Optimized Trickle Scheme for Low Power Lossy Network in Industrial Internet of Things

Rubina Akter, Mohtasin Golam, Jae Min Lee and Dong-Seong Kim, *Senior Member, IEEE,*
Networked Systems Laboratory, Department of IT Convergence Engineering,
Kumoh National Institute of Technology, Gumi, South Korea.
(rubinaakter2836, golam248, ljmpaul, dskim)@kumoh.ac.kr

Abstract—The building block of internet-of-things (IoT) follows the routing standardization of low power and lossy networks known as RPL for maintaining the routing constraints. The trickle timer algorithm is one of the major components of RPL that controls and maintains the routing frequency through suppression and traffic frequency adaptation mechanism. However, trickle algorithm suffers from the short listen problem that hold some nodes without transmitting messages. As a result, high power consumption and latency is observed in the network. Therefore, our proposed trickle scheme deal with the short listen problem and handle message transmission in a dynamic way. Thus the proposed algorithm optimize the latency and the power consumption. Simulation results show that the proposed algorithm has exhibited the superior performance than the existing algorithm in terms of packet delivery ratio.

Index Terms—Low power and lossy networks (LLN), routing protocol, trickle algorithm, wireless sensor networks.

I. INTRODUCTION

Now a days industrial internet of things (IIoT) is considered a new emerging topic that offer a wide research attention in modern era [1]. The main vision behind IIoT is to integrate sensor nodes that establish inter communication through standard communications protocol [2]. In this context, energy consumption and long life time of sensor nodes is the main concern as they run on a very stringent energy budget. Accordingly the routing algorithm choose the best path between the sensor nodes. IoT platform use IPv6 over low power and lossy networks (RPL) [3]. Basically, RPL is a proactive distance-vector routing protocol and constructs topology through destination-oriented directed acyclic graphs (DODAGs) [4]. The main components of RPL [5] is trickle timer algorithm that govern the routing decisions and constructs the DODAG topology in low power and lossy networks (LLNs). During construction of DoDAG topology LLN follows two types of routing matrices such as, the link metrics and the node metrics. The duty cycle maintains the power dissemination by turning the radio off and on depends on the data transmission. The basic idea behind trickle algorithm is to exchange the code and update all nodes in simple, yet scalable and energy efficient way.

The goal of trickle [6] is achieved through two approaches. The first approach is adaptively increases the signaling rate if there any inconsistent state is found. In contrast, when the network returns back the steady state it rapidly and exponentially reduces the signaling rate. The second approach involves the network suppression mechanism which aim to reduces energy consumption. In the suppression mechanism, one node suppress the transmission of same control packet once the node noticed that the same data packet is available in network which is transmitted by the neighboring nodes. However, due to the fixed listen only period in the first half of each interval trickle algorithm increases the delay in propagating transmission. As a result, high network convergence time is required specially when networks choose higher values for minimum interval period. Motivated by the aforementioned issues this paper proposed an algorithm that solves the short listen problem and fairly manages the data transmissions. The simulation result illustrates that the proposed algorithm reduces the energy consumption and speeding up the data transmission rate.

II. PROBLEM FORMULATION

Trickle is a propagation scheduling algorithm that governed the transmission cycle following three configuration parameters, three trickle variables and six steps. To configure the timeline, trickle uses three configuration parameters such as, the minimum interval size (I_{min}), the maximum interval size (I_{max}) and the redundancy factor (k). Furthermore, trickle uses three variables to maintain the current state such as, the length of the ongoing interval size (I), a message counter (c) and the transmit time (t) which is randomly chosen from the current interval.

Trickle sets an intervals for each node and the range of interval is (I_{min}, I_{max}). When interval starts, trickle sets the counter x to 0 and randomly select the transmit time t from sub-interval which range is ($\frac{I}{2}, I$). When a consistent message is received the message counter increment by 1. At the randomly chosen transmit time t, the value of counter c is checked with the redundancy constant k, whether it is greater or equal to the redundant factor. If the value of x is less than k then message will transmit otherwise suppