IOT Project No.1 Report

1. Bahram Hedayati Person Code: 10870276
2. Mahsa Delaram Person Code: 10847175

1. Introduction

In this report, we are going to explain breifly how we did the tasks that were specified in the project number 1. The project is done in TinyOS and simulated in the Cooja framework. Basically, we developed two kinds of nodes:

- **PanC:** It is the Pan Coordinator which is in the center of topology in Cooja. It is addressed 9. The related files are PanC.nc and PanAppC.nc. Furthermore, PanC acts as a server and after collecting published data from the subscribed nodes, transmits the topics-related data to the Node-Red platform in order to send to a public channel on the ThingSpeak for representation goals.
- **RadioC:** There are 8 nodes which are responsible for connecting to the PanC, subscribing to a specific topic and publishing data related to the topic that are subscribed to. The related files are RadioC.nc and RadioAppC.nc.

You can consider the topology of the WSN which we designed and worked on it.

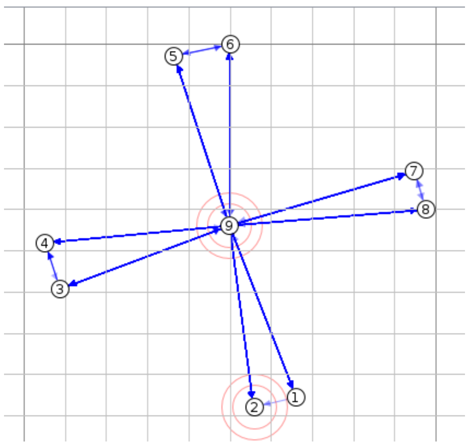


Figure 1: The WSN topology of the project

2. Files

According to this fact that the operation of nodes (RadioC) and the PAN Coordinator (PanC) are different, we declared their components, interfaces and implementations in different files categorized in two folders in which there is a separated Makefile. Thus, we used two kinds of Sky motes in Cooja based on two different compiles we got from 8 Radio nodes and the PANC.

First, we specified the structure of different messages in RadioMessages.h file. There are 5 types of messages including necessary fields. Although some fields are common among them, we separate the messages' structures for more clarification. The table below demonstrates the messages' structure and their related fields.

Message Type/ Fields	node_id	command_id	topic_id	payload
1. ConnectAckMessage	*	*		
2. SubscribeMessage	*	*	*	
3. SubscribeAckMessage	*	*		
4. PublishMessage	*	*	*	*
5. ForwardMessage	*	*		*

Table 1: General Structure of Message types and their related fields. * Means a message type contains a specific field.

In addition, some other constants and definitions like command codes and topic codes are declared in the Constants.h file. Both types of nodes use these header files to handle messages including requests, acknowledgements, and data transmissions. The considerations about subscription of nodes to the topics is illustrated in the table below.

Topics / Nodes	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
1. Temperature	*	*	*					
2. Humidity				*	*	*		
3. Luminosity							*	*

Table 2: Topic subscription of nodes. * Means a topic is subscribed by a specific node.

3. Considerations

The main approach to transmission among nodes and PANC is based on timers and events (we do not use tasks). Connection and Subscription requests and the acknowledgements should be handled in case of losing their related packets. At first step, each node tries to connect to the PANC. So, they send a connection request to the PANC and wait for the acknowledgement. If they do not receive the acknowledgement, they send again their connection request to the PANC until they get the acknowledgement. The same applies to the subscription process. The main difference between Connection and Subscription requests that we consider during the project are:

1. Subscription requests are accepted by the PANC only if the requester node connected to the PANC before. Otherwise, the request is discarded.
2. In addition to node and command identifiers, subscription requests contain a topic identifier.

It is obvious that the PANC should keep the connection status of the nodes to have the ability of checking which subscription request can be approved. It is done by using *connectionStatus* array in PanC.nc file. Moreover, the subscribed nodes to each topic should be saved by the PANC. In this case the PANC can forward the received topic-related data (random numbers) from other nodes to the nodes which are subscribed to the specific topics. The *subscribeStatus* array in PanC.nc file does this for us.

To gain a general view of the project considerations we draw the flow-chart below (for the sake of simplicity we sketched the interaction of a single node with the other entities).

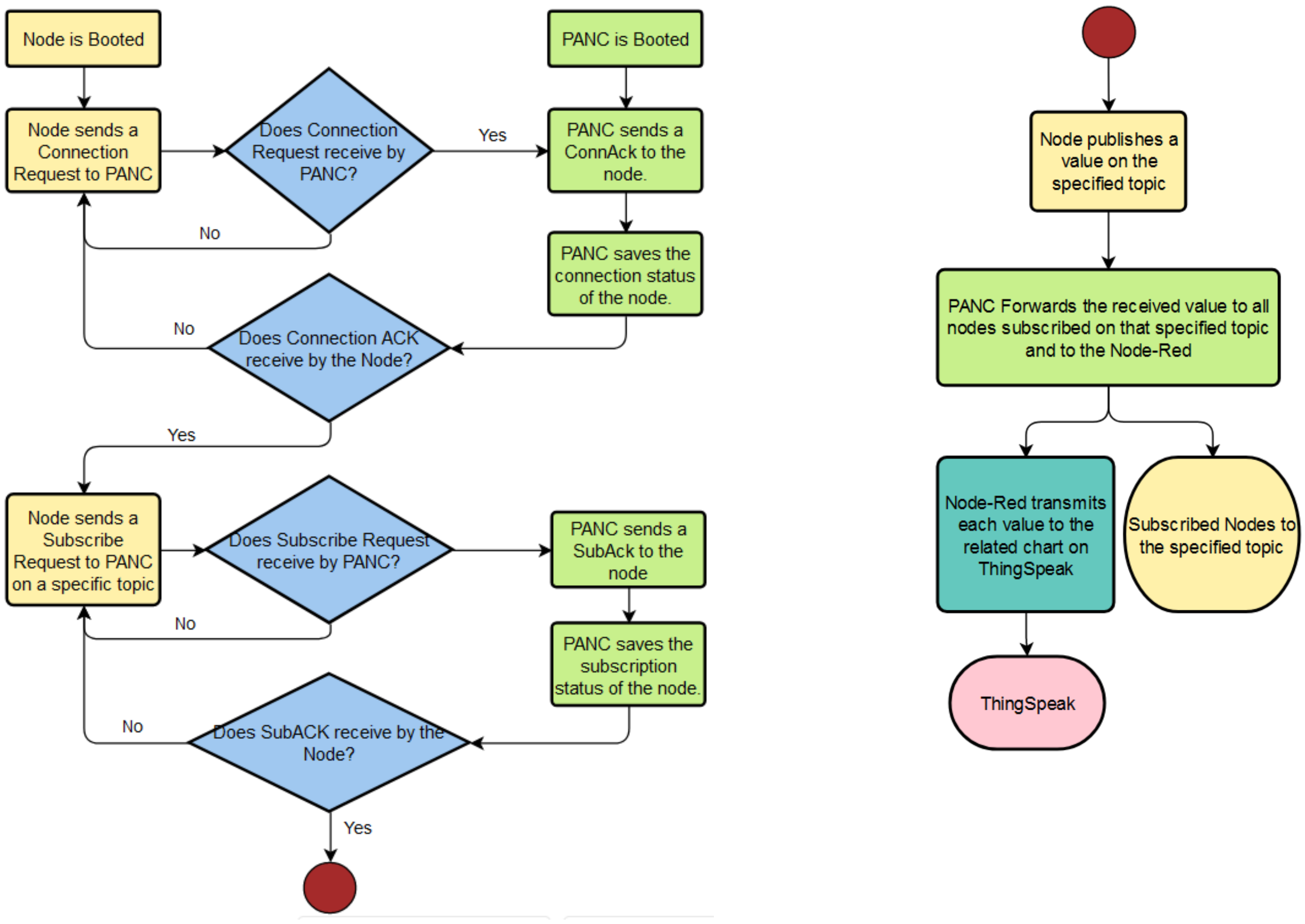

Node is Booted
Node sends a Connection Request to PANC
Does Connection Request receive by PANC?
Yes
No
PANC is Booted
PANC sends a ConnAck to the node.
PANC saves the connection status of the node.
Does Connection ACK receive by the Node?
No
Yes
Node sends a Subscribe Request to PANC on a specific topic
Does Subscribe Request receive by PANC?
No
PANC sends a SubAck to the node
PANC saves the subscription status of the node.
Does SubACK receive by the Node?
No
Yes
Node publishes a value on the specified topic
PANC Forwards the received value to all nodes subscribed on that specified topic and to the Node-Red
Node-Red transmits each value to the related chart on ThingSpeak
Subscribed Nodes to the specified topic
ThingSpeak

Figure 2: The flowchart of the project considered the interactions among a single node, the PANC, the Node-Red platform and the ThingSpeak tool.

