ANALYSIS OF RPL ATTACKS USING COOJA SIMULATOR

Professional Practice/Seminar (IT890) Project Report Submitted in partial fulfilment of the requirements for the degree of

MASTER OF TECHNOLOGY
in

INFORMATION TECHNOLOGY
by

BAVYA BALAKRISHNAN (192402IT003)

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025
APRIL, 2020

D E C L A R A T IO N

I hereby *declare* that the *Professional Practice/Seminar (IT890) Project Work Report* of the M.Tech.(IT) entitled .....

.....

.....

which is being submitted to the National Institute of Technology Karnataka Surathkal, in partial fulfilment of the requirements for the award of the Degree of Master of Technology in the department of Information Technology, is a ***bonafide report of the work carried out by me.*** The material contained in this project report has not been submitted to any University or Institution for the award of any degree.

(BAVYA BALAKRISHNAN - 192402IT003) Signature of the Student

Department of Information Technology

Place : NITK, SURATHKAL
Date :

C E R T I F I C A T E

This is to certify that the Professional Practice/Seminar (IT890) Project Work Report entitled

.....

.....

submitted by .....,

(Register Number: .....) as the record of the work carried out by him/her, is accepted as the Professional Practice/Seminar (IT890) Project Work Report submission in partial fulfilment of the requirements for the award of degree of Master of Technology in the Department of Information Technology.

w

Mrs. Tanmayee
Signature of the Examiner with Date

Dr. Jaidhar C D
Signature of the Guide with Date

ABSTRACT

RPL is a routing protocol designed for low power and lossy networks which contains devices constrained in memory, energy and processing power. Such devices will operate at low data rate and experience relatively significant packet losses. In addition to that due to the open environment in which they operate devices are more susceptible to several security attacks. In this paper we analyse the various attacks on RPL protocol using simulation in contiki platform. We are mainly concentrating on the attacks at the network layer. Main objective is to test and show the effect of some chosen attacks.

Index Terms—IoT, 6LoWPAN, RoLL, COOJA, RPL, DODAG & WSN

TABLE OF CONTENT

1. Introduction

To monitor the physical and environmental conditions in various application needs sensor devices have been used very actively. Recent advances in this area is towards connecting things embedded with sensors to internet. This devices which are usually better operated are employed to monitor, measure and report data. To meet the specific requirements of this kinds of networks new technologies and protocols are coming into picture. For example, 6LowPAN[3] to enable this devices with small bandwidth, low power and limited processing capabilities to use Internet protocols. Its main features include mechanisms such as header compression and encapsulation that allows the packets from IPv6 to be sent and received over IEEE 802.15.4 based networks. This IEEE 802.15.4 standard[4] is capable of handling contained devices by specifying several physical layer options and medium access control sub-layer.

RPL(Routing Protocol for Low-Power and Lossy Networks) or ripple is one such routing protocol specifically designed for Low power and Lossy networks with low power consumption and generally susceptible to packet loss. In this paper we are concentrating on the security side of RPL. We analyse various attacks such as Packet dropping attack, Flooding, Version number modification and Decreased rank.

The remainder of this document is structured as follows. An overview of RPL is given first followed by a brief description of various attacks is given in session Security threats on RPL. How the experiments are conducted to test the attacks are explained in Methodology session. The results and observations in Experimental Results and Analysis. Then at the end Conclusion.

2. Overview of RPL

RPL organizes a topology as a destination oriented DAG (called DODAG) to route packets. A RPL Instance is a set of one or more DODAGs that share a RPLInstanceID. A network may run multiple instances of RPL concurrently. Each such instance may have different performance criteria. RPL uses objective function to form routes. The objective function make use of the set of metrics/constraints imposed by the network. So RPL builds DODAG as a

logical routing topology over the physical network based on the objective function specified. A node can be part of one or more DODAGS with different set of requirements.

2.1 RPL operation

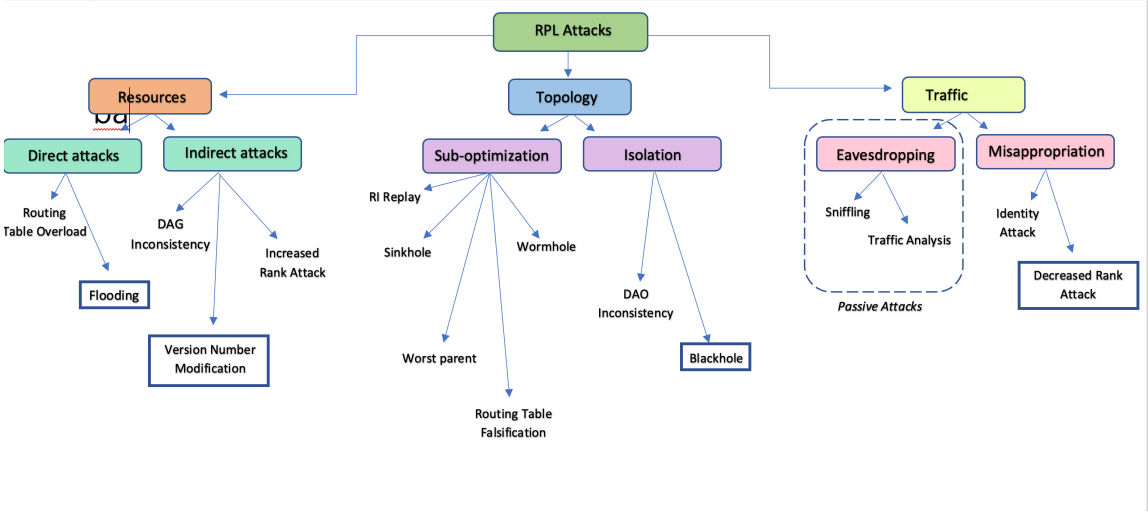
The process of constructing the route begins when a node designated as root broadcast its DODAG information object (DIO) message .The DODAG Information Object carries information that allows a node to discover a RPL Instance, learn its configuration parameters, select a DODAG parent set, and maintain the DODAG. DIO consists of RPLInstanceID, Rank and DODAGID. integer set by a DODAG root that uniquely identifies a DODAG. RPLInstanceID is an 8-bit field set by the DODAG root that indicates of which RPL Instance the DODAG is a part. A node's rank represents the distance to the DODAG root with respect to a given metric. The neighbours of the root node (nodes that can directly reach the root) set their rank to 1, add root address to update their parent and continues broadcasting their own DIO. Every node when it receives multiple DIO Message choose the parent offering better metric to make it as the default next hope towards root. Whenever a node has to join the network, can either wait for a DIO message or send a DODAG Information Solicitation message (DIS) to ask others to send a DIO if it fails to receive a DIO within some stipulated time. After receiving the DIO, it chooses its preferred parent and builds a Destination Advertisement Object (DAO) message, containing the information about its address and prefix parent.

[2] also explains the Global Repair and Local Repair mechanisms provided by RPL to fix the broken link. Global repair is started by DODAG root re-estimate the whole topology by sending a new sequence number. After receiving the new DIO messages, the nodes restart the selection of parents and reupdates the link cost. Local repair is used to save time. To get the new topology, the node sends a DIS message for joining the network.

3. Security threats on RPL

3.1 Taxonomy

This session explains the attacks we test. Given below is the taxonomy of RPL attacks. The attacks in blue frames are the ones we chose to explore.



The first category of attacks targets the exhaustion of network resources such as, memory, energy and power. This can be done by forcing the legitimate nodes to perform unnecessary actions to increase the use of their resources. This may impact on the availability of the network by congesting available links or by incapacitating nodes and may therefore impact on the lifetime of the network. This category can be further subdivided in two sub-categories ; direct attacks, in which the malicious node directly generates the overload disturbing the network either by flooding attacks or routing table overload attacks, and indirect attacks, in which the malicious node provokes the other nodes to make them generate the overload. This category can again be divided into direct attacks and indirect attacks. Malicious node directly produce the overload to imbalance the network in first category but it indirectly overloads the target using intermediate node in second category.

The second category holds the attacks targeting the RPL network topology. The goal of these attacks is to disturb the normal operation of the network. These could then cause the isolation of one or more nodes. This category can also be subdivided in two sub-categories ; sub-optimisation, meaning that the network will converge to a non-optimal form, inducing poor

performance, and isolation of a node or a subset of nodes, and there is a possibility that the nodes are unable to communicate with their parents or with the root.

The third category covers attacks against the network traffic. These attacks are aimed to make a malicious node introduce itself inside the network, not disturbing it's working. This leads to information leakage by eavesdropping the traffic or impersonating legitimate nodes. This category is again subdivided in two sub-categories ; eavesdropping (passively) the information that is forwarded through the network or misappropriation of a node or a set of nodes, namely for tampering the legitimate exchanged information. Security techniques from other networks cannot be
applied straight away because of the specific constraints of RPL.

3.2. Implemented Attacks

We chose some sample attacks from different categories to test on RPL .

Flooding:

This is a direct attack concerning the exhaustion of network resources. We generate overload through DIS traffic. It cause the nodes in the neighbourhood of it to reply with DIO messages advertising information about DODAG's to new nodes and reset their trickle timers which is supposed to increase as the network stabilizes.

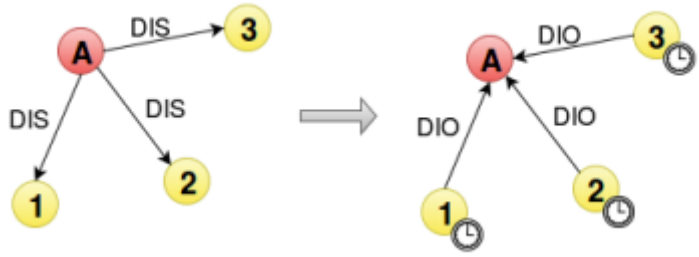


Figure 3. 1 Flooding attack: A is the Attacker ; 1,2,3 are legitimate nodes

Version Number Modification

This is an indirect attack causing the exhaustion of resources. When global repair is required DODAG's root increase the version number. So attacking is by Version Number Modification cause unnecessary graph rebuildings. Indeed, as the root receives the DIO with an invalid version number, it updates it and resets its trickle timer (as depicted by the timer in the figure below) for resending a new DIO. By contrast, normal nodes initiate a global repair

(as depicted by the wrench), that is, they remove their parents and use the received DIO to update their new parent.

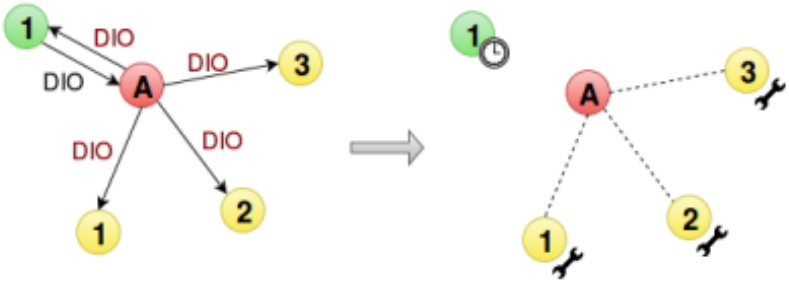


Figure 3. 2 Versioning attack : DIO in black is legitimate; red one has version increased by 1

Decreased Rank

It is implemented by advertising a lower rank to make the legitimate nodes connect to the DODAG via the attacker. This can be a basis for sinkhole, blackhole or also eavesdropping attacks.

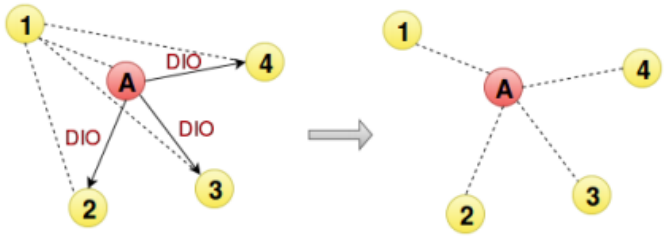


Figure 3. 3 Decreased rank attack: DIO sent contains a better rank

Blackhole

Instead of forwarding the packets malicious node drops them. This attack can be seen as a denial-of-service attack. If the position of the node is well chosen, it can isolate several nodes from the network. The selective forwarding attack (grey hole) is a variant of this type of attack. This attack has as consequence to disturb routing paths, it can be used to filter any protocol.

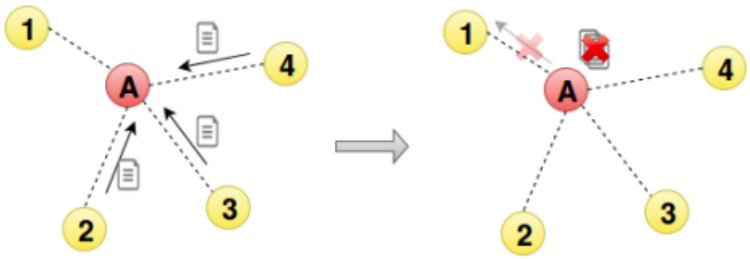


Figure 3. 4 Blackhole attack (data received from legitimate nodes is dropped)

A solution to deal with this attack is to create disjoint paths between the source and the destination nodes but it is difficult to create this for the entire network. Another solution is to dynamically select the path to parents/children every time e transmission occur. There are different indicators to detect these attacks, such as rate and frequency of DIO messages, packet delivery ratio, loss percentage and delay. It is generally difficult to defend against all selective forwarding attacks.

4. Methodology

Some features of the Contiki OS and Contiki RPL have been exploited to monitor the malicious behaviour of nodes in the network. The Cooja simulator is used to observe the network across different scenarios discussed in the paper. The network consists of 10 nodes total. Out of which node 1 is sink node and others are sender nodes. One among sender nodes is chosen as malicious node to launch attack. We used sky mote for implementation. For analysing the parameters such as packet delay and the rate of control messages (DIO messages), wireshark is used. For all the attacks, we tested the network with and without malicious nodes.

Flooding

The default values for control messages are declared and defined in rpl-private.h. Set both this parameters to zero. These are the two ContikiRPL configuration constants defined to immediately start sending DIS at a sustained rate and thereby helping the malicious node to launch DIS attack

#define RPL_DIS_INTERVAL 0

#define RPL_DIS_START_DELAY 0

Also in rpl-timers.c replace "next_dis++;" with "next_dis++; int i=0; while (i<10) {i++; dis_output(NULL);

Version Number modification

In rpl-icmp6.c replace dag->version with dag->version++
With its modified RPL file, the malicious node increases the version number before forwarding received DIO messages in the function dio_output, thus triggering unnecessary global repairs.

Decreased Rank

min_hoprankinc (in contiki) or MinHopRankIncrease (in RPL RFC) is the minimum increase in Rank between a node and any of its DODAG parents. The root is responsible on provisioning the MinHopRankIncrease value. So, all nodes must use this value (same for all) if they want to calculate their rank via their parents.

RPL_CONF_MIN_HOPRANKINC is set to 0 in udp_sender.c of attacker node.
We can change the RPL_MAX_RANKINC in rpl-private.h to limit the number of levels of children in the network. i.e. leaf nodes will not forward any DIOs. Also the INFINITE_RANK is reduced to 256 from 0xffff. The local repair is instigated after periodic "rpl_recalculate_ranks" indicating rank of that parent is not acceptable anymore. We should remove this line from rpl-timers.c

Blackhole attack

Malicious node execute selective forwarding by dropping DAO messages. To launch that below lines in rpl-icmp6.c are commented
1.uip_icmp6_send(rpl_get_parent_ipaddr(dag->preferred_parent),ICMP6_RPL,
RPL_CODE_DAO, buffer_length);
2.goto fwd_dao
The DAO are not forwarded to parent and it cause instability in topology causing nodes to send more DIO messages.

5. Experimental Results and Analysis

All the attacks mainly affect the availability of all or part of the WSN. Moreover, the Version Number modification and Blackhole attacks affect the integrity.

Flooding

The malicious node immediately starts sending DIS messages to its neighbours, then triggering DIO messages and trickle timers reset. There was no change in DAG. But important energy exhaustion is observed. We observe that malicious node 2 impact its nearby nodes 3, 6 and 9. We can demonstrate the attack efficiency using this information to compare the power consumption in the simulation without and with the malicious node. We observe that nodes are particularly impacted by the attack in terms of ON and RX times. However, these nodes are not impacted in term of TX time. The reason is that upon the reception of a DIS, the nodes reset their trickle timers but do not immediately send a DIO due to multicast DIS. Only in case of unicast DIS the receiver immediately reply with DIO.

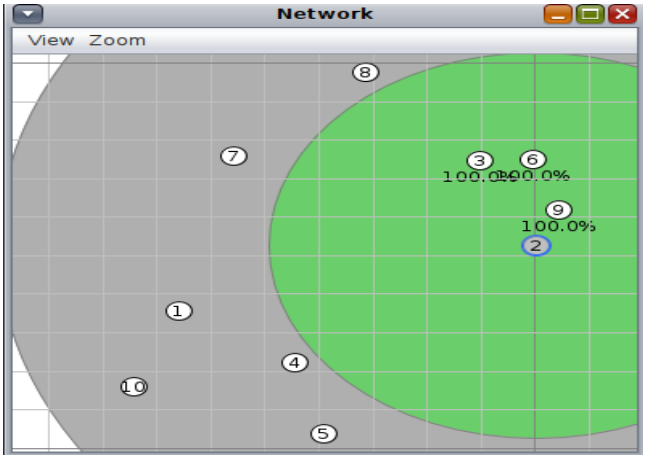


Figure 5. 1 Network used for Flooding (2: Attacker)

When the packets were analysed using Wireshark, we found around 1% of total packets were DIS messages generated from malicious node 2 to disturb the network and every other node generated 0.1% DIS messages.

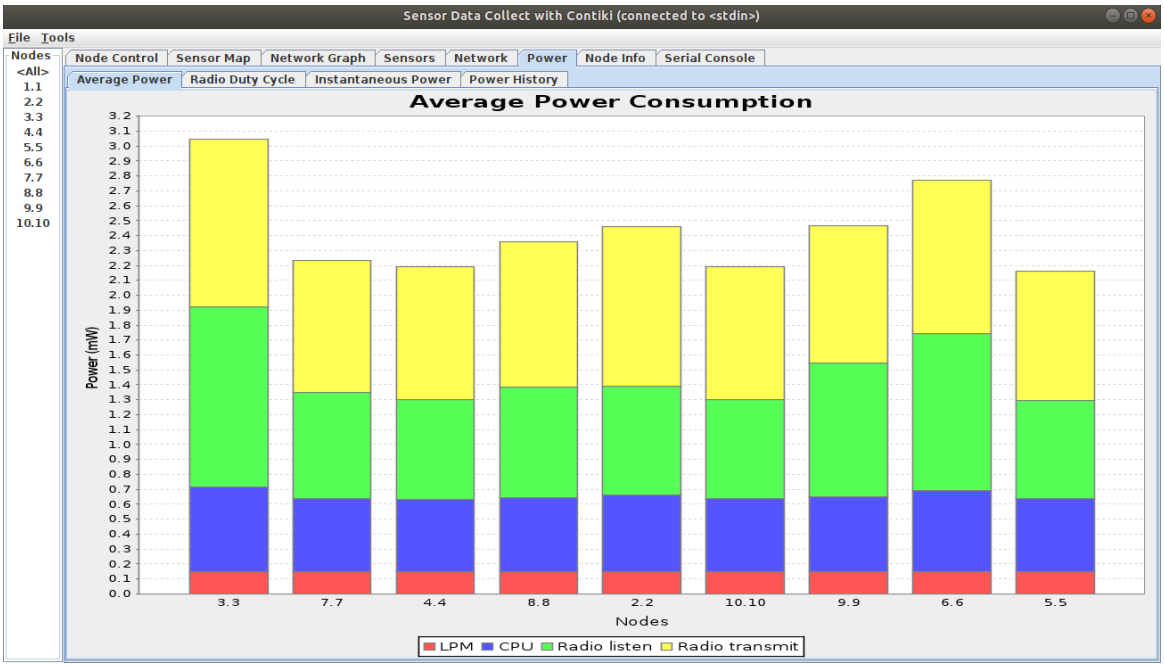


Figure 5. 2 Figure 5. 2 Powertrace per mote: with Flooding

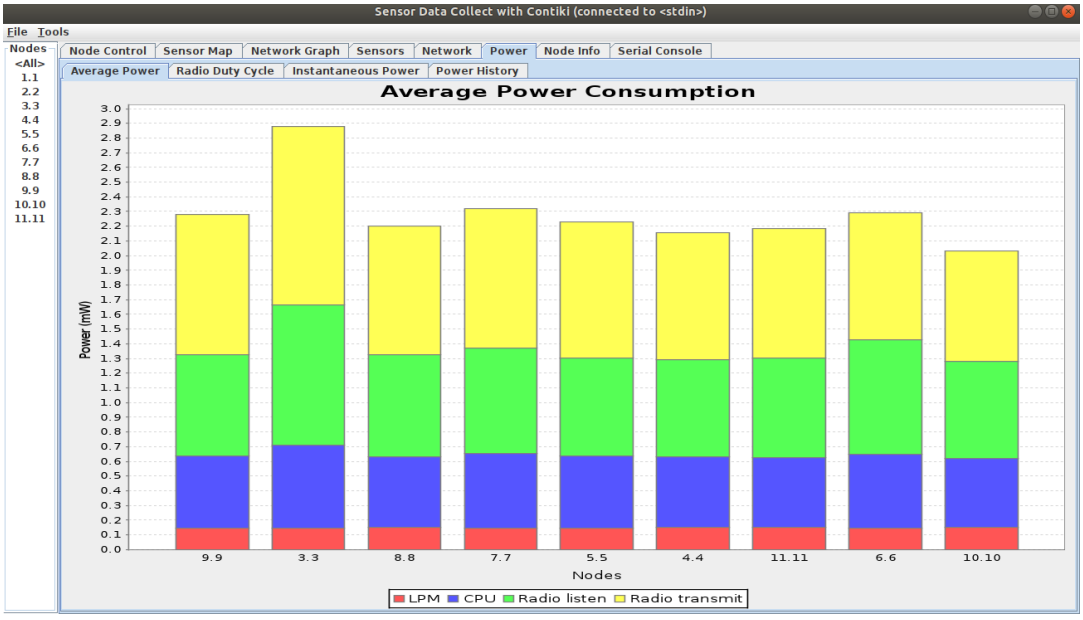
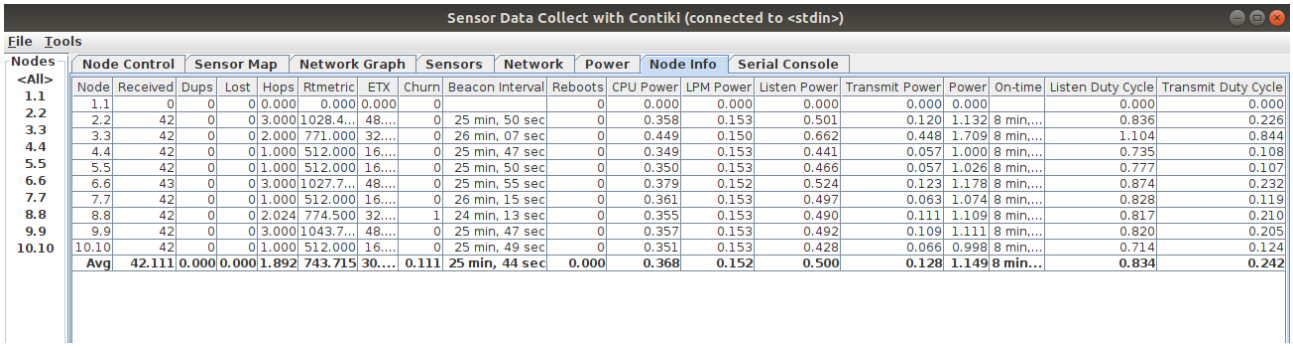


Figure 5. 3 Powertrace per mote: without Flooding



Node	Received	Dups	Lost	Hops	Rtmetric	ETX	Churn	Beacon Interval	Reboots	CPU Power	LPM Power	Listen Power	Transmit Power	Power	On-time	Listen Duty Cycle	Transmit Duty Cycle
1.1	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000	0.000	0.000		0.000	0.000
2.2	42	0	0	3.000	1028.4...	48....	0	25 min, 50 sec	0	0.358	0.153	0.501	0.120	1.132	8 min,...	0.836	0.226
3.3	42	0	0	2.000	771.000	32....	0	26 min, 07 sec	0	0.449	0.150	0.662	0.448	1.709	8 min,...	1.104	0.844
4.4	42	0	0	1.000	512.000	16....	0	25 min, 47 sec	0	0.349	0.153	0.441	0.057	1.000	8 min,...	0.735	0.108
5.5	42	0	0	1.000	512.000	16....	0	25 min, 50 sec	0	0.350	0.153	0.466	0.057	1.026	8 min,...	0.777	0.107
6.6	43	0	0	3.000	1027.7...	48....	0	25 min, 55 sec	0	0.379	0.152	0.524	0.123	1.178	8 min,...	0.874	0.232
7.7	42	0	0	1.000	512.000	16....	0	26 min, 15 sec	0	0.361	0.153	0.497	0.063	1.074	8 min,...	0.828	0.119
8.8	42	0	0	2.024	774.500	32....	1	24 min, 13 sec	0	0.355	0.153	0.490	0.111	1.109	8 min,...	0.817	0.210
9.9	42	0	0	3.000	1043.7...	48....	0	25 min, 47 sec	0	0.357	0.153	0.492	0.109	1.111	8 min,...	0.820	0.205
10.10	42	0	0	1.000	512.000	16....	0	25 min, 49 sec	0	0.351	0.153	0.428	0.066	0.998	8 min,...	0.714	0.124
Avg	**42.111**	**0.000**	**0.000**	**1.892**	**743.715**	**30....**	**0.111**	**25 min, 44 sec**	**0.000**	**0.368**	**0.152**	**0.500**	**0.128**	**1.149**	**8 min...**	**0.834**	**0.242**

Figure 5. 4 Flooding attack: Node info

Version Number modification

Since root handles this there is no change in DAG. But important energy exhaustion is observed. Significant power increase in all nodes (above 1.50mW). Attacker and its neighbours have power above 4mW.

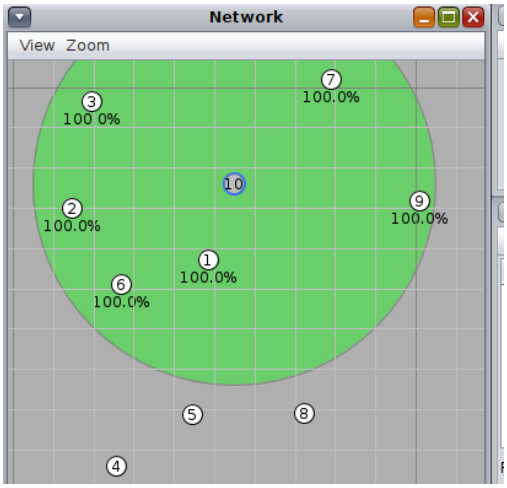


Figure 5. 5 Network used for Version Number modification attack(10:Attacker)

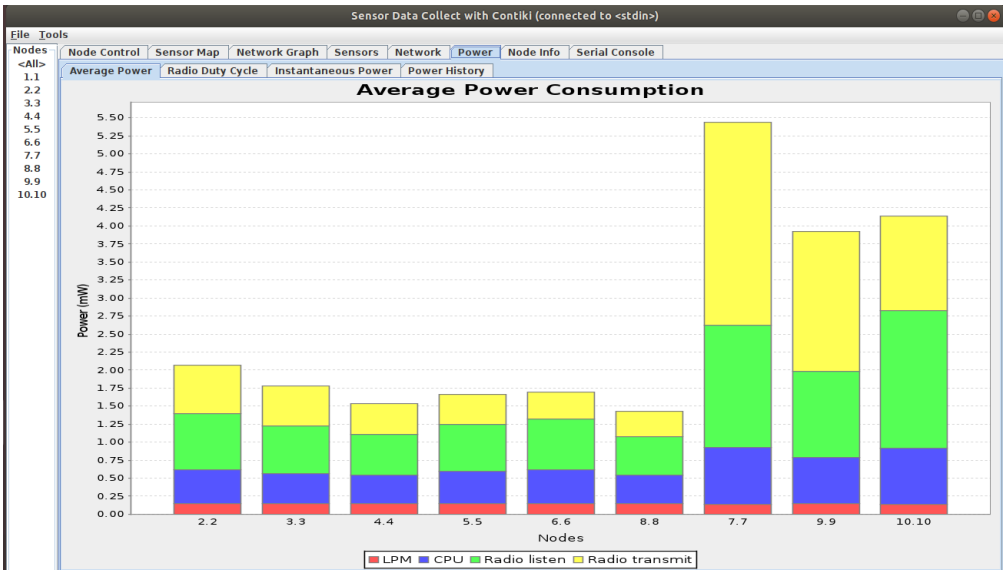


Figure 5. 6 Powertrace per mote: With version modification attack

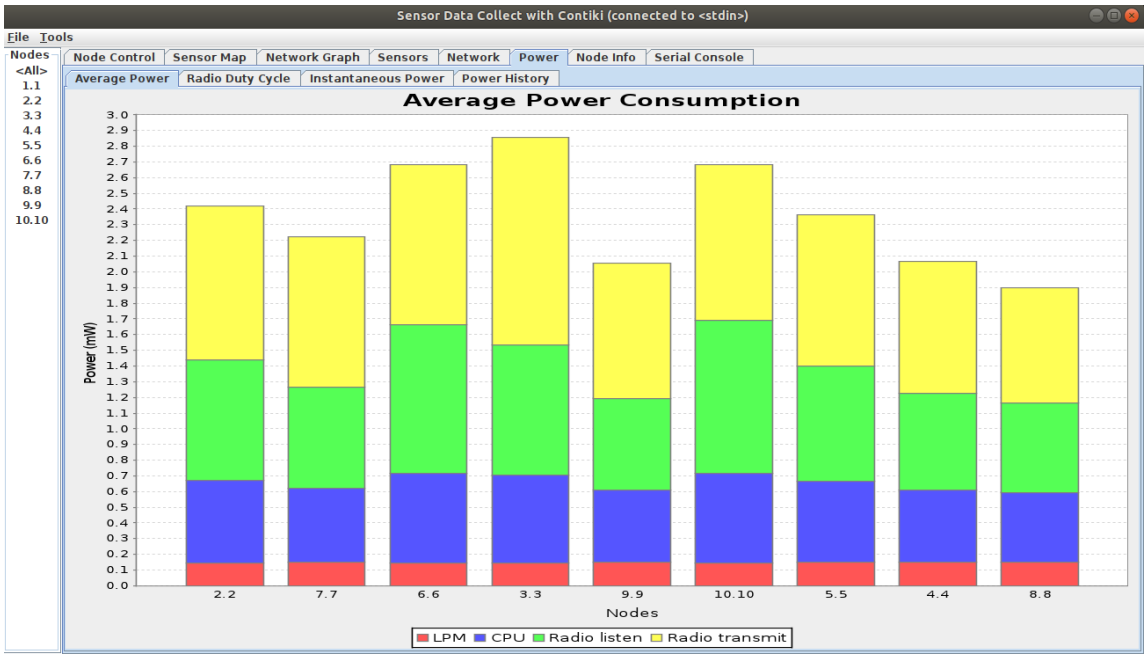
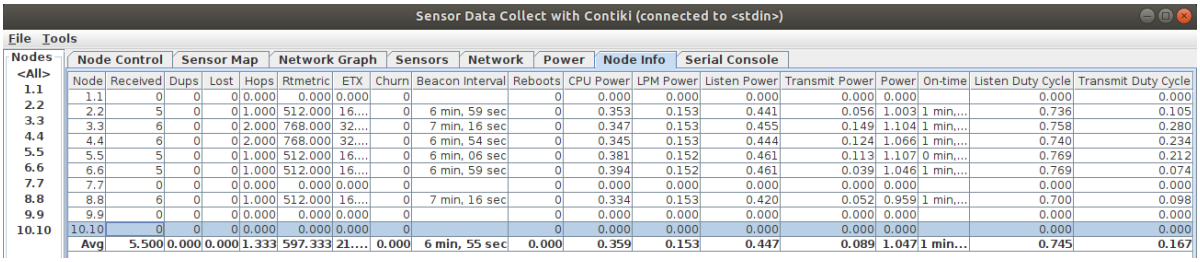


Figure 5. 7 Powertrace per mote: Without version modification attack



Node	Received	Dups	Lost	Hops	Rtmetric	ETX	Churn	Beacon Interval	Reboots	CPU Power	LPM Power	Listen Power	Transmit Power	Power	On-time	Listen Duty Cycle	Transmit Duty Cycle
1.1	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000	0.000	0.000		0.000	0.000
2.2	5	0	0	1.000	512.000	16....	0	6 min, 59 sec	0	0.353	0.153	0.441	0.056	1.003	1 min,...	0.736	0.105
3.3	6	0	0	2.000	768.000	32....	0	7 min, 16 sec	0	0.347	0.153	0.455	0.149	1.104	1 min,...	0.758	0.280
4.4	6	0	0	2.000	768.000	32....	0	6 min, 54 sec	0	0.345	0.153	0.444	0.124	1.066	1 min,...	0.740	0.234
5.5	5	0	0	1.000	512.000	16....	0	6 min, 06 sec	0	0.381	0.152	0.461	0.113	1.107	0 min,...	0.769	0.212
6.6	5	0	0	1.000	512.000	16....	0	6 min, 59 sec	0	0.394	0.152	0.461	0.039	1.046	1 min,...	0.769	0.074
7.7	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000	0.000	0.000		0.000	0.000
8.8	6	0	0	1.000	512.000	16....	0	7 min, 16 sec	0	0.334	0.153	0.420	0.052	0.959	1 min,...	0.700	0.098
9.9	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000	0.000	0.000		0.000	0.000
10.10	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000	0.000	0.000		0.000	0.000
Avg	**5.500**	**0.000**	**0.000**	**1.333**	**597.333**	**21....**	**0.000**	**6 min, 55 sec**	**0.000**	**0.359**	**0.153**	**0.447**	**0.089**	**1.047**	**1 min...**	**0.745**	**0.167**

Figure 5. 8 Version modification attack: Node info

Decreased Rank

The malicious node will advertise a better rank than neighbours, causing the DAG to be modified. This attack does not damage a network, however, combining with other building blocks could be very effective because it allows the attacker to suck all the traffics to him. DAG is changed, legitimate nodes in the neighbourhood of the malicious node have now set it as their parent. Took a lot of time to start data transmission(network stabilization took time). High power consumption by attacker node and its neighbours. Initially only the unaffected nodes send packets.

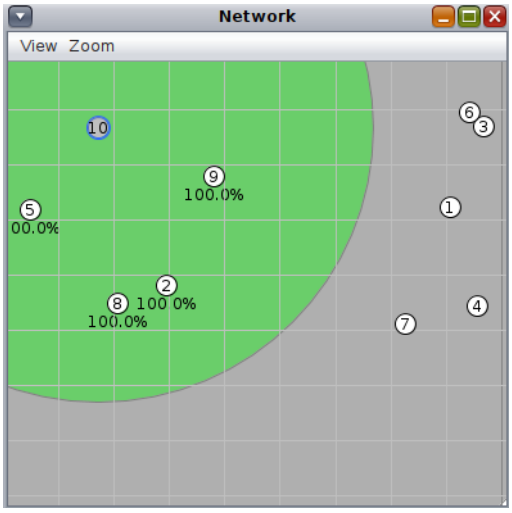


Figure 5. 9 Network used for Decreased Rank attack (10:Attacker)

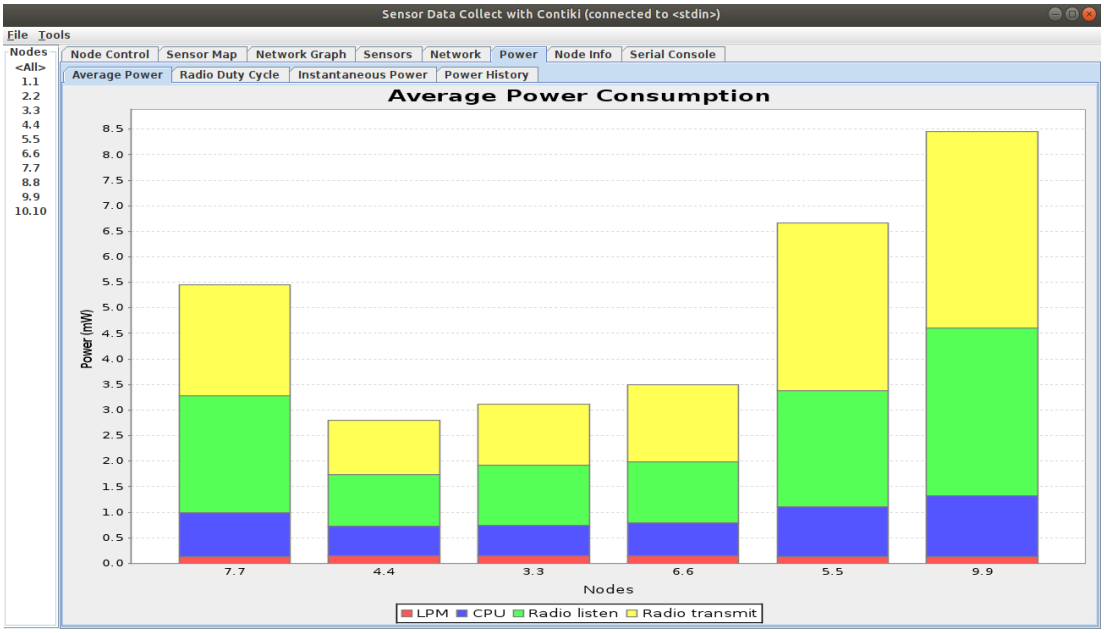
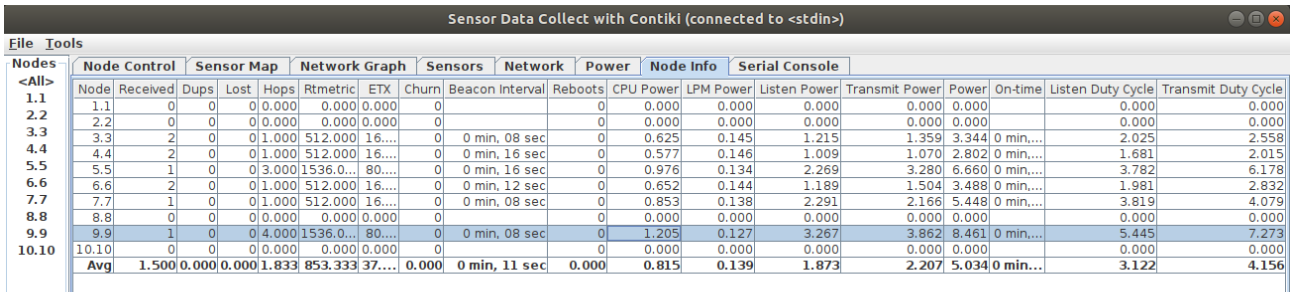


Figure 5. 10 Powertrace per mote: With Decreased rank attack



Node	Received	Dups	Lost	Hops	Rtmetric	ETX	Churn	Beacon Interval	Reboots	CPU Power	LPM Power	Listen Power	Transmit Power	Power	On-time	Listen Duty Cycle	Transmit Duty Cycle
1.1	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000	0.000	0.000		0.000	0.000
2.2	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000	0.000	0.000		0.000	0.000
3.3	2	0	0	1.000	512.000	16....	0	0 min, 08 sec	0	0.625	0.145	1.215	1.359	3.344	0 min,...	2.025	2.558
4.4	2	0	0	1.000	512.000	16....	0	0 min, 16 sec	0	0.577	0.146	1.009	1.070	2.802	0 min,...	1.681	2.015
5.5	1	0	0	3.000	1536.0...	80....	0	0 min, 16 sec	0	0.976	0.134	2.269	3.280	6.660	0 min,...	3.782	6.178
6.6	2	0	0	1.000	512.000	16....	0	0 min, 12 sec	0	0.652	0.144	1.189	1.504	3.488	0 min,...	1.981	2.832
7.7	1	0	0	1.000	512.000	16....	0	0 min, 08 sec	0	0.853	0.138	2.291	2.166	5.448	0 min,...	3.819	4.079
8.8	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000	0.000	0.000		0.000	0.000
9.9	1	0	0	4.000	1536.0...	80....	0	0 min, 08 sec	0	1.205	0.127	3.267	3.862	8.461	0 min,...	5.445	7.273
10.10	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000	0.000	0.000		0.000	0.000
Avg	**1.500**	**0.000**	**0.000**	**1.833**	**853.333**	**37....**	**0.000**	**0 min, 11 sec**	**0.000**	**0.815**	**0.139**	**1.873**	**2.207**	**5.034**	**0 min...**	**3.122**	**4.156**

Figure 5. 11 Decreased rank attack : Node info

Blackhole

The increase in number of DIO message exchanged indicates the unstable topology with respect to routing of packets. DIO packets generated by each nodes establishes the fact that the nodes have the idea of network unstability .

Time duration: 500000ms

Scenario 1: Without blackhole attack

Total no packets: 26933

Total no DIO packets: 23726(88.1%)

Scenario 2: With blackhole attack

Total no packets: 26391

Total no DIO packets: 23830(90.3%)

Node	Scenario 1	Scenario 2
1	2784(10.3%)	2752(10.4%)
2	2687(10%)	2659(10.1%)
3	2499(9.3%)	2441(9.2%)
4	2737(10.2%)	2693(10.2%)
5	0	0
6	2632(9.8%)	2550(9.7%)
7	2490(9.2%)	2678(10.1%)
8	2547(8.5%)	2617(9.9%)
9	2660(9.9%)	2717(10.3%)
10	2438(9.1%)	2507(9.5%)

Table 5. 1 DIO Messages sent across scenarios

We can observe an increase in DIO packets for nodes 1,2,7,8,9,10 . Where 8,9 are the neighbours of malicious node 10.

6. Conclusion

Impact of various attacks uniformly chosen amongst the presented taxonomy on RPL is analysed using Cooja simulator. We found that such attacks affected the resources, topology and integrity of the WSN network. Possible further improvements includes, testing the network with multiple malicious nodes, explore the mitigation strategies for each attack.

7. References

1.Impact of Packet Dropping Attacks on RPL, Arvind Kumar ,Rakesh Matam , Shailendra Shukla

2.T. Winter et. al, RPL: IPv6 Routing protocol for Low power and Lossy

Networks, Internet Draft draft-ietf-roll-rpl-17 Retrieved, June 2015.

3.Geoff Mulligan, The 6LoWPAN architecture, EmNets 2007: Proceedings of the 4th workshop on Embedded networked sensors, ACM, 2007.

4. IEEE Standard. 802.15.4, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low Rate Wireless Personal Area Networks (LR-WPANs), IEEE 2011.

5.Mobile and Embedded Computing: RPL Attacks Framework, Alexandre D'Hondt