPacketId != 0)**  
**{**  
**str[0] = pstruEventDetails->pPacket->szPayload->packet[0];**  
**str[1] = pstruEventDetails->pPacket->szPayload->packet[1];**  
**}**  
**char thresholdvalue = '5';**  
**if (pstruEventDetails->nDeviceId == GatewayDeviceId && (pstruEventDetails->pPacket->nPacketId != 0) && pstruEventDetails->pPacket->szPayload->packet[0] < '5')**  
**{**  
**fprintf(stderr, "\n Packet dropped. Payload is %s\n", str); packet->pstruNetworkData->nTTL = 0;**  
**}**  
**NETWORK\_LAYER\_PROTOCOL nLocalNetworkProtcol; nLocalNetworkProtcol = fnGetLocalNetworkProtocol(pstruEventDetails);**

**4. Store the details of each packet in a file**

In order to obtain a log file containing details such as **PayLoad value, Sender Id, Packet Status (Forwarded or Dropped) and packet Id,**the following steps are to be followed:

**Step 1:**Open **fn\_NetSim\_IP\_Init()**function present in **IP.c**file inside IP project.

**Step 2:**Insidethe function**fn\_NetSim\_IP\_Init()**make the modifications as given below (highlighted in red) :

**{**

**FILE\* fp;**

**fp = fopen("TempLogFile.csv", "w+");**

**if (fp)**

**{**

**fprintf(fp, "PAYLOAD,STATUS,SENDER\_ID,PACKET\_ID,"); fclose(fp);**

**}**

**NETSIM\_ID loop;**

**if (nVersion\_Type / 10 != VERSION)**

**{**

**printf("IP---Version number mismatch\n Dll Version=%d\n NetSim Version=%d\n FileName=%s\n Line=%d\n", VERSION, nVersion\_Type / 10, \_\_FILE\_\_, \_\_LINE\_\_);**

**exit(0);**

**}**

**Step 3:**Open**fn\_NetSim\_IP\_Run()** function present in**IP.c**file inside IPproject.

**Step 4:**Insidethe function **fn\_NetSim\_IP\_Run(),**make the modificationsas given below (highlighted in red) :

**\_declspec(dllexport) int fn\_NetSim\_IP\_Run()**

**{**

**int i = 0; int GatewayDeviceId = -1;**

**for (i = 0; i < NETWORK->nDeviceCount; i++)**

**if (NETWORK->ppstruDeviceList[i]->nDeviceType == 1 &amp;&amp; NETWORK->ppstruDeviceList[i]->nNumOfInterface == 2)**

**GatewayDeviceId = NETWORK->ppstruDeviceList[i]->nDeviceId;**

**switch (pstruEventDetails->nEventType)**

**{**

**case NETWORK\_OUT\_EVENT:**

**{**

**ptrIP\_FORWARD\_ROUTE route = NULL;**

**NetSim\_PACKET\* packet = pstruEventDetails->pPacket;**

**FILE\* fp;**

**unsigned char str[3] = "";**

**if (pstruEventDetails->pPacket->nPacketId != 0)**

**{**

**fp = fopen("TempLogFile.csv", "a+");**

**str[1] = pstruEventDetails->pPacket->szPayload->packet[1];**

**}**

**char thresholdvalue = '5';**

**if (pstruEventDetails->nDeviceId == GatewayDeviceId &amp;&amp; (pstruEventDetails->pPacket->nPacketId != 0) &amp;&amp; pstruEventDetails->pPacket->szPayload->packet[0] <'5' )**

**{**

**if (fp)**

**{**

**fprintf(fp, "\n%s,%s,%u,%lld,", str," Dropped ",pstruEventDetails->pPacket->nSourceId, pstruEventDetails->nPacketId);**

