# Project-2
# RPL Attacks

**Prepared By : Lohith Reddy Kalluru and Shubham Makwana**

## Attack:

RPL is a routing protocol for low-power and lossy constrained node networks.It creates a tree-like routing topology called the destination-oriented directed acyclic graph (DODAG), which is in the direction towards one or more nodes known as root node or sink node. RPL protocols are used with resource constraint nodes.

• **DODAG Information Object (DIO)**:it stores information including rank of a node, RPL Instance, the IPv6 address of the root or sink and so on.
• **Destination Advertisement Object (DAO)**: it consists of information that can used for downward traffic towards child nodes.
• **DODAG Information Solicitation (DIS)**: Used by nodes to request graph related information from the neighboring nodes.
• **Destination Advertisement Object Acknowledgement (DAOACK)**: Sent by a DAO recipient in response to a DAO message.

Whenever a new node enters into a rpl network, it starts sending a DIS message to all nodes and waits for DIO message to be received whcih contains details regarding node id and objective code point. The DIO messages are broadcasted at particular intervals based on the trickle algorithm. The node on recieving the info , calculates the rank . and based on that it selects a parent . To send a message downwards , node should send a DAO message containing the routable fixes up the tree.To prevent the loops ,RPL does not allow the data going in down direction and sent from a desecendent . RPL uses two header options which Is flow of direction (O)and rank error(R) . Rank error is a flag set when there is a mismatch in the rank of sender and direction of flow.

When a malicious node is introduced into the network ,it can manipulate the header-options used by RPL to track DODAG. Once the header-options are manipulated , the malicious node can cause denial of service attacks ,drain power from nodes and also can target certain nodes by creating black hole.

### DODAG attack using fit iot lab

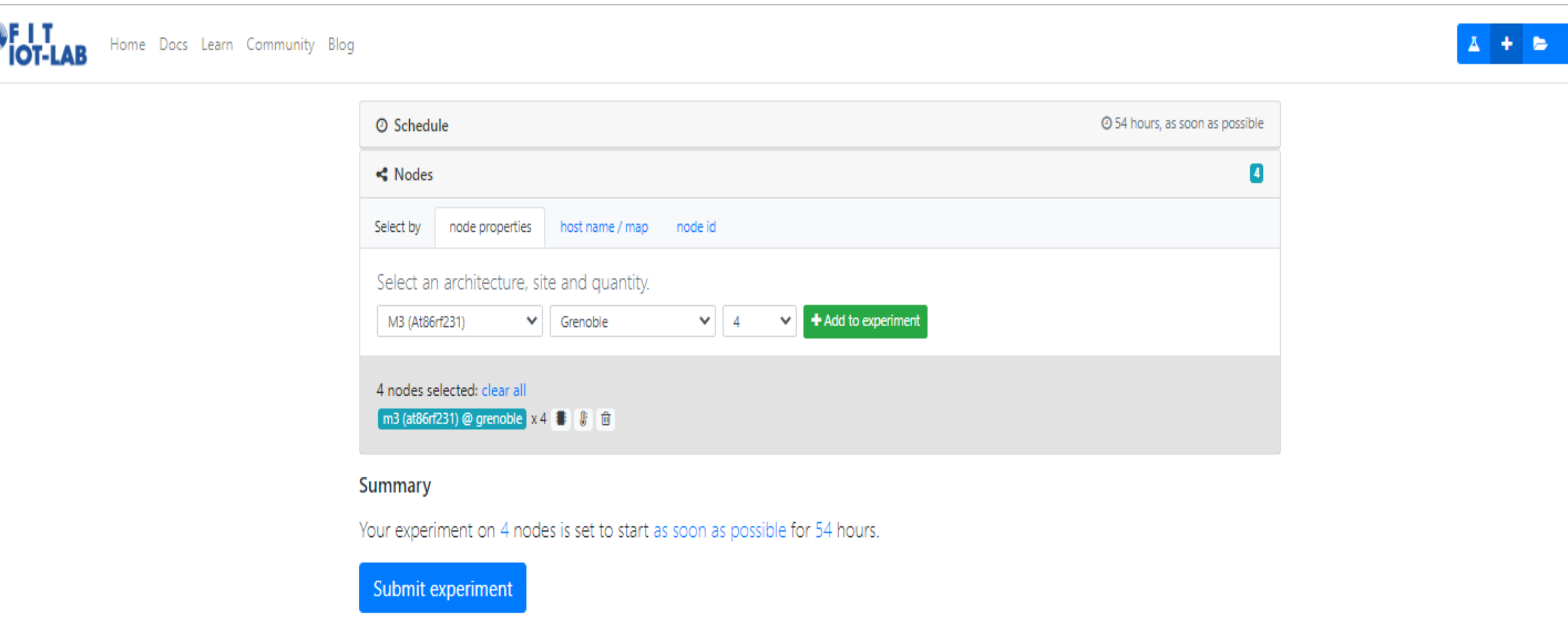
Here we have implemented a flood attack on a real testbed using the m3 node ( at86rf231) at site=grenoble. We flashed 1 up-UDP-server and 3 RPL-UDP-clients. Here we have considered a flooding attack. The power consumption is shown in the figure. We modify the DIS-related code to define multiple DIS constants which causes the node to send DIS messages repeatedly.

We also modify the timers to send the multiple DIS messages. We start a serial -aggregator and power consumption monitor which is used to capture the packets and also measure the power consumed at each node.

# Process

1. **Implementation of Flooding Attack (DDOAG attack)**

**Created 4 nodes with grenoble  m3**



54 hours, as soon as possible

Select an architecture, site and quantity.

M3 (At86rf231) | Grenoble | 4

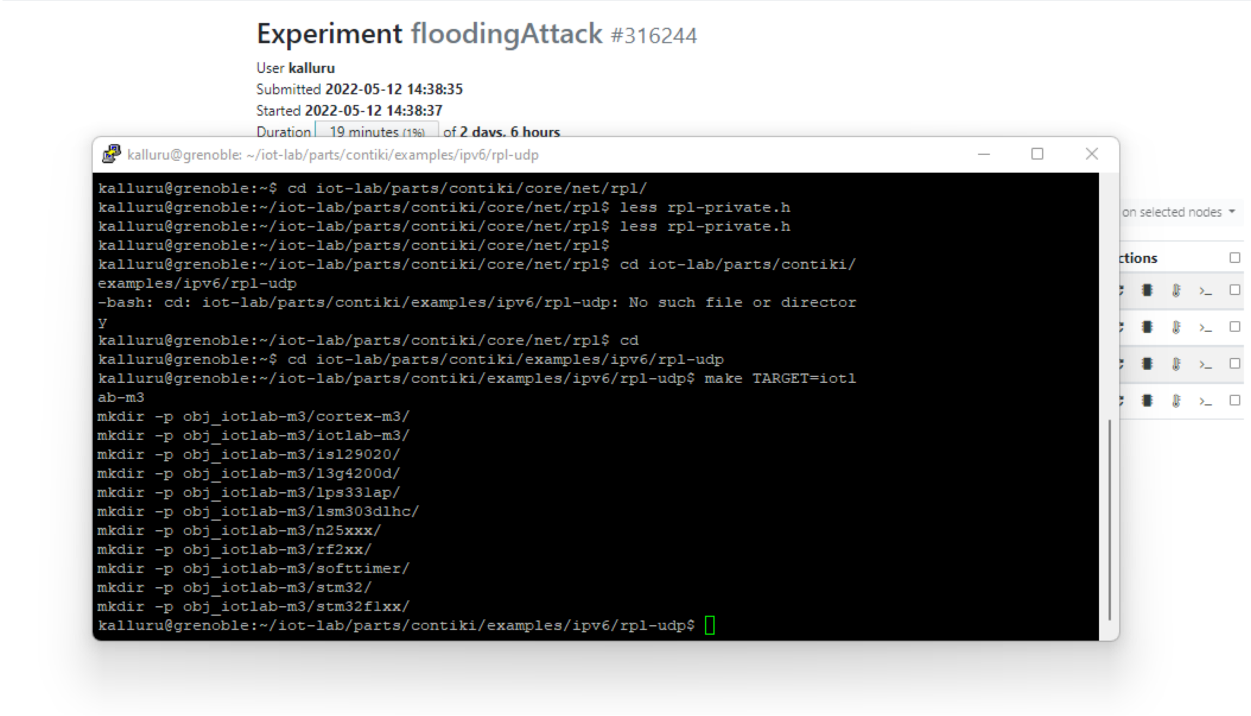
4 nodes selected: clear all

m3 (at86rf231) @ grenoble x 4

Summary

Your experiment on 4 nodes is set to start as soon as possible for 54 hours.

**Compile firmware for target iotlab-m3**



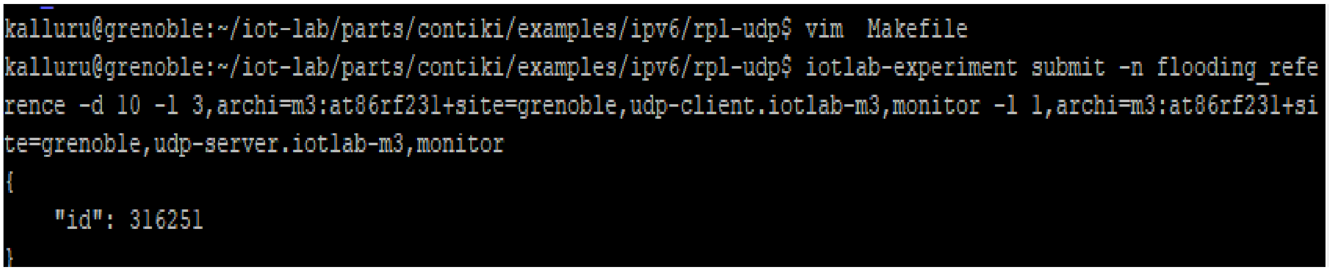
Experiment floodingAttack #316244

User kalluru
Submitted 2022-05-12 14:38:35
Started 2022-05-12 14:38:37
Duration 19 minutes (1%) of 2 days, 6 hours

kalluru@grenoble: ~/iot-lab/parts/contiki/examples/ipv6/rpl-udp
on selected nodes
ctions

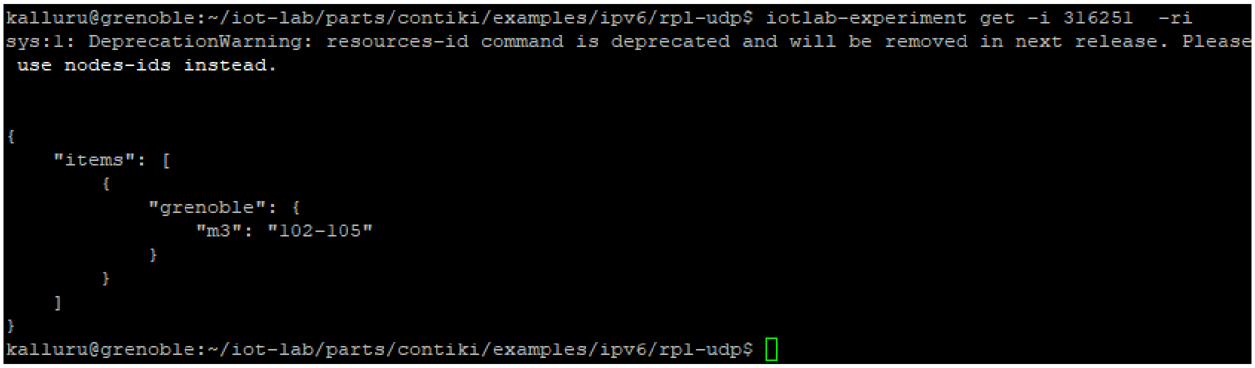
```
kalluru@grenoble:~$ cd iot-lab/parts/contiki/core/net/rpl/
kalluru@grenoble:~/iot-lab/parts/contiki/core/net/rpl$ less rpl-private.h
kalluru@grenoble:~/iot-lab/parts/contiki/core/net/rpl$ less rpl-private.h
kalluru@grenoble:~/iot-lab/parts/contiki/core/net/rpl$
kalluru@grenoble:~/iot-lab/parts/contiki/core/net/rpl$ cd iot-lab/parts/contiki/
examples/ipv6/rpl-udp
-bash: cd: iot-lab/parts/contiki/examples/ipv6/rpl-udp: No such file or director
y
kalluru@grenoble:~/iot-lab/parts/contiki/core/net/rpl$ cd
kalluru@grenoble:~$ cd iot-lab/parts/contiki/examples/ipv6/rpl-udp
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ make TARGET=iotl
ab-m3
mkdir -p obj_iotlab-m3/cortex-m3/
mkdir -p obj_iotlab-m3/iotlab-m3/
mkdir -p obj_iotlab-m3/isl29020/
mkdir -p obj_iotlab-m3/l3g4200d/
mkdir -p obj_iotlab-m3/lps331ap/
mkdir -p obj_iotlab-m3/lsm303dlhc/
mkdir -p obj_iotlab-m3/n25xxx/
mkdir -p obj_iotlab-m3/rf2xx/
mkdir -p obj_iotlab-m3/softtimer/
mkdir -p obj_iotlab-m3/stm32/
mkdir -p obj_iotlab-m3/stm32f1xx/
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$
```

## Consumption monitoring



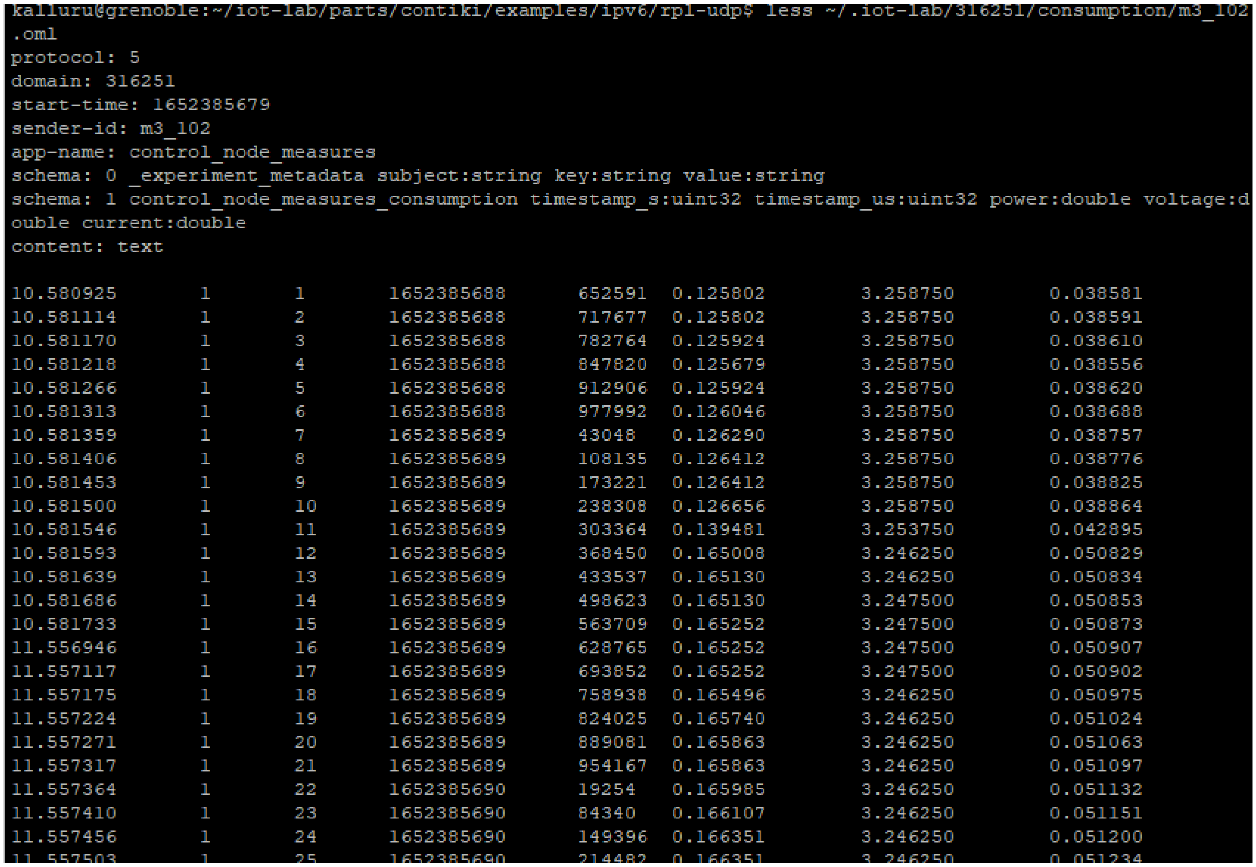
```
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ vim  Makefile
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ iotlab-experiment submit -n flooding_refe
rence -d 10 -l 3,archi=m3:at86rf231+site=grenoble,udp-client.iotlab-m3,monitor -l 1,archi=m3:at86rf231+si
te=grenoble,udp-server.iotlab-m3,monitor
{
    "id": 316251
}
```

## iotlab-experiment get -i 316251 -ri  to check the nodes



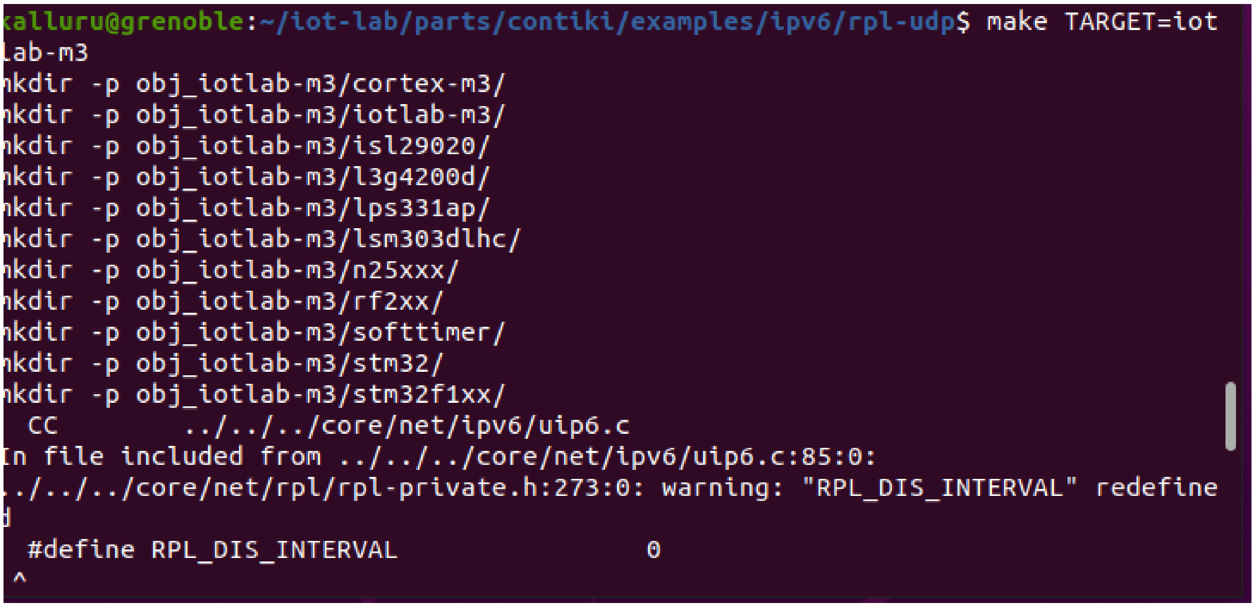
```
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ iotlab-experiment get -i 316251  -ri
sys:1: DeprecationWarning: resources-id command is deprecated and will be removed in next release. Please
 use nodes-ids instead.


{
    "items": [
        {
            "grenoble": {
                "m3": "102-105"
            }
        }
    ]
}
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$
```

**Measured consumption data is shown below  without running the flood attack (m3-102)**



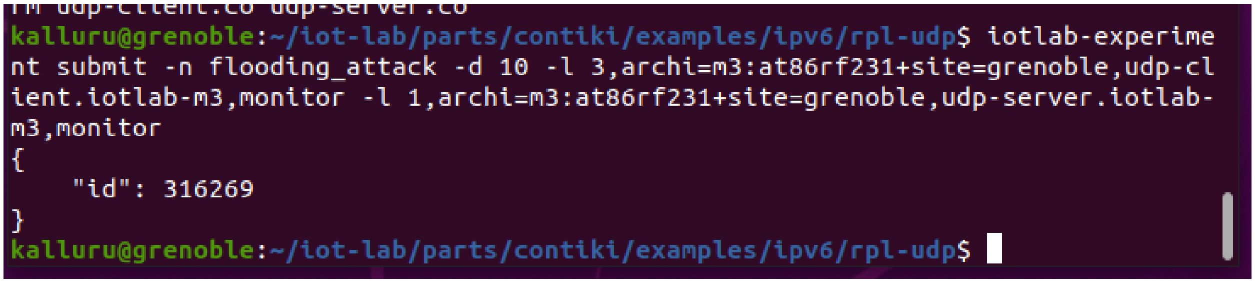
```
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ less ~/.iot-lab/316251/consumption/m3_102
.oml
protocol: 5
domain: 316251
start-time: 1652385679
sender-id: m3_102
app-name: control_node_measures
schema: 0 _experiment_metadata subject:string key:string value:string
schema: 1 control_node_measures_consumption timestamp_s:uint32 timestamp_us:uint32 power:double voltage:d
ouble current:double
content: text

10.580925       1       1       1652385688      652591  0.125802        3.258750        0.038581
10.581114       1       2       1652385688      717677  0.125802        3.258750        0.038591
10.581170       1       3       1652385688      782764  0.125924        3.258750        0.038610
10.581218       1       4       1652385688      847820  0.125679        3.258750        0.038556
10.581266       1       5       1652385688      912906  0.125924        3.258750        0.038620
10.581313       1       6       1652385688      977992  0.126046        3.258750        0.038688
10.581359       1       7       1652385689      43048   0.126290        3.258750        0.038757
10.581406       1       8       1652385689      108135  0.126412        3.258750        0.038776
10.581453       1       9       1652385689      173221  0.126412        3.258750        0.038825
10.581500       1       10      1652385689      238308  0.126656        3.258750        0.038864
10.581546       1       11      1652385689      303364  0.139481        3.253750        0.042895
10.581593       1       12      1652385689      368450  0.165008        3.246250        0.050829
10.581639       1       13      1652385689      433537  0.165130        3.246250        0.050834
10.581686       1       14      1652385689      498623  0.165130        3.247500        0.050853
10.581733       1       15      1652385689      563709  0.165252        3.247500        0.050873
11.556946       1       16      1652385689      628765  0.165252        3.247500        0.050907
11.557117       1       17      1652385689      693852  0.165252        3.247500        0.050902
11.557175       1       18      1652385689      758938  0.165496        3.246250        0.050975
11.557224       1       19      1652385689      824025  0.165740        3.246250        0.051024
11.557271       1       20      1652385689      889081  0.165863        3.246250        0.051063
11.557317       1       21      1652385689      954167  0.165863        3.246250        0.051097
11.557364       1       22      1652385690      19254   0.165985        3.246250        0.051132
11.557410       1       23      1652385690      84340   0.166107        3.246250        0.051151
11.557456       1       24      1652385690      149396  0.166351        3.246250        0.051200
11.557503       1       25      1652385690      214482  0.166351        3.246250        0.051234
```

### Compile the firmware for flooding attack code



```
alluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ make TARGET=iot
lab-m3
nkdir -p obj_iotlab-m3/cortex-m3/
nkdir -p obj_iotlab-m3/iotlab-m3/
nkdir -p obj_iotlab-m3/isl29020/
nkdir -p obj_iotlab-m3/l3g4200d/
nkdir -p obj_iotlab-m3/lps331ap/
nkdir -p obj_iotlab-m3/lsm303dlhc/
nkdir -p obj_iotlab-m3/n25xxx/
nkdir -p obj_iotlab-m3/rf2xx/
nkdir -p obj_iotlab-m3/softtimer/
nkdir -p obj_iotlab-m3/stm32/
nkdir -p obj_iotlab-m3/stm32f1xx/
  CC         ../../../core/net/ipv6/uip6.c
In file included from ../../../core/net/ipv6/uip6.c:85:0:
../../../core/net/rpl/rpl-private.h:273:0: warning: "RPL_DIS_INTERVAL" redefine
d
  #define RPL_DIS_INTERVAL                0
 ^
```

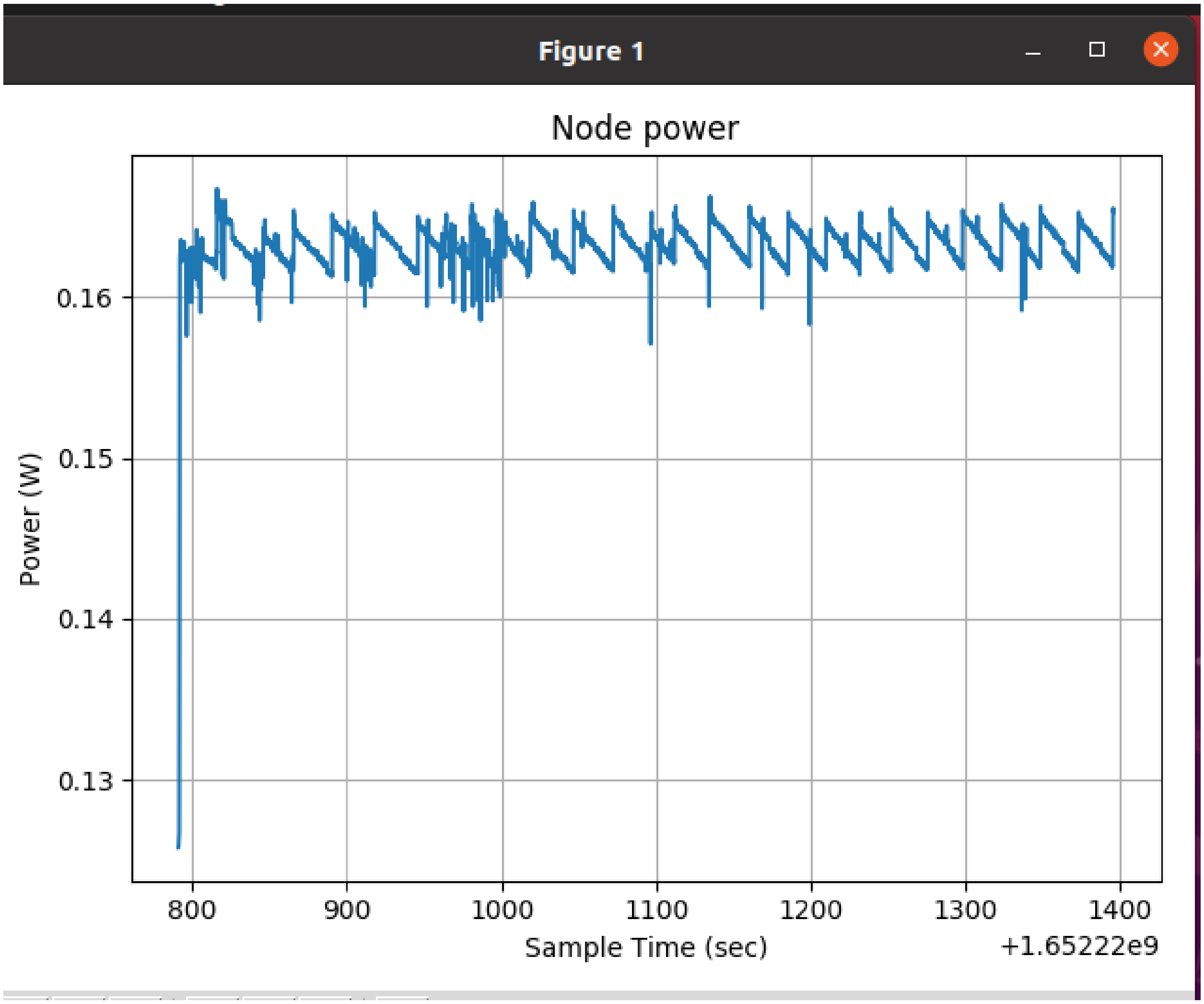
### Create the experiment by flashing with malicious nodes with flooding attack



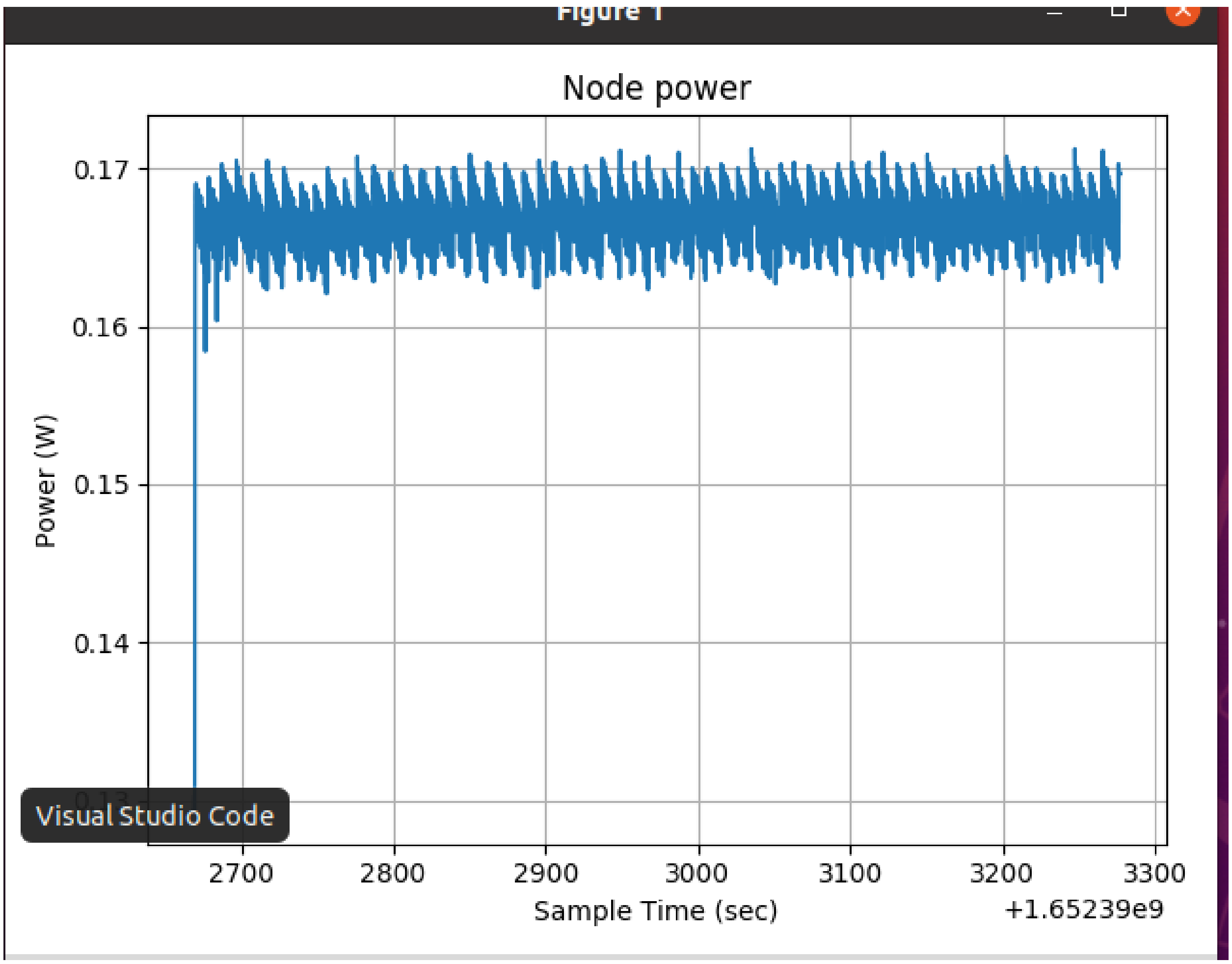
```
rm udp-client.co udp-server.co
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ iotlab-experime
nt submit -n flooding_attack -d 10 -l 3,archi=m3:at86rf231+site=grenoble,udp-cl
ient.iotlab-m3,monitor -l 1,archi=m3:at86rf231+site=grenoble,udp-server.iotlab-
m3,monitor
{
    "id": 316269
}
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ 
```

## RESULTS

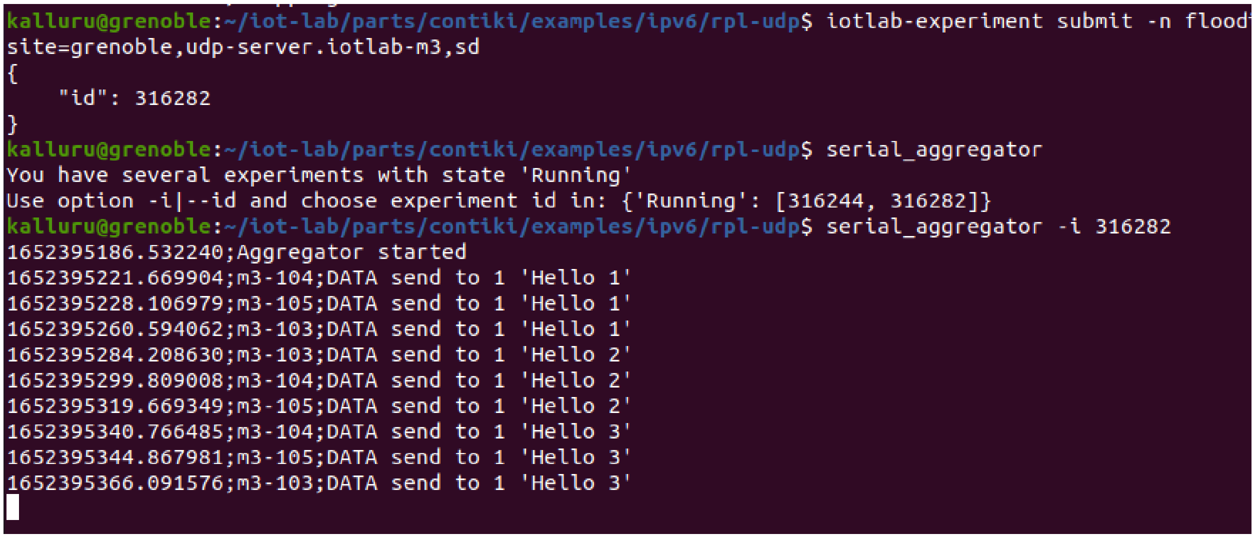
**Before Intrusion power consumed by node:**



**When the graph is plotted for one of the nodes we observe that power consumption has been increased**

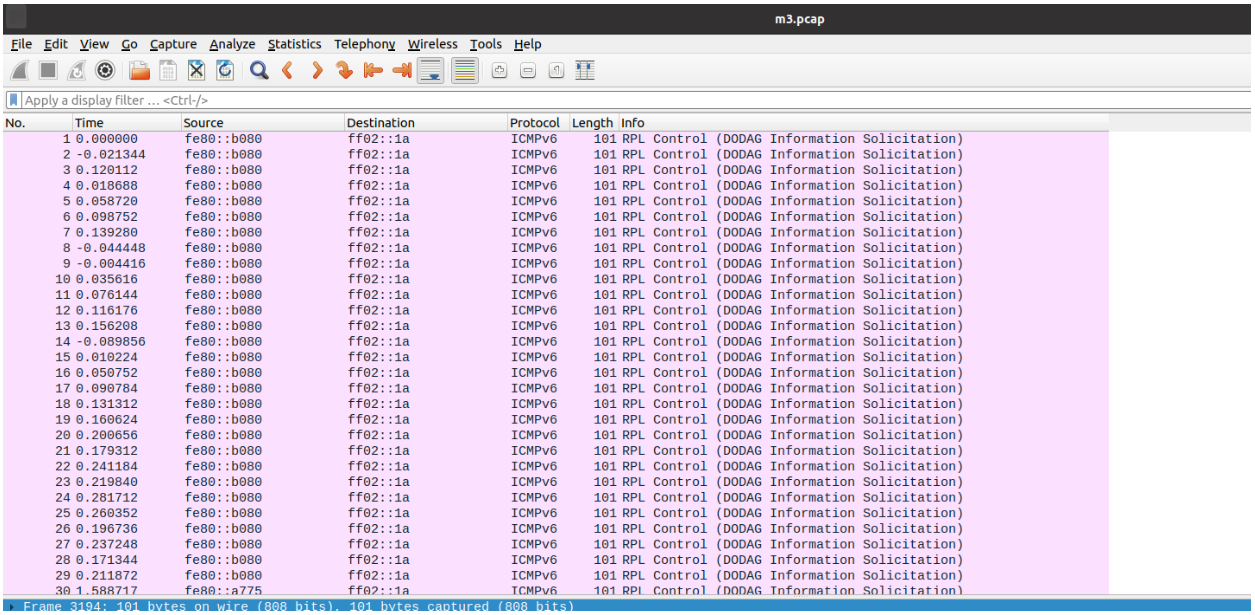


**The below output shows the various datapackets captured by monitoring**



```
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ iotlab-experiment submit -n flood
site=grenoble,udp-server.iotlab-m3,sd
{
    "id": 316282
}
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ serial_aggregator
You have several experiments with state 'Running'
Use option -i|--id and choose experiment id in: {'Running': [316244, 316282]}
kalluru@grenoble:~/iot-lab/parts/contiki/examples/ipv6/rpl-udp$ serial_aggregator -i 316282
1652395186.532240;Aggregator started
1652395221.669904;m3-104;DATA send to 1 'Hello 1'
1652395228.106979;m3-105;DATA send to 1 'Hello 1'
1652395260.594062;m3-103;DATA send to 1 'Hello 1'
1652395284.208630;m3-103;DATA send to 1 'Hello 2'
1652395299.809008;m3-104;DATA send to 1 'Hello 2'
1652395319.669349;m3-105;DATA send to 1 'Hello 2'
1652395340.766485;m3-104;DATA send to 1 'Hello 3'
1652395344.867981;m3-105;DATA send to 1 'Hello 3'
1652395366.091576;m3-103;DATA send to 1 'Hello 3'
```

**Wireshark capture is shown below .**
**It indicates the various dodag information solicitation**



| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 2 | -0.021344 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 3 | 0.120112 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 4 | 0.018688 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 5 | 0.058720 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 6 | 0.098752 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 7 | 0.139280 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 8 | -0.044448 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 9 | -0.004416 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 10 | 0.035616 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 11 | 0.076144 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 12 | 0.116176 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 13 | 0.156208 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 14 | -0.089856 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 15 | 0.010224 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 16 | 0.050752 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 17 | 0.090784 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 18 | 0.131312 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 19 | 0.160624 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 20 | 0.200656 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 21 | 0.179312 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 22 | 0.241184 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 23 | 0.219840 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 24 | 0.281712 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 25 | 0.260352 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 26 | 0.196736 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 27 | 0.237248 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 28 | 0.171344 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 29 | 0.211872 | fe80::b080 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |
| 30 | 1.588717 | fe80::a775 | ff02::1a | ICMPv6 | 101 | RPL Control (DODAG Information Solicitation) |

Frame 3194: 101 bytes on wire (808 bits), 101 bytes captured (808 bits)

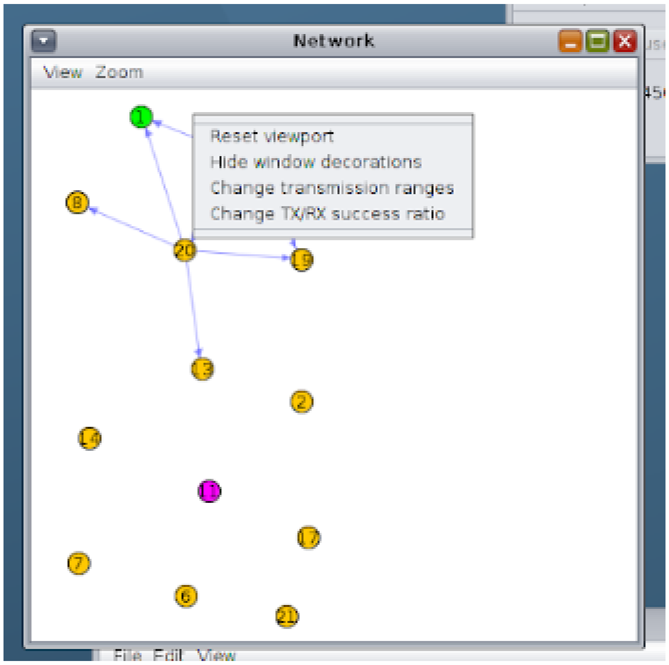
### 2. DoDAG/Blackhole attack using Contiki and cooja simulator(by modifying ranks and flow of direction of packets)

**<u>Process:</u>**

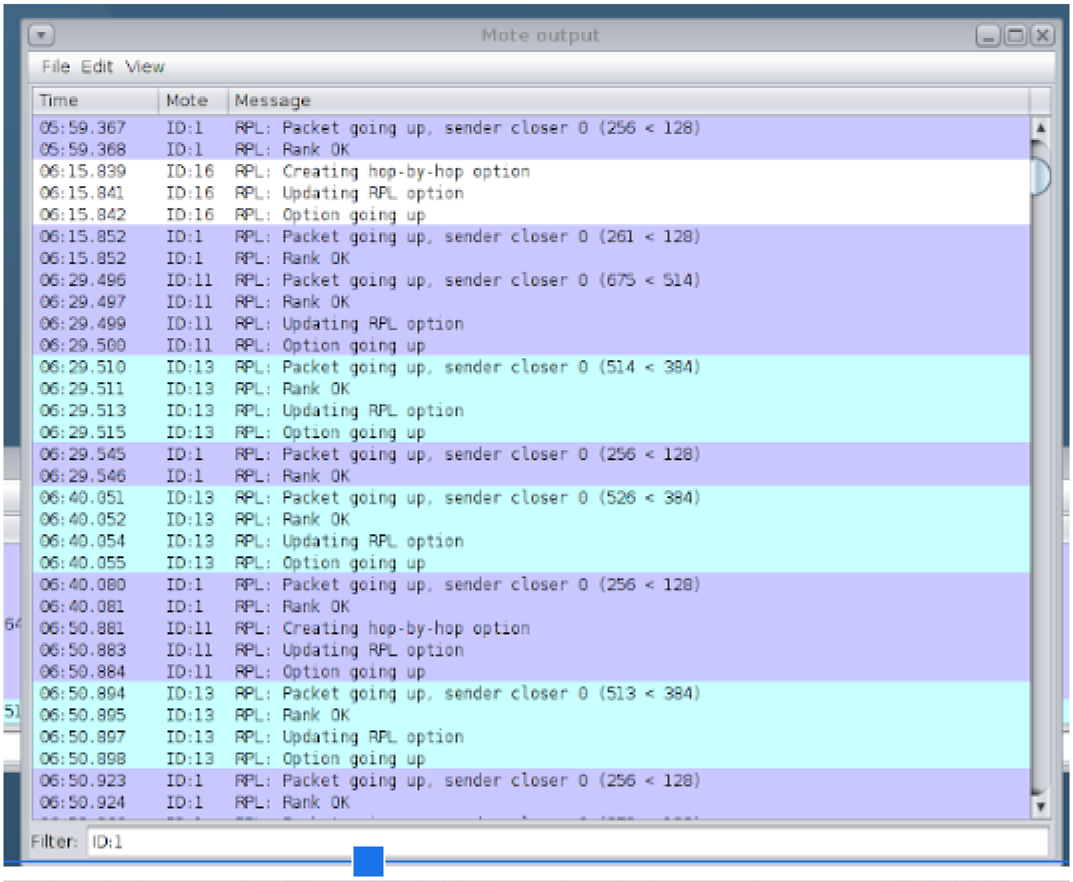
:

We created a network of 11 motes out of which one is an intrusion udp server and others are udp clients. In this particular scenario , the intrusion node tries to attack the parent by transmitting packets that have both the O and R flag set. When this happens , the parent node drops the packet which leads to trickle timer being reset. Due to this , control messages are broad casted more commonly which causes loss of packets .All this can be observed in the packets captured

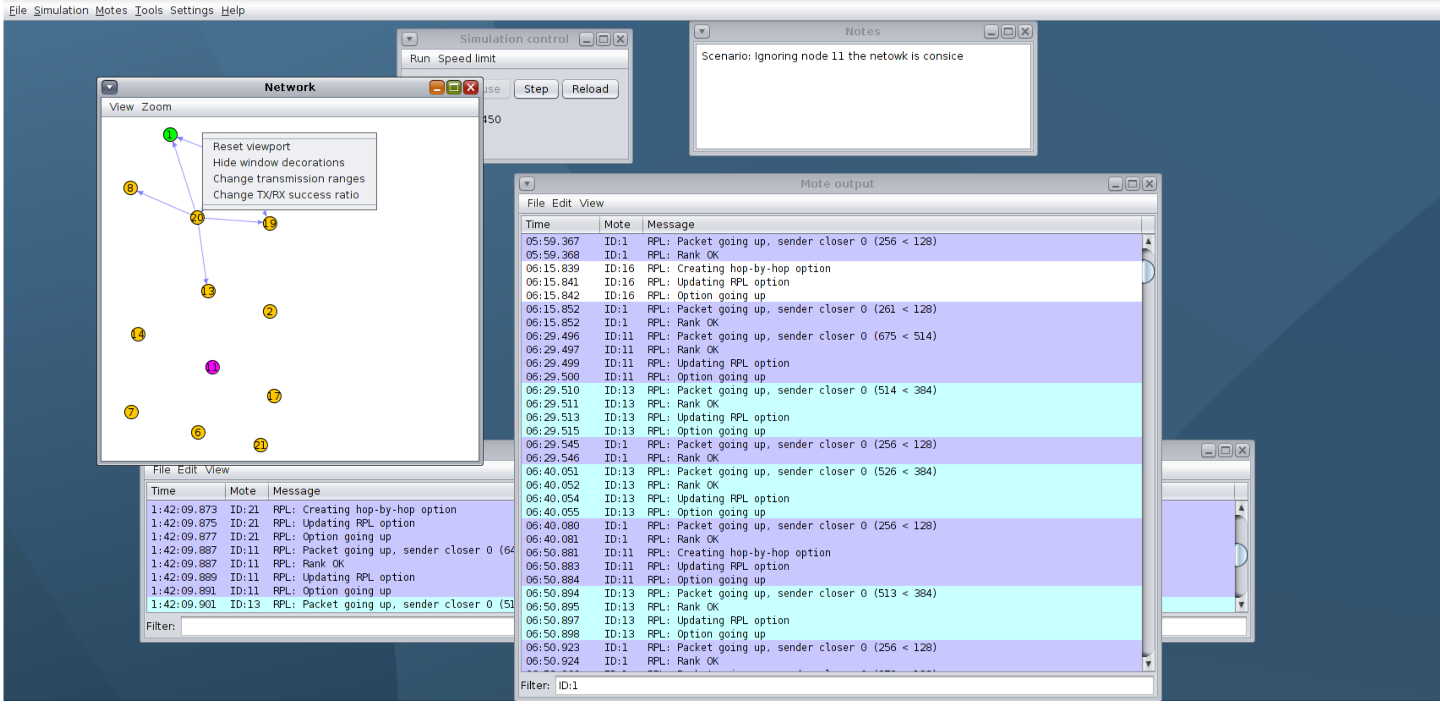
**Blue node indicates the sink node and 1 is the intrusion node**



**Packets captured are shown below**



```
Time        Mote   Message
05:59.367   ID:1   RPL: Packet going up, sender closer 0 (256 < 128)
05:59.368   ID:1   RPL: Rank OK
06:15.839   ID:16  RPL: Creating hop-by-hop option
06:15.841   ID:16  RPL: Updating RPL option
06:15.842   ID:16  RPL: Option going up
06:15.852   ID:1   RPL: Packet going up, sender closer 0 (261 < 128)
06:15.852   ID:1   RPL: Rank OK
06:29.496   ID:11  RPL: Packet going up, sender closer 0 (675 < 514)
06:29.497   ID:11  RPL: Rank OK
06:29.499   ID:11  RPL: Updating RPL option
06:29.500   ID:11  RPL: Option going up
06:29.510   ID:13  RPL: Packet going up, sender closer 0 (514 < 384)
06:29.511   ID:13  RPL: Rank OK
06:29.513   ID:13  RPL: Updating RPL option
06:29.515   ID:13  RPL: Option going up
06:29.545   ID:1   RPL: Packet going up, sender closer 0 (256 < 128)
06:29.546   ID:1   RPL: Rank OK
06:40.051   ID:13  RPL: Packet going up, sender closer 0 (526 < 384)
06:40.052   ID:13  RPL: Rank OK
06:40.054   ID:13  RPL: Updating RPL option
06:40.055   ID:13  RPL: Option going up
06:40.080   ID:1   RPL: Packet going up, sender closer 0 (256 < 128)
06:40.081   ID:1   RPL: Rank OK
06:50.881   ID:11  RPL: Creating hop-by-hop option
06:50.883   ID:11  RPL: Updating RPL option
06:50.884   ID:11  RPL: Option going up
06:50.894   ID:13  RPL: Packet going up, sender closer 0 (513 < 384)
06:50.895   ID:13  RPL: Rank OK
06:50.897   ID:13  RPL: Updating RPL option
06:50.898   ID:13  RPL: Option going up
06:50.923   ID:1   RPL: Packet going up, sender closer 0 (256 < 128)
06:50.924   ID:1   RPL: Rank OK
Filter: ID:1
```



Scenario: Ignoring node 11 the netowk is consice

```
Time          Mote   Message
1:42:09.873   ID:21  RPL: Creating hop-by-hop option
1:42:09.875   ID:21  RPL: Updating RPL option
1:42:09.877   ID:21  RPL: Option going up
1:42:09.887   ID:11  RPL: Packet going up, sender closer 0 (64
1:42:09.887   ID:11  RPL: Rank OK
1:42:09.889   ID:11  RPL: Updating RPL option
1:42:09.891   ID:11  RPL: Option going up
1:42:09.901   ID:13  RPL: Packet going up, sender closer 0 (51
Filter:
```

```
Time        Mote   Message
05:59.367   ID:1   RPL: Packet going up, sender closer 0 (256 < 128)
05:59.368   ID:1   RPL: Rank OK
06:15.839   ID:16  RPL: Creating hop-by-hop option
06:15.841   ID:16  RPL: Updating RPL option
06:15.842   ID:16  RPL: Option going up
06:15.852   ID:1   RPL: Packet going up, sender closer 0 (261 < 128)
06:15.852   ID:1   RPL: Rank OK
06:29.496   ID:11  RPL: Packet going up, sender closer 0 (675 < 514)
06:29.497   ID:11  RPL: Rank OK
06:29.499   ID:11  RPL: Updating RPL option
06:29.500   ID:11  RPL: Option going up
06:29.510   ID:13  RPL: Packet going up, sender closer 0 (514 < 384)
06:29.511   ID:13  RPL: Rank OK
06:29.513   ID:13  RPL: Updating RPL option
06:29.515   ID:13  RPL: Option going up
06:29.545   ID:1   RPL: Packet going up, sender closer 0 (256 < 128)
06:29.546   ID:1   RPL: Rank OK
06:40.051   ID:13  RPL: Packet going up, sender closer 0 (526 < 384)
06:40.052   ID:13  RPL: Rank OK
06:40.054   ID:13  RPL: Updating RPL option
06:40.055   ID:13  RPL: Option going up
06:40.080   ID:1   RPL: Packet going up, sender closer 0 (256 < 128)
06:40.081   ID:1   RPL: Rank OK
06:50.881   ID:11  RPL: Creating hop-by-hop option
06:50.883   ID:11  RPL: Updating RPL option
06:50.884   ID:11  RPL: Option going up
06:50.894   ID:13  RPL: Packet going up, sender closer 0 (513 < 384)
06:50.895   ID:13  RPL: Rank OK
06:50.897   ID:13  RPL: Updating RPL option
06:50.898   ID:13  RPL: Option going up
06:50.923   ID:1   RPL: Packet going up, sender closer 0 (256 < 128)
06:50.924   ID:1   RPL: Rank OK
Filter: ID:1
```

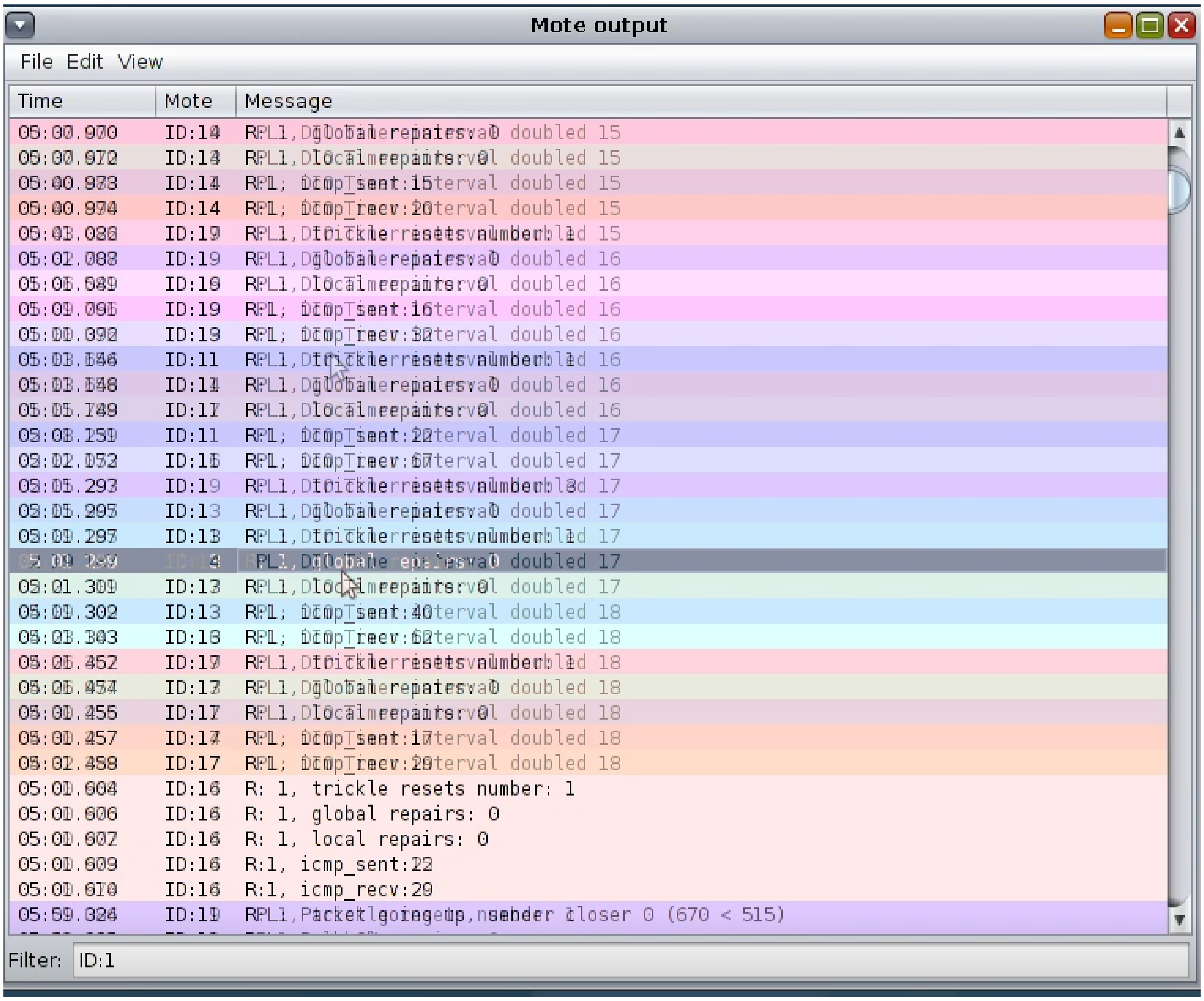
**DEFENCE:**

Coming to mitigation we had used a algorithm where whenever we are receiving the packet with inconsistency with o set and R set , the packet is dropped ,but the trickle timer is not reset . But we set up a threshold to prevent the trickle timer from resetting.

```
if (O = 1 and ri < rj ) or (O = 0 and ri > rj ) then if R = 1
then count + +
drop(packet)
 if count <  threshold then
      Reset timer
```

In future,

We can use a controller to verify the nodes for a lot of attacks. One scenario is that , the intrusion node is detected using kMeans algorithm.The node network  is organised in such a way that the intrusion nodes donot have child nodes.



```
Time         Mote    Message
05:00.970    ID:14   R: 1, global repairs: 0
05:00.972    ID:14   R: 1, local repairs: 0
05:00.973    ID:14   R:1, icmp_sent:15
05:00.974    ID:14   R:1, icmp_recv:20
05:01.086    ID:19   R: 1, trickle resets number: 1
05:01.088    ID:19   R: 1, global repairs: 0
05:01.089    ID:19   R: 1, local repairs: 0
05:01.091    ID:19   R:1, icmp_sent:16
05:01.092    ID:19   R:1, icmp_recv:32
05:01.146    ID:11   R: 1, trickle resets number: 1
05:01.148    ID:11   R: 1, global repairs: 0
05:01.149    ID:11   R: 1, local repairs: 0
05:01.151    ID:11   R:1, icmp_sent:22
05:01.152    ID:15   R:1, icmp_recv:67
05:01.293    ID:13   R: 1, trickle resets number: 3
05:01.295    ID:13   R: 1, global repairs: 0
05:01.297    ID:13   R: 1, trickle resets number: 1
05:01.299    ID:14   R: 1, global repairs: 0
05:01.301    ID:13   R: 1, local repairs: 0
05:01.302    ID:13   R:1, icmp_sent:40
05:01.303    ID:13   R:1, icmp_recv:62
05:01.452    ID:17   R: 1, trickle resets number: 1
05:01.454    ID:17   R: 1, global repairs: 0
05:01.455    ID:17   R: 1, local repairs: 0
05:01.457    ID:17   R:1, icmp_sent:17
05:01.458    ID:17   R:1, icmp_recv:29
05:01.604    ID:16   R: 1, trickle resets number: 1
05:01.606    ID:16   R: 1, global repairs: 0
05:01.607    ID:16   R: 1, local repairs: 0
05:01.609    ID:16   R:1, icmp_sent:22
05:01.610    ID:16   R:1, icmp_recv:29
05:59.324    ID:19   RPL1, Packet going up, sender closer 0 (670 < 515)
```

Filter: ID:1

## Challenges and Interesting aspects

We had a interesting experience when searching for the attacks. The fit iot lab provides an enriched way of accessing various parameters. Using instant contiki we had a lot of dependency issues when running programs . What we find interesting is that the references in paper are easily available but the actual baseline implementations are very scarce from our perspective.The topic related to rpl routing is vey interesting.

## References:

[1] https://github.com/iot-lab/iot-lab/wiki/Control-Node-Sniffer

[2] https://iot-lab.github.io/docs/tools/radio-monitoring/

[3] https://www.iot-lab.info/legacy/dev-center/index.html

[4] https://www.iot-lab.info/legacy/tutorials/contiki-public-ipv6-m3/index.html

[5] https://www.iot-lab.info/legacy/tutorials/contiki-coap-m3/index.html

[6] https://www.iot-lab.info/legacy/tutorials/contiki-private-ipv6-m3/index.html

[7] https://anrg.usc.edu/contiki/index.php/Network_Stack

[8] Solapure, Sharwari & Kenchannavar, Harish. (2019). RPL And COAP Protocols, Experimental Analysis for IOT: A Case Study. International Journal of Ad hoc, Sensor & Ubiquitous Computing. 10. 01-15. 10.5121/ijasuc.2019.10201.

[9] https://github.com/contiki-ng/contiki-ng

[10] https://github.com/contiki-ng/contiki-ng/tree/develop/test

[11] RPL_DODAG_Visualization_v13.1/Documentation at main · NetSim-TETCOS/RPL_DODAG_Visualiz

[12] Softwarized & Wireless Networks Research Group (github.com)

: