Packet Delay Analysis of the RPL routing protocol for Wireless Sensor Network

Utpal Solanki, Gauthamvarma Nandyala
Drexel University
email:{utpal, gn75}@drexel.edu

***Abstract*- Foreseeing billions of connected things, there are recent research and engineering efforts carried out by Internet Engineering Task Force (IETF) working group. Along with IETF, Routing over Low power Lossy networks (RoLL) working group have proposed drafts for interoperable network protocols for Wireless Sensor Networks (WSN). Routing Protocol for Lossy networks (RPL) is an IETF draft that suggests standard routing of IPv6 packets for WSN. In this paper, we present performance evaluation of end to end packet delay with RPL enabled sensor nodes. We present results of simulations and compare it with [1] and suggest proper justification of analysis.**

***Keywords*- RPL, Wireless Sensor Networks, Cooja, 802.15.4, IPv6.**

I. INTRODUCTION

Recently research in WSN has picked up momentum due to bright future of Internet of Things (IoT). WSN is an essential part of IoT by enabling billions of connected things. Tiny sensor nodes in WSN, now commonly called as nodes in rest of the paper, are constrained embedded sensing devices. This devices are made up of resource constrained microcontrollers and add-on sensors. They have limited computing and memory resources. These nodes has wireless connectivity in form of light weight protocols such as IEEE 802.15.4, Bluetooth and Wi-fi. Out of these, 802.15.4 is a very popular among WSN applications because of its design feasibility within tiny embedded device. Its low data-rate, easy operations and power aware implementation give perfect fit for cheap and reliable applications. When these nodes are ameliorated with internet connectivity, they create unbounded and meaningful applications in agriculture, smart-city, home automation, industry automation and healthcare [2].

WSN classify a different part of IoT where it focuses on various nodes that are inter-connected and few of them have actual internet connectivity. Rest of the nodes in network form specific network routing topology to inherit internet connectivity. IETF suggest a draft of compressing IPv6 packets over IEEE 802.15.4 MAC known as 6LowPAN [3]. It allows sending IPv6 packets with compressed header specially designed for 802.15.4. Using 6LowPAN protocol, every 802.15.4 enabled nodes are able to communicate with each other based on complete IPv6 network layer. IEEE 802.15.4 protocol has a MAC (physical) address of 8 Bytes that suggest 264 bits of unique address space for IPv6 base address. This is a plenty of address space if we have to connect billion devices with unique public IPv6 address. Though, the general trend in industry is to connect one or few nodes to internet and form WSN under that node.

After IPv6 and 6LowPAN, one of the major concern to robust WSN is routing mechanism. As discussed earlier, sensor nodes are not heavy enough to implement routing protocol that are utilized by ISPs worldwide. Concept of routing with WSN is totally different. Routing under WSN has to be with minimal overhead, minimum latency and must have self-healing mechanism. Nodes for traditional application require self-healing of routing under dynamic wireless environment. Routing Protocol for Lossy networks (RPL) is one of the robust, self-healing and low latency routing protocol for IPv6 enabled WSN. RPL is reliable, has less overhead and is implemented at network layer that allows easy inter-operability. Using Direct Acyclic Graph (DAG), it can scale nodes from few hundreds to thousands of sensor nodes under one sink node. Among many performance characteristics of RPL, for rest of this paper, we measure node to node packet delay with RPL as routing protocol and then present results of our simulation analysis for various hop of nodes. Section II provide overview of RPL, section III gives details of simulation environment and testbed. For this paper, section IV presents statistical results and suggests a possible reason with proper mathematical model.

II. OVERVIEW OF RPL

RPL is a distance vector routing protocol used

in Low power Lossy Network (LLN) [4]. RPL supports different types of network layers like one which are noisy and the one's in conjunction with the host or router devices with limited resources. The network devices connected in this protocol are connected in such a way that there are no cycles in between them. RPL is also known as gradient based routing protocol. In RPL the gradient is the sink to which the nodes are connected. RPL minimizes the cost to reach the sink through objective function. The object function specifies how routing constraints and other functions are taken into account during topology construction. RPL prevent routing loops-cycles in connection through Directed Acyclic graph (DAG).

A DAG is divided into one or more Destination oriented DAG's (DODAG). One DODAG per sink. Each DODAG is identified by RPL instance id, DODAG id (set by DAG root), DODAG version number (DODAG iteration number) and Parameters advertised by DODAG route. The protocol tries to avoid the cycles by computing nodes position related to other nodes position this position are known as ranks. The rank increases when node moves away from the path they decrease when they are in the path. The RPL specifies four type of control messages for topology maintenance and exchange of information. The first one is called DAG information object (DIO). DIO stores the current rank, RPL instance and IPv6 address of the route. The DIO messages are sent for DODAG's recovery and maintenance and to multicast to the required nodes according to trickle timers. The second one is Destination Advertisement Object (DAO). It enables the down traffic and propagates destination information upwards along DODAG path. The third one is named DODAG information solicitation(DIS) and makes it possible to get DIO information from the nearest neighbor. The fourth and the final is DAO-ACK and is sent by the recipient in response to DAO message.

To avoid redundancies and controlling the signal overhead we use trickle timers. The control plane traffic over load is a major concern in that of the LLN's where bandwidth and power are often scarce. The emission of this control messages is not possible at the same time keeping it alive for routing adjacency is not useful. A different approach in RPL consists of controlling the control plane packets frequency update by using adaptive mechanisms controlled by the use of dynamic timers, referred to as trickle timers. DIO messages are "multicast" on expiry of trickle timers thereby the DIO messages are sent more frequently when a DAG consistency issue is detected to improve the convergence time. As DAG stabilizes, messages are sent less frequently. In the trickle algorithm, the time is split in an endless sequence of intervals with size I. The size of I is not fixed but it is varied over a time of range [I,2I]. In particular, starting from the minimum size I, I is doubled, at the end of each interval, until maximum number of times M. When a cycle is detected (e.g., there is the detection of a loop or the join to a new DODAG version), the trickle timer is reset; that is, I is set to the value 2I.

III. SIMULATION ENVIRONMENT

We measure node to node packet delay under RPL network with following tools and setup. We decided to use Cooja simulator of Conitki-OS [5]. Cooja is WSN simulator that not only simulate sensor microcontroller but also simulate radio environment and 802.15.4 MAC protocol. We found that Cooja has proper debug logs and simulation environment that are appropriate for packet delay calculation. Cooja can simulate more than hundreds of nodes with proper host machine. Since Cooja is designed to run as singly process, it could really optimize if scaled to multi-process system. We restrict our simulation with nodes till 5 hops due to host machine limitation.

Test network has total 15 nodes where 1 node is a sink node which initiate the RPL network. Rest of the 14 nodes are connected with the sink node in form of binary tree topology as shown in figure 1. Nodes with direct hop under sink node are referred as Node2, nodes with third hop are Node3 and likewise. Each node sends a test packet every minute to sink node. Starting time for this interval is randomized among different node. Nodes create a test packet of UDP with payload size of 60 Bytes with destination prefix IP of sink node. Packet sending time on sending node and reception time on sink node are logged in Cooja log buffer for later analysis.

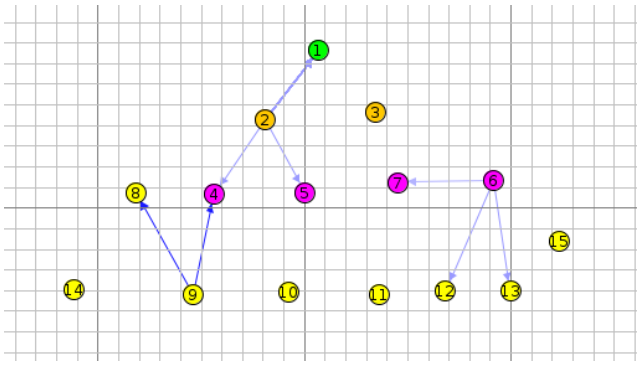

figure 1.0, Tree Topology for Test Bed

Contiki-OS provides different options for network layer, MAC layer and radio layer selection. Our network layer is fixed with RPL and 6LowPAN.

We have used Contiki MAC as 802.15.4 MAC. Nodes use Radio Duty Cycling (RDC) stack to simulate low power device. Most of the time, nodes are sleeping with radio off. It periodically wakes up for certain time. It is sender node's responsibility to send probe packets until target node is awake and has acknowledged packet with acknowledgement frame of 802.15.4. A full data frame with long unicast address takes around 4ms to transmit at a rate of 256 kbps. We have used Packet Error Rate (PER) of 0.0. Unlike [1], we have used single DODAG initiated by sink node.

IV. SIMULATION RESULTS

In this section, we present performance analysis of end to end packet delay with reference to the test setup described in section III. First, we discuss results and compare it with research results of [1]. Later, we give possible suggestions and mathematical modelling of packet delay in terms of RPL overhead, RPL rank of nodes and 802.15.4 implementation. We also describe how nature of random variable function can significantly affect packet delays.

Measuring packet delay is challenging especially between two different nodes which has no common time reference. Nodes may have time drifting error due to underlying hardware and software design. Since we are not analyzing network with actual hardware of sensor nodes, we were able to calculate proper timing reference between sender node and sink node. Even though, simulation timing for different nodes vary, we were able to establish a proper delay calculation probe using Cooja log buffers. As soon as software stack of sending node initiate a test UDP packet, it logs the event with timing precision of 1ms. The precision is accurate enough to measure packet delay which has range of tens of milliseconds to several hundred milliseconds. At sink side, as soon as test packet is received at UDP socket layer, sink node logs this event with specific information such as which node sent the packet and what was RPL rank for that node.

We run this simulation test for around 10 simulation hours that gives statistics for around 600 test packets. While simulation is running, it stores logs in a file for later parsing. Using Python script [6] we perform delay calculation for each test packet. We have published all of our test logs and scripts on github repository for further use [6].

Figure 1.1 shows CDF for end to end packet delay for different node ranks of 2,3,4 and 5 in network. It clearly shows that delay distribution is Gaussian random distribution. This results closely match with result of work from [1]. As node's RPL rank increases, its Gaussian distribution becomes narrow with more predictable mean delay. However, authors of work [1] only comment on performance and shows results. Their work does not represent any suggestion that why we get this particular results. We believe that we have strong statistical represent for this problem as follow.

Our test bed, Cooja simulator and Contiki Software, for SkyMotes uses random number generator library with Gaussian distribution. This random variable selection significantly impacts back-of timing for CSMA-CA of 802.15.4 simulated within Contiki MAC for 802.15.4 frames. Back-of timing is important factor in deciding end to end packet delay especially when it is lossy network with low data rate [7]. WSN is more about packet routing. When end to end delay is calculated for a packet that travel through various hops. This routing contribute to more random variable added to end delay. It can be described as sum of N independent random variables, where N is number of nodes through which test packet has to travel. Other than back-of timing, trickle timer based RPL traffic also contribute to overall delay of packet. As packet moves upwards (in direction of sink node), the probability that test packet is delayed due to overhead of RPL traffic becomes exponentially high.

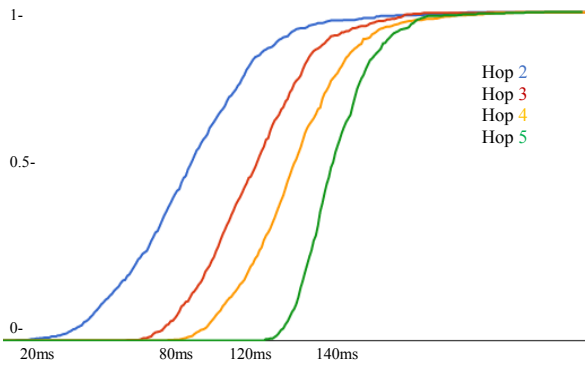

figure 1.1 CDF for packet delay

The end to end delay is a function node's RPL rank and back-of timing for 802.15.4. If $X_{back\text{-}of}$ is a random variable representing packet delay introduced because of CSMA-CA check and probing, X_{RPL} as a random variable representing packet delay due to traffic overhead then final delay sample space S at sink node can be modeled as follows. Note that X_{RPL} is a function of RPL hop rank of node that introduce RPL overhead.

$$S = \sum_{i=0}^{N} (X_{RPL}(i) + X_{back\text{-}of})$$

where N is RPL hop count for sender node.

It can be seen in figure 1.1 that as hop rank increases Gaussian distribution becomes narrow. As hop increases, obviously mean value for delay increases but variance decreases. End to end packet delay for first hop varies from 10ms to 100ms. Each hop adds on an average 40 to 80ms delay. As hop count increases, packet arrival time becomes more predictable. It is observed that if network topology is maintained within maximum 8 hops, probability that packet has end to end delay of less than 1 second is high. Above 8 hops, RPL generate significant percentage of traffic overhead near the sink nodes as mentioned with [8]. CDF for hop 2 to 5 partially match with CDF present in [1]. We were able to generate CDF that closely match with CDF presented for network of 100 nodes with [1]. It is clearly visible that as hop count increases, CDF converge to a sharp upward line suggesting high mean and less variance of Gaussian distribution.

Hop Count	Mean μ	Variance δ
2	86ms	22ms
3	119ms	14ms
4	136ms	13ms
5	141ms	5ms

Table 1.0

V. CONCLUSION

In this paper, we performed simulation for end to end packet delay calculation using COOJA WSN simulator for Contiki-OS. We presented our mathematical model for estimating packet delay from sending node to sink node. We also showed that delay time distribution is Gaussian random process. On sink node, packet arrival time is sum of independent random variables contributed by each node that participate as route to sink node. We successfully compare our statistics with performance analysis presented by [1].

REFERENCES

[1]. N.Accettura, L. A. Grieco, G. Boggia, P. Camarda's "Performance analysis of RPL routing protocol" in mechatronics(ICM), 2011 international conference on 13-15 april 2011,pp: 767-772.

[2]. K F Navarro, "WSN Applications in Personal Healthcare Monitoring Systems: A Heterogeneous Framework" in 2nd International Conference on eHealth, Telemedicine and Social Medicine, 2010.

[3]. 6LowPan IETF Draft, RFC 4944. (https://tools.ietf.org/html/rfc4944)

[4]. RPL IETF Draft, RFC6554. (https://tools.ietf.org/html/rfc6554)

[5]. Adam Dunkels and Thiemo Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In Proceedings of the First IEEE Workshop on Embedded Networked Sensors (Emnets-I), Tampa, Florida, USA, November 2004.

[6]. Github repository for script source code and simulation material. (https://github.com/utpalsolanki/ECEC632)

[7]. A Gkelias, M Dohler, V Friderikos, A H Aghvami, "Average Packet Delay of CSMA/CA with Finite User Population" in IEEE Communication Letters, Vol. 9, March 2005.

[8]. IETF Draft - Performance Evaluation of Routing Protocol for Low Power and Lossy Networks. (https://tools.ietf.org/html/draft-tripathi-roll-rpl-simulation-08)

[9]. H.Kang's "Improving packet reception performance in high traffic sensor networks" in PerCom 2009, IEEE conference on 9-13 March 2009.pp:1-2

[10]. Nguyen Thanh Long1, Niccolò De Caro1, Walter Colitti, Abdellah Touhafi, Kris Steenhaut's "Comparative performance study of RPL in wireless sensor networks" in SCVT 2012 IEEE 19th Symposium on 16 Nov 2012.pp 1-6.

[11]. Pietro Gonizzi, Riccardo Monica, Gianluigi Ferrari's "Design and evaluation of a delay efficient RPL routing metric" in IWCMC,2013 9th international on 1-5 July 2013. PP 1573 - 1577.