**fclose(fp);**

**}**

**fprintf(stderr,"\nPacket dropped. Payload is %s\n",str);**

**packet->pstruNetworkData->nTTL = 0;**

**}**

**else if (pstruEventDetails->nDeviceId == GatewayDeviceId &amp;&amp; pstruEventDetails->pPacket->nPacketId != 0)**

**{**

**if (fp)**

**{**

**fprintf(fp, "\n%s,%s,%u,%lld,", str, " Forwarded ", pstruEventDetails->pPacket->nSourceId, pstruEventDetails->nPacketId);**

**fclose(fp);**

**}**

**}**

**NETWORK\_LAYER\_PROTOCOL nLocalNetworkProtcol;**

**nLocalNetworkProtcol = fnGetLocalNetworkProtocol(pstruEventDetails);**

**if (nLocalNetworkProtcol)**

**{**

**fnCallProtocol(nLocalNetworkProtcol);**

**return 0;**

**}**

#### Related work:

The concept is further be extended by configuring the server nodes to send a response packet upon reception of sensor data from any sensor node. A simple example is discussed below where we attempt to generate a control packet which is sent to the source sensor device.

**Sending response packet from Wired Node (Server) upon successful reception of Sensor data:**

1. Open **fn\_NetSim\_Application\_Run()** function present in **Application.c** file inside Application project.  
2. In the **APPLICATION\_IN\_EVENT**make the modifications as given below (highlighted in red):  
**case APPLICATION\_IN\_EVENT:  
{  
.**

**.**

**.**

**.**

**.**

**if (pstruappinfo->nAppType == TRAFFIC\_BSM\_APP)  
{  
process\_saej2735\_packet(pstruPacket);  
}  
//Saving the informations before deleting a packet  
int sourceId = pstruappinfo->destList[0];  
NETSIM\_ID\* nDestination = pstruappinfo->sourceList;  
int destId = pstruappinfo->sourceList[0];  
int destPort = pstruappinfo->sourcePort;  
int sourcePort = pstruappinfo->destPort;  
double time = pstruEventDetails->dEventTime;  
int nApplicationId = pstruappinfo->id;  
int packetSize = pstruEventDetails->dPacketSize;  
//Delete the packet  
fn\_NetSim\_Packet\_FreePacket(pstruPacket);  
//Checking the conditions for response (Receiver is not a sensor, Sender is a sensor, App Type is Traffic\_Sensing ,**

**//Device is the destination of the packet & It is not a control packet  
if ((NETWORK->ppstruDeviceList[pstruEventDetails->nDeviceId -1]->ppstruInterfaceList[0]->nInterfaceType != INTERFACE\_LAN\_802\_15\_4) && \  
(NETWORK->ppstruDeviceList[pstruappinfo->sourceList[0]]->ppstruInterfaceList[0]->nInterfaceType == INTERFACE\_LAN\_802\_15\_4) && \  
pstruappinfo->nAppType == TRAFFIC\_SENSING && \  
pstruEventDetails->nDeviceId == pstruappinfo->destList[0] && \  
pstruEventDetails->nPacketId != 0)  
{  
fprintf(stderr, "\nPacket reached at Node\  
//Generating a packet from destination (Node) to the source (Sensor)  
pstruEventDetails->pPacket = fn\_NetSim\_Application\_GeneratePacket(pstruappinfo,  
pstruEventDetails->dEventTime,  
sourceId,  
1,  
nDestination,  
0,  
16,  
pstruappinfo->qos,  
sourcePort,  
destPort);  
//Change the name of Packet (Response packet which we will send back)  
char packetname[25] = "ACK\_SENSOR\_APP\_";  
char appId[20];  
itoa(nApplicationId, appId, 10);  
strcat(packetname, appId);  
strcpy(pstruEventDetails->pPacket->szPacketType, packetname);  
fnpAddEvent(pstruEventDetails);  
}  
// Here which type is placed is only getting processed next one is not getting processed  
else if (pstruPacket->nControlDataType == packet\_COAP\_REQUEST)  
{  
APP\_INFO\* pstruappinfo;**