4. Connection and Subscription

We continue our project description from the left hand-side of the flow-chart and deep more to details. In particular, *Boot.booted* event is the start point of RadioC.nc file. It is responsible for starting the node by calling *AMControl.start()* command. After successful booting of the node, a periodic timer (it is named *timer_connect*) starts to handle the transmission and the probable re-transmission of connection requests from nodes to the PANC in its *fire* event. This timer works until the connection acknowledgement is received. It is worth noting that we exploit the *TOS_NODE_ID* macro to specify the connection request of each specific node. The process of preparing the connection request packet including the node identifier (*node_id*) and the command Identifier (*command_id*) is done in *connect_request* function. The same approach is done for the subscription process which is run after the node received its connection acknowledgement. The respective event and function are *timer_subscribe.fired()* and *subscribe()*. Obviously, the node's topic is specified in the subscribe request packet based on the table 2.

5. Publish

Each node can publish its measurements about the specified topic to the PANC periodically. The main difference of Publish compared to the Connection and Subscription processes is that after the node publishes its data, it does not wait for the acknowledgement. This periodic operation is done in *timer_publish* which is fired as soon as the node receives its subscription's acknowledgement from the PANC. Since there is just a simulation, a random number is generated in each turn as the node's measurement for publishing. Also, the packet preparation is done in *publish* function. In addition to node_id and command_id, a publish packet contains topic_id and payload (the generated random number).

6. Node Receiver

There is a *Receive.receive* event in RadioC.nc that responsible for receiving all packets that are sent to the node by the PANC. According to the project considerations, there are three types of packets can be received by the node:

- Connection Request Acknowledgement
- Subscription Request Acknowledgement
- Forwarded Data

The last type of received packet is related to measured data by other nodes from the same subscribed topic which is sent by the PANC. There in no need to do a special operation for it since the level of QoS is zero and no acknowledgement is required according to the project specification.

All these types are handled using a *switch* conditional structure and consequent operations are done during the *switch*. For instance, after receiving the Connection Request Acknowledgement, the process of subscription to a topic starts. Also, after receiving Subscription Request Acknowledgement, the process of publishing measured data on the topic can be done.

7. PANC Receiver

From the PANC point of view, three types of packets can be received:

- Connection Request
- Subscription Request
- Published Data

It is worked as same as the *Receive.receive* event in RadioC.nc but its reactions to the received packets are different. As soon as the connection request is received, the PANC sends the Connection Request Acknowledgement to the requester node and changes the connection status of that node to connected in *connectionStatus* array. Moreover, when the PANC receives a subscription request, it checks the connection status of the requester node before sending Subscription Request Acknowledgement. Subscription requests of nodes which are not connected to the PANC will be discarded. After connecting and subscribing of nodes, the PANC expects data related to each topic from the subscribed nodes. It sends the received data to:

- **The nodes which are subscribed to that topic:** It can be done because we save the *topic_id* of each subscribe requester node after the PANC ensures about connected status of that node.
- **The Node-Red platform:** In Node-Red we can get the data using a TCP input node which is connected to the server (PANC) port on the localhost. After extracting the topic and its value via *Function* node, the value can be sent to the ThingSpeak online tool for representing on the related chart which is considered for each topic using MQTT output node. The channel identifier which is specified in the ThingSpeak is **2203129** which we used it to set ***channels/2203129/publish*** to the topic field in the MQTT output node. The pictures below show the charts assigned to the three topics separately.

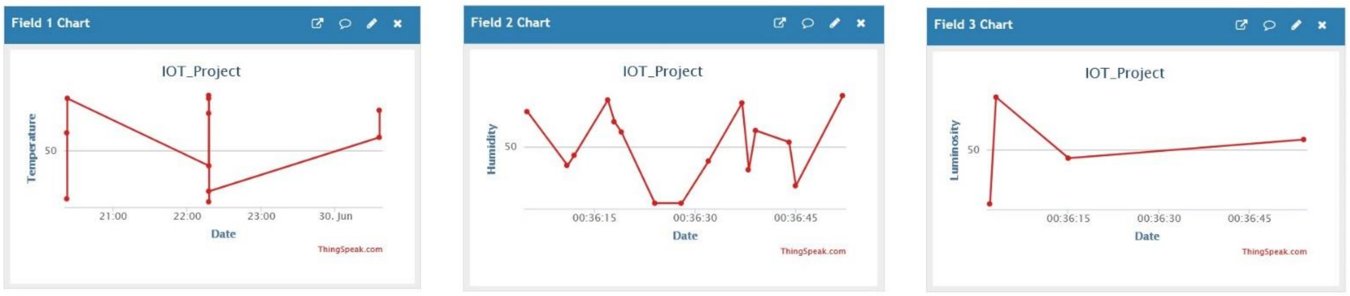
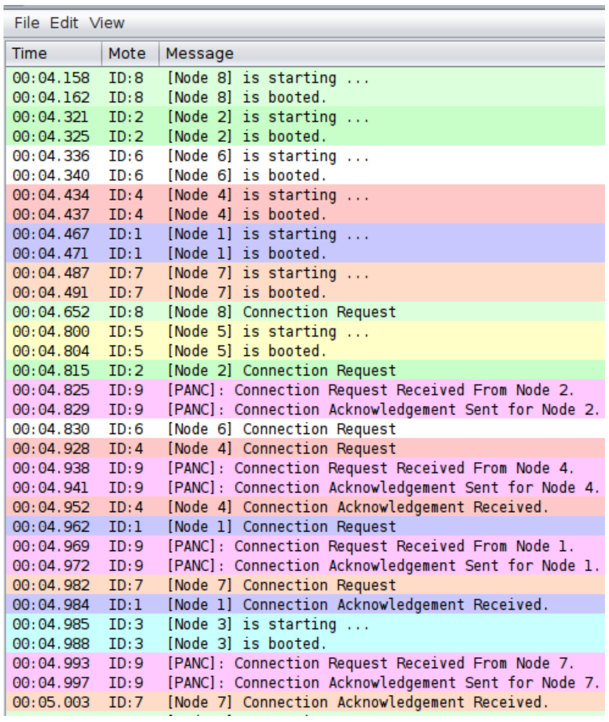


Figure 3: Left picture shows the temperature values, the center one is related to the humidity measures and the right picture displays the Luminosity values.

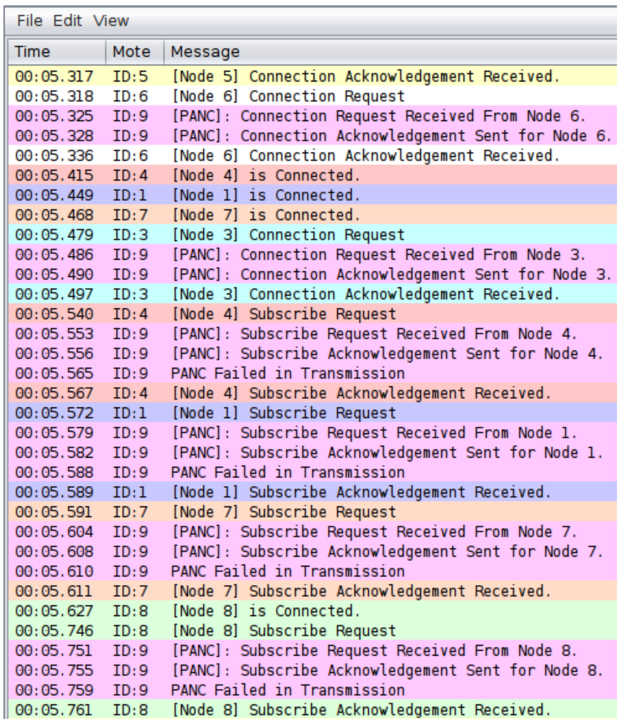
8. PANC Transmitter

The PANC exploits *send_connection_ack* and *send_subscribe_ack* functions to reacts the received requests and send the related acknowledgements to the requester node. It is worth noting that the PANC has a lazy paradigm to the transmitted acknowledgements. It means that when an acknowledgement is sent, the PANC does not follow the delivering of it. Therefore, if a node does not receive the acknowledgement of its request, it has to send again its request to the PANC. According to the explanation above, there is not any timer in the PANC since the QoS level is zero. The last point that we would like to mention is the *forward* function which is used for forwarding the published data from each node to the subscribed nodes on the specific topic and to the Node-Red platform. Other structures like lock mechanism, messages and variables declarations are nearly identical to the RadioC.nc file and are avoided to write in the report to keep it brief.

The pictures below illustrate some parts of different operations log which are done during the project running in the Cooja simulation environment.

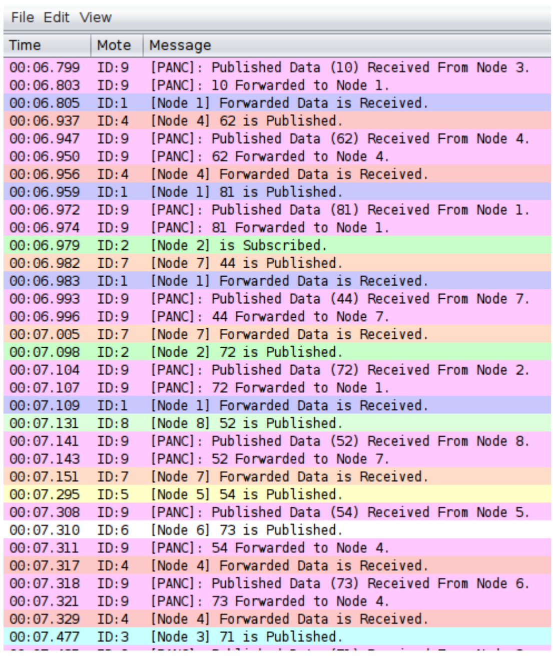


```
File  Edit  View
Time       Mote   Message
00:04.158  ID:8   [Node 8] is starting ...
00:04.162  ID:8   [Node 8] is booted.
00:04.321  ID:2   [Node 2] is starting ...
00:04.325  ID:2   [Node 2] is booted.
00:04.336  ID:6   [Node 6] is starting ...
00:04.340  ID:6   [Node 6] is booted.
00:04.434  ID:4   [Node 4] is starting ...
00:04.437  ID:4   [Node 4] is booted.
00:04.467  ID:1   [Node 1] is starting ...
00:04.471  ID:1   [Node 1] is booted.
00:04.487  ID:7   [Node 7] is starting ...
00:04.491  ID:7   [Node 7] is booted.
00:04.652  ID:8   [Node 8] Connection Request
00:04.800  ID:5   [Node 5] is starting ...
00:04.804  ID:5   [Node 5] is booted.
00:04.815  ID:2   [Node 2] Connection Request
00:04.825  ID:9   [PANC]: Connection Request Received From Node 2.
00:04.829  ID:9   [PANC]: Connection Acknowledgement Sent for Node 2.
00:04.830  ID:6   [Node 6] Connection Request
00:04.928  ID:4   [Node 4] Connection Request
00:04.938  ID:9   [PANC]: Connection Request Received From Node 4.
00:04.941  ID:9   [PANC]: Connection Acknowledgement Sent for Node 4.
00:04.952  ID:4   [Node 4] Connection Acknowledgement Received.
00:04.962  ID:1   [Node 1] Connection Request
00:04.969  ID:9   [PANC]: Connection Request Received From Node 1.
00:04.972  ID:9   [PANC]: Connection Acknowledgement Sent for Node 1.
00:04.982  ID:7   [Node 7] Connection Request
00:04.984  ID:1   [Node 1] Connection Acknowledgement Received.
00:04.985  ID:3   [Node 3] is starting ...
00:04.988  ID:3   [Node 3] is booted.
00:04.993  ID:9   [PANC]: Connection Request Received From Node 7.
00:04.997  ID:9   [PANC]: Connection Acknowledgement Sent for Node 7.
00:05.003  ID:7   [Node 7] Connection Acknowledgement Received.
```

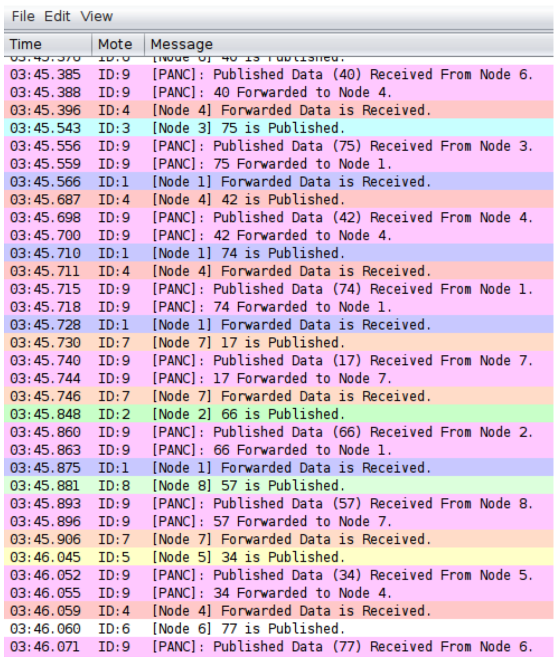


```
File  Edit  View
Time       Mote   Message
00:05.317  ID:5   [Node 5] Connection Acknowledgement Received.
00:05.318  ID:6   [Node 6] Connection Request
00:05.325  ID:9   [PANC]: Connection Request Received From Node 6.
00:05.328  ID:9   [PANC]: Connection Acknowledgement Sent for Node 6.
00:05.336  ID:6   [Node 6] Connection Acknowledgement Received.
00:05.415  ID:4   [Node 4] is Connected.
00:05.449  ID:1   [Node 1] is Connected.
00:05.468  ID:7   [Node 7] is Connected.
00:05.479  ID:3   [Node 3] Connection Request
00:05.486  ID:9   [PANC]: Connection Request Received From Node 3.
00:05.490  ID:9   [PANC]: Connection Acknowledgement Sent for Node 3.
00:05.497  ID:3   [Node 3] Connection Acknowledgement Received.
00:05.540  ID:4   [Node 4] Subscribe Request
00:05.553  ID:9   [PANC]: Subscribe Request Received From Node 4.
00:05.556  ID:9   [PANC]: Subscribe Acknowledgement Sent for Node 4.
00:05.565  ID:9   PANC Failed in Transmission
00:05.567  ID:4   [Node 4] Subscribe Acknowledgement Received.
00:05.572  ID:1   [Node 1] Subscribe Request
00:05.579  ID:9   [PANC]: Subscribe Request Received From Node 1.
00:05.582  ID:9   [PANC]: Subscribe Acknowledgement Sent for Node 1.
00:05.588  ID:9   PANC Failed in Transmission
00:05.589  ID:1   [Node 1] Subscribe Acknowledgement Received.
00:05.591  ID:7   [Node 7] Subscribe Request
00:05.604  ID:9   [PANC]: Subscribe Request Received From Node 7.
00:05.608  ID:9   [PANC]: Subscribe Acknowledgement Sent for Node 7.
00:05.610  ID:9   PANC Failed in Transmission
00:05.611  ID:7   [Node 7] Subscribe Acknowledgement Received.
00:05.627  ID:8   [Node 8] is Connected.
00:05.746  ID:8   [Node 8] Subscribe Request
00:05.751  ID:9   [PANC]: Subscribe Request Received From Node 8.
00:05.755  ID:9   [PANC]: Subscribe Acknowledgement Sent for Node 8.
00:05.759  ID:9   PANC Failed in Transmission
00:05.761  ID:8   [Node 8] Subscribe Acknowledgement Received.
```

Figure 4: Left picture shows the booting, connection request and connection acknowledgement processes and the right picture displays the subscription of nodes and related acknowledgements.

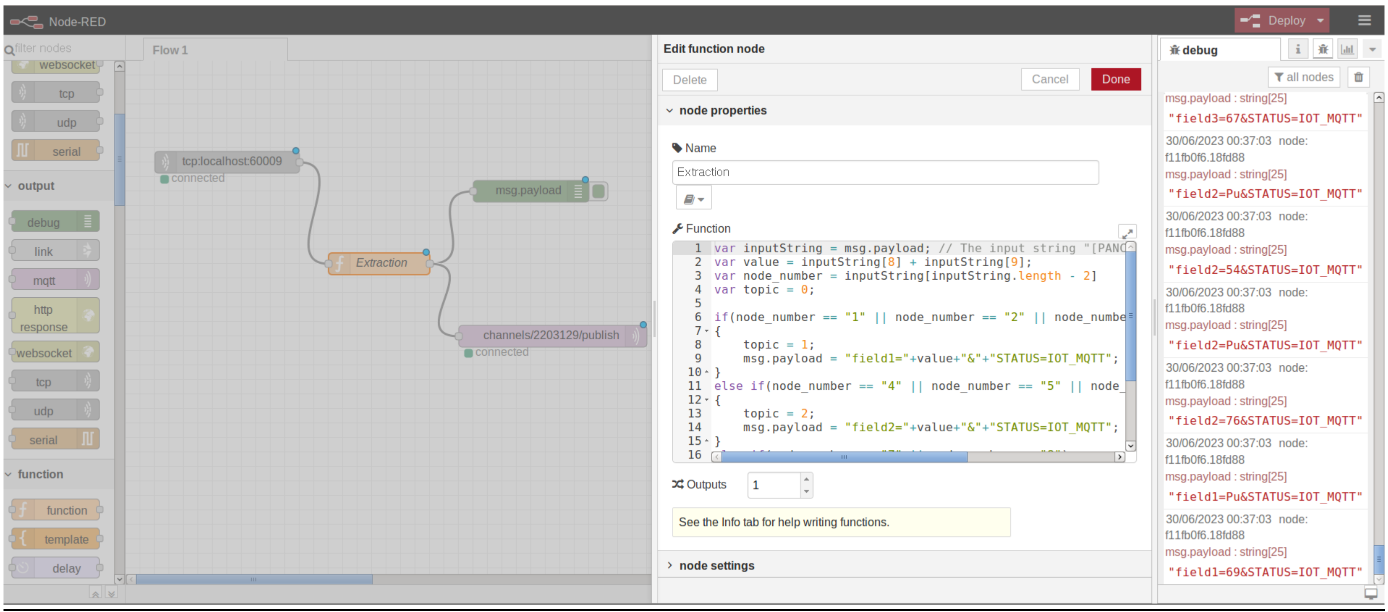


```
Time       Mote  Message
00:06.799  ID:9  [PANC]: Published Data (10) Received From Node 3.
00:06.803  ID:9  [PANC]: 10 Forwarded to Node 1.
00:06.805  ID:1  [Node 1] Forwarded Data is Received.
00:06.937  ID:4  [Node 4] 62 is Published.
00:06.947  ID:9  [PANC]: Published Data (62) Received From Node 4.
00:06.950  ID:9  [PANC]: 62 Forwarded to Node 4.
00:06.956  ID:4  [Node 4] Forwarded Data is Received.
00:06.959  ID:1  [Node 1] 81 is Published.
00:06.972  ID:9  [PANC]: Published Data (81) Received From Node 1.
00:06.974  ID:9  [PANC]: 81 Forwarded to Node 1.
00:06.979  ID:2  [Node 2] is Subscribed.
00:06.982  ID:7  [Node 7] 44 is Published.
00:06.983  ID:1  [Node 1] Forwarded Data is Received.
00:06.993  ID:9  [PANC]: Published Data (44) Received From Node 7.
00:06.996  ID:9  [PANC]: 44 Forwarded to Node 7.
00:07.005  ID:7  [Node 7] Forwarded Data is Received.
00:07.098  ID:2  [Node 2] 72 is Published.
00:07.104  ID:9  [PANC]: Published Data (72) Received From Node 2.
00:07.107  ID:9  [PANC]: 72 Forwarded to Node 1.
00:07.109  ID:1  [Node 1] Forwarded Data is Received.
00:07.131  ID:8  [Node 8] 52 is Published.
00:07.141  ID:9  [PANC]: Published Data (52) Received From Node 8.
00:07.143  ID:9  [PANC]: 52 Forwarded to Node 7.
00:07.151  ID:7  [Node 7] Forwarded Data is Received.
00:07.295  ID:5  [Node 5] 54 is Published.
00:07.308  ID:9  [PANC]: Published Data (54) Received From Node 5.
00:07.310  ID:6  [Node 6] 73 is Published.
00:07.311  ID:9  [PANC]: 54 Forwarded to Node 4.
00:07.317  ID:4  [Node 4] Forwarded Data is Received.
00:07.318  ID:9  [PANC]: Published Data (73) Received From Node 6.
00:07.321  ID:9  [PANC]: 73 Forwarded to Node 4.
00:07.329  ID:4  [Node 4] Forwarded Data is Received.
00:07.477  ID:3  [Node 3] 71 is Published.
```



```
Time       Mote  Message
03:45.385  ID:9  [PANC]: Published Data (40) Received From Node 6.
03:45.388  ID:9  [PANC]: 40 Forwarded to Node 4.
03:45.396  ID:4  [Node 4] Forwarded Data is Received.
03:45.543  ID:3  [Node 3] 75 is Published.
03:45.556  ID:9  [PANC]: Published Data (75) Received From Node 3.
03:45.559  ID:9  [PANC]: 75 Forwarded to Node 1.
03:45.566  ID:1  [Node 1] Forwarded Data is Received.
03:45.687  ID:4  [Node 4] 42 is Published.
03:45.698  ID:9  [PANC]: Published Data (42) Received From Node 4.
03:45.700  ID:9  [PANC]: 42 Forwarded to Node 4.
03:45.710  ID:1  [Node 1] 74 is Published.
03:45.711  ID:4  [Node 4] Forwarded Data is Received.
03:45.715  ID:9  [PANC]: Published Data (74) Received From Node 1.
03:45.718  ID:9  [PANC]: 74 Forwarded to Node 1.
03:45.728  ID:1  [Node 1] Forwarded Data is Received.
03:45.730  ID:7  [Node 7] 17 is Published.
03:45.740  ID:9  [PANC]: Published Data (17) Received From Node 7.
03:45.744  ID:9  [PANC]: 17 Forwarded to Node 7.
03:45.746  ID:7  [Node 7] Forwarded Data is Received.
03:45.848  ID:2  [Node 2] 66 is Published.
03:45.860  ID:9  [PANC]: Published Data (66) Received From Node 2.
03:45.863  ID:9  [PANC]: 66 Forwarded to Node 1.
03:45.875  ID:1  [Node 1] Forwarded Data is Received.
03:45.881  ID:8  [Node 8] 57 is Published.
03:45.893  ID:9  [PANC]: Published Data (57) Received From Node 8.
03:45.896  ID:9  [PANC]: 57 Forwarded to Node 7.
03:45.906  ID:7  [Node 7] Forwarded Data is Received.
03:46.045  ID:5  [Node 5] 34 is Published.
03:46.052  ID:9  [PANC]: Published Data (34) Received From Node 5.
03:46.055  ID:9  [PANC]: 34 Forwarded to Node 4.
03:46.059  ID:4  [Node 4] Forwarded Data is Received.
03:46.060  ID:6  [Node 6] 77 is Published.
03:46.071  ID:9  [PANC]: Published Data (77) Received From Node 6.
```

Figure 5: Left picture shows the publish and forward processes. The right picture displays the log of operations after more than 3 minutes running.



Name: Extraction

```
var inputString = msg.payload; // The input string "[PANC
var value = inputString[8] + inputString[9];
var node_number = inputString[inputString.length - 2]
var topic = 0;

if(node_number == "1" || node_number == "2" || node_numbe
{
    topic = 1;
    msg.payload = "field1="+value+"&"+"STATUS=IOT_MQTT";
}
else if(node_number == "4" || node_number == "5" || node_
{
    topic = 2;
    msg.payload = "field2="+value+"&"+"STATUS=IOT_MQTT";
}
```

Outputs: 1

See the Info tab for help writing functions.

```
msg.payload : string[25]
"field3=67&STATUS=IOT_MQTT"
30/06/2023 00:37:03  node: f11fb0f6.18fd88
msg.payload : string[25]
"field2=Pu&STATUS=IOT_MQTT"
30/06/2023 00:37:03  node: f11fb0f6.18fd88
msg.payload : string[25]
"field2=54&STATUS=IOT_MQTT"
30/06/2023 00:37:03  node: f11fb0f6.18fd88
msg.payload : string[25]
"field2=Pu&STATUS=IOT_MQTT"
30/06/2023 00:37:03  node: f11fb0f6.18fd88
msg.payload : string[25]
"field2=76&STATUS=IOT_MQTT"
30/06/2023 00:37:03  node: f11fb0f6.18fd88
msg.payload : string[25]
"field1=Pu&STATUS=IOT_MQTT"
30/06/2023 00:37:03  node: f11fb0f6.18fd88
msg.payload : string[25]
"field1=69&STATUS=IOT_MQTT"
```

Figure 6: The Node-Red platform and the codes which are written to extract the values of different topics