EXPERIENCE FROM IMPLEMENTATION OF RPL, ITS CHALLENGES AND POSSIBLE SOLUTION

Raj Ardeshna[1], Rahul Goradia[2]

[1] *Student, Master of Engineering, Electronics and Communication, G.H.Patel College of Engineering and Technology, Gujarat, India*

[2] *Assi. Professor, Electronics and Communication, G.H.Patel College of Engineering and Technology, Gujarat, India*

ABSTRACT

This paper focuses on routing protocol for IPv6 enable wireless sensor network. We studied routing protocol for low power and lossy networks (RPL), a tree-based routing protocol designed for sensor network. We have simulated the RPL in cooja and also implemented it in our hardware platform. We have done a performance study of RPL, found out its limitations and possible solutions also. In our study, we found instability and packet loss in WSN. We gave the solution for this problem by defining a new Objective Function. We have improved the efficiency and stability of RPL.

Keyword: - *RPL, IPv6, WSN, DAG, Routing*

---

1. INTRODUCTION

WIRELESS sensor networks consist of hundreds of distributed sensors connected to each other. Today sensor hold a wide area of usage such as in industry, environmental monitoring, healthcare applications and military applications. The IETF Working Group IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) proposed an RFC [1] to enable IPv6 packets to be carried over IEEE 802.15.4. Eventually, the IETF Working Group Routing over Low power and Lossy networks (ROLL) designed a routing protocol named IPv6 Routing Protocol for Low power and Lossy network (RPL). RPL was proposed because none of the existing known protocols such as AODV, OLSR or OSPF met the specific requirements of Low power and Lossy networks (LLN) [9].

Routing in LLNs is one of the key challenges for the IoT emergence. The constraints of LLNs have a significant impact on the protocol design. Small memory limits the number of stored route entries. Limited energy supply dictates minimal radio usage and optimized control overhead.

Much research work focused on the performance of RPL leading to the common observation that RPL performs well in case of multipoint-to-point traffic, but induces a large overhead in scenarios where point-to-multipoint traffic is non-negligible. We show how Contiki-Rpl implementation behaves in realistic scenarios.

The paper is organized as follow. Section-II describes the basic of RPL and formation of DODAG, Section-III describes the challenges related to implementation of RPL, Section-IV describes the possible solutions for that challenges, Section-V shows the result of the modified RPL and Section-VI conclude the paper.

2. RPL-routing protocol for low power and lossy network

RPL is a Distance Vector protocol that specifies how to construct a Destination Oriented Directed Acyclic Graph (DODAG) with a defined Objective Function and a set of metrics and constraints. RPL uses a proactive approach: it finds and maintains routes without any traffic considerations – route are created even if not used [1].

RPL defines mechanisms for the formation of the topology and repair mechanisms to recover from failure of nodes and routing loops. For this purpose, routers exchange an RPL specific type of ICMPv6 messages which contain the required routing information. Up- and Down-ward routes along the DODAG are created by sending messages in the opposite direction [1].

There are three ICMPv6 messages: 1. DIO (DODAG Information Object) 2. DAO (Destination Advertisement Object) 3. DIS (DODAG Information Solicitation)

Upward paths toward the root are thus created by sending DODAG Information Objects (DIOs) downwards from the root toward the leaves. Downward routes are established by sending Destination Advertisement Objects (DAOs) from leave towards the root. DODAG Information Solicitation (DIS) messages proactively solicit the DODAG related information from neighboring nodes [1].

A root starts the DODAG building process by transmitting a DIO (fig-1(a)). Neighboring nodes process DIOs and make a decision on joining the DODAG based on the objective function and/or local policy. A node computes its Rank with respect to the root and starts advertising DIO messages to its neighbors with the updated information (fig-1(b, c)). As the process converges, each node in the network receives one or more DIO messages and has a preferred parent towards the sink. Hence, RPL optimizes the upward routes for multipoint-to-point traffic that accounts for most of the traffic in LLNs (fig-1(h)).

To support downward routes, RPL uses DAO control messages that give the prefix information, the route lifetime, and other information about the distance of the prefix. RPL RFC defines the storing and non-storing modes. In the non-storing mode, packets use source-routing for downward traffic. In our study, we focus on the storing mode in which each node keeps track of all accessible downlink prefixes.

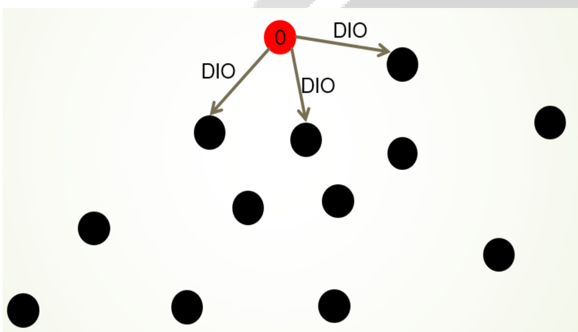


Figure – 1(a)

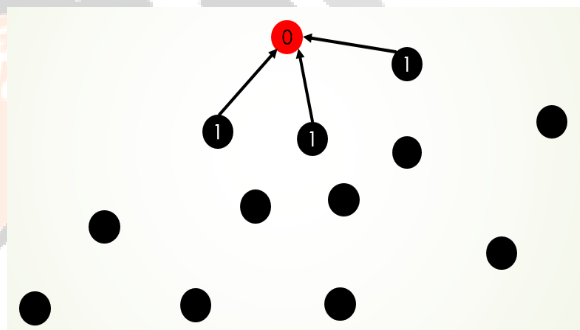


Figure – 1(b)

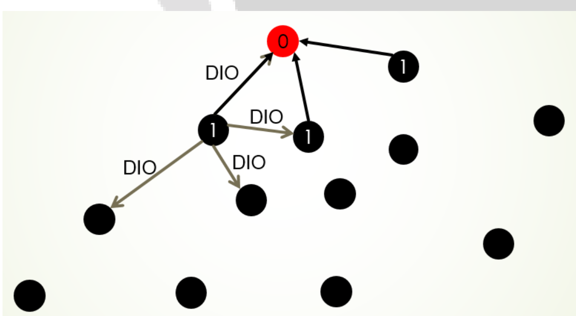


Figure – 1(c)

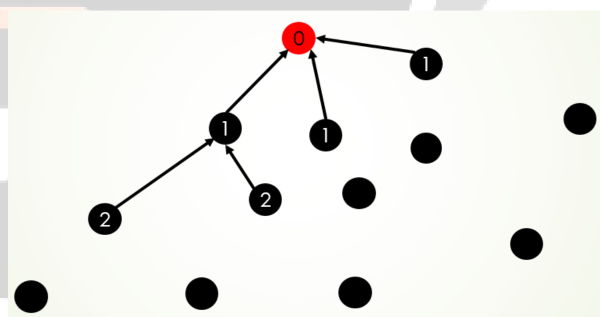


Figure – 1(d)

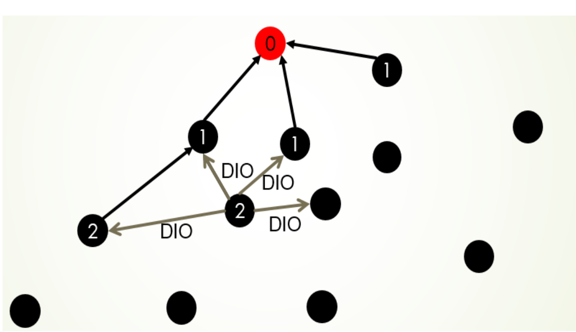


Figure – 1(e)

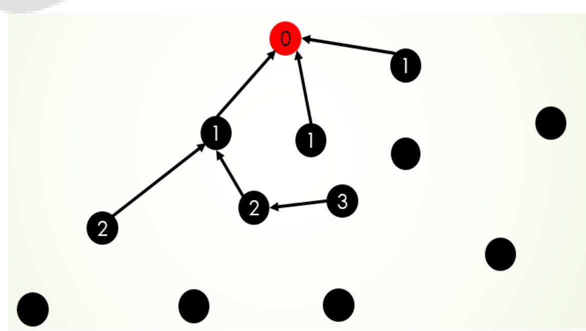


Figure – 1(f)

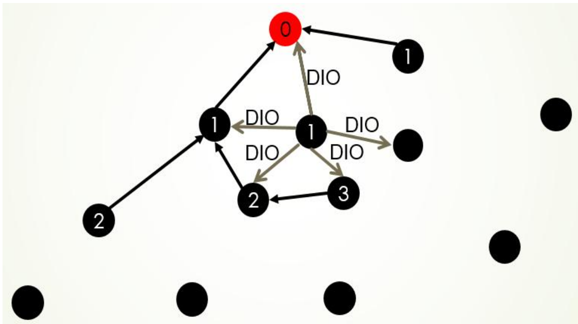


Figure – 1(g)

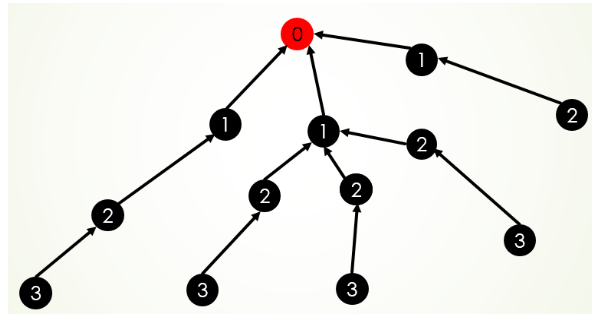


Figure – 1(h)

The Trickle algorithm governs the emission interval of DIOs. The idea is to reduce the control overhead of the protocol by sending DIOs less frequently when there is no change in the topology. In case of a change in network, trickle forces more frequent emission of DIOs. The RPL RFC does not specify the mechanisms for the DAO emission.

As the DAO messages are used to feed the routing tables in the network, they grow with time and size of the network. Nevertheless, no constraint was imposed on the size of the routing table nor on how much information the node can store. The routing table size is not expressed in terms of Kbyte of memory usage but measured in terms of number of entries for each node. Each entry has the next hop node and path cost associated with the destination node. The link ETX (Expected Transmission Count) metric is used to build the DODAG.

3. CHALLENGES IN RPL

During the implementation of RPL, we observe following problems [11, 13, 14]:

Neighbor Unreachability Detection (NUD): A router may receive, transiently, a DIO from a router, closer (in terms of rank) to the DODAG root than any other router from which a DIO has been received. Some, especially wireless, link layers may exhibit different transmission characteristics between multicast and unicast transmissions, leading to a (multicast) DIO being received from farther away than a unicast transmission can reach. DIOs are sent (downward) using link-local multicast, whereas the traffic flowing in the opposite direction (upward) is unicast. Thus, a received DIO may not be indicative of useful unicast connectivity, yet RPL might cause this router to select this seemingly attractive router as its preferred parent. This may happen both at initialization, or at any time during the LLN lifetime as RPL allows attachment to a “better parent” over the network lifetime. A DODAG so constructed may appear stable and converged until such time that unicast traffic is to be sent and, thus, NUD (Neighbor Unreachable Detection) invoked. Detecting only at that point that unicast connectivity is not maintained, and causing local (and possibly global) repairs exactly at that time, may lead traffic not being deliverable.

RPL Implementability and Complexity: Since RPL is designed to be the routing protocol for LLNs, which covers all the diverse applications requirements listed in [2, 3, 4, 5], it is possible that (i) due to limited memory capacity of the RPL routers, and (ii) due to expensive development cost of the routing protocol implementation, many RPL implementations will only support a partial set of features from the specification, leading to non-interoperable implementations.

RPL Underspecification: For DIOs, the trickle timer specifies an efficient and easy to understand timing for message transmission, the timing of DAO transmission is not explicit. As each DAO may have a limited lifetime, one “best guess” for implementers would be to send DAO periodically, just before the life-time of the previous DAO expires. Since DAOs may be lost, another “best guess” would be to send several DAOs shortly one after the other in order to increase probability that at least one DAO is successfully received. The same under specification applies for DAO-ACK messages: optionally, on reception of a DAO, an RPL router may acknowledge successful reception by returning a DAO-ACK. Timing of DAO-ACK messages is unspecified by RPL.

By not specifying details about message transmission intervals and required actions when receiving DAO and DAO-ACKs, implementations may exhibit a bad performance if not carefully implemented. Some examples are:

1. If DAO messages are not sent in tie before the previous DAO expires, the routing entry will expire before it is renewed, leading to a possible data traffic loss.

2. RPL does not specify to use jitter. If DAOs are sent periodically, adjacent routers may transmit DAO messages at the same time, leading to link layer collisions.

Trickle Convergence: Trickle [6] is used by RPL to schedule transmission of DIO messages, with the objective to minimize the amount of transmitted DIOs while ensuring a low convergence time of the network. In real-world environments, however, varying link qualities may cause the algorithm to converge less well: frequent message losses entail resets of the Trickle timer and more frequent and unpredicted message emissions. The resulting higher control overhead due to frequent DIO emission, leads to higher bandwidth and energy consumption as well as possibly to an increased number of collisions of frames.

Loops: In order to trigger a local repair, RPL relies on the "direction" information (with values 'up' and 'down'), contained in an IPv6 hop-by-hop option header of a data packet. If an "upward" data packet is received by a RPL router, but the previous hop of the packet is listed with a lower rank in the neighbour set, the RPL router concludes that there must be a routing loop and it may therefor trigger a local repair. The reason for RPL to repair loops only when detected by a data traffic transmission is to reduce control overhead. However, there are two problems in repairing loops only when so triggered: (i) the triggered local repair mechanism delays forward progress of data packets, increasing end-to-end delays, and (ii) the data packet has to be buffered during repair.

4. POSSIBLE SOLUTION

For problems or challenges listed in section - III, we have discovered the possible solution that helped to improve the performance of RPL. As describe in [1], the rank of routers is calculated based on various Objective Function, based on various matric container like, Minimum Hop, ETX (expected transmission), Delay Estimation, etc. In which MRHOF Objective Function, based on ETX, is widely used in the implementation of RPL. ETX of a link is the expected transmission required to send a packet over that link. The path ETX is the sum of the ETX of all the links along the path. The ETX of a path with 3 links of 100% delivery ratio is 3, whereas the ETX of the ETX of path with 2 links of 50% delivery ratio is 4. But in MRHOF we faced the problems as discussed in section - III.
So we have introduced a new objective function based on the ETX and Link Quality (LQI). LQI is measured by the radio chipset for every received packet, be it a data of control. The algorithm for the proposed Objective Function is shown in Algorithm 1.
Each RPL router listen to the DIOs from neighboring nodes. If it does not receive any DIO, it has no neighbors and is isolated from the rest of the network. The node will then be in idle mode and continue to send DIS messages until receiving a DIO to join a DAG (when it comes within range of a node). When RPL router receives the first DIO, it computes its rank and LQI using Algorithm 1, and broadcasts the updated DIO message to neighbors. If it receives multiple DIOs node selects the parent. This it does based on the ETX and LQI threshold. It selects the parent which has minimum path ETX and LQI greater than the threshold we have set. This operation is shown in Algorithm 1.

Algorithm 1 Operation of a router to select the parent

```
repeat
    broadcast DIS
until receive DIOs
for all replies do
    if LQI (received DIO) ≥ LQI_THRESHOLD then
            Best parent ← Neighbor with min path ETX
    else
     Neighbor node will discard DIO {Node should not select a parent with a lower Link Quality (LQI) then
     threshold}
Schedule DAO to build downward path.
```

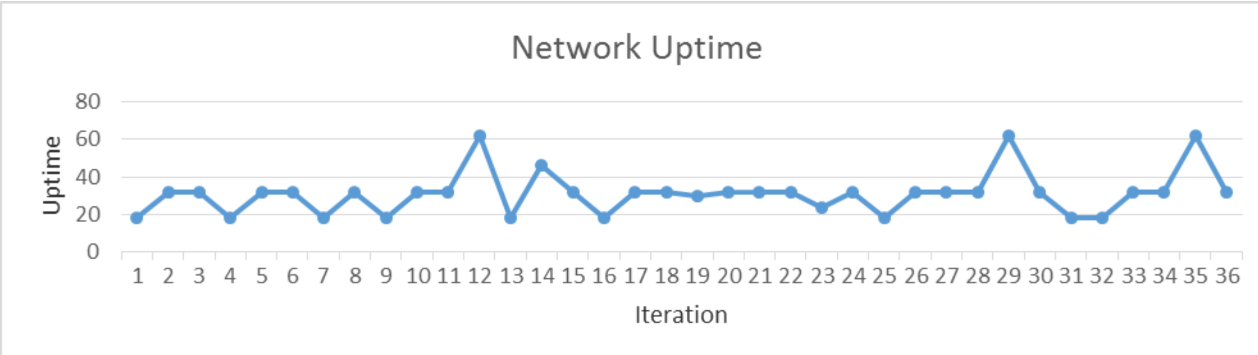
5. RESULT

We have performed several tests on this new Objective Function and compared the results with the MRHOF Objective Function. The results of these tests are:

Network Uptime: The average uptime of network of 20 nodes after complete rebooting is 26 seconds. We have performed this test for both type of OFs and we get almost same result that proves that our solution has no impact on network uptime.

Packet Loss Ratio: In the network of 40 nodes, we sent the commands to each nodes at 2 second interval. We got 99.88% availability of nodes, means only 0.12% packets have been loss, compared to 0.98% packet loss in MRHOF.

Average Packet Delay: In this test, we ping each node in different hops from gateway. We observed average delay of 8-10 ms for each hop.



Network uptime for ETX+LQI based OF

Test Result Overview	Percentage %	Sent	Received
Cloud to Gateway	100	27762	27762
Gateway to Cloud	98.90	26868	26841
Gateway to Node	99.8	27762	27714
Node to Gateway	98.6	27714	26868

Packet delivery ration of MRHOF

Test Result Overview	Percentage %	Sent	Received
Cloud to Gateway	100	27930	27930
Gateway to Cloud	99.91	27867	27843
Gateway to Node	99.93	27930	27913
Node to Gateway	99.83	27913	27867

Packet delivery ration of MRHOF + LQI

6. CONCLUSION

From the results we can say that our new objective function is more efficient than MRHOF. Main cause for packet loss was the increased traffic due to triggering of trickle timer that cause the frequent emission of DIOs due to poor link quality, but as results shows our OF has significantly improved the packet delivery ratio from 99% to 99.8%. Also the problem caused by NUD is also solved by this solution.

REFERENCES

[1] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur, “RPL: IPv6 Routing Protocol for Low power and Lossy Networks," March 2011, Internet Draft, work in progress, draft-ietf-roll-rpl-19.

[2] J. Martocci, P. D. Mi, N. Riou, and W. Vermeylen, “Building Automation Routing Requirements in Low Power and Lossy Networks," June 2010, Informational RFC 5867.

[3] K. Pister, P. Thubert, S. Dwars, and T. Phinney, “Industrial Routing Requirements in Low-Power and Lossy Networks," October 2009, Informational RFC 5673.

[4] A. Brandt, J. Buron, and G. Porcu, “Home Automation Routing Requirements in Low-Power and Lossy Networks," April 2010, Informational RFC5826.

[5] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, “Routing Requirements for Urban Low-Power and Lossy Networks," May 2009, Informational RFC 5548.

[6] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, “RFC6206: The Trickle Algorithm," March 2011, Standards Track RFC 6206.

[7] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks," September 2007, Standards Track RFC 4944.

[8] S. Deering and R. Hinden, “RFC2460: Internet Protocol, Version 6 (IPv6) Speci_cation," December 1998, Standards Track RFC 2460.

[9] J. Hui and P. Thubert, “Compression Format for IPv6 Datagrams in Low Power and Lossy Networks (6LoWPAN)," February 2011, Internet Draft, work in progress, draft-ietf-6lowpan-hc-15.

[10] Oana Iova, Fabrice Theoleyre, Thomas Noel, “Stability and Efficiency of RPL under Realistic Conditions in Wireless Sensor Networks”

[11] Leila Be Saad, Cedric Chauvenet, Bernard Tourancheau, “ Simulation of RPL Routing for IPv6 Sensor Networks: two case studies”, SENSORCOMM 2011: The Fifth International Conference on Sensor Technologies and Applications, ISBN: 978-1-61208-144-1

[12] R.Rajeswari, N.Magadevi, “Self Organizing Mesh Radio Based Solution For Smart Metering Communication”, International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459

[13] Gen Xu, Gang Lu, “Multipath Routing Protocol for DAG-based WSNs with Mobile Sinks”, Proceeding of the 2nd International Conference on Computer Science and Electronics Engineering.

[14] Kevin C. Lee, Raghuram Sudhaakar, Lillian Dai, Sateesh Addepalli, Mario Gerla, “RPL under Mobility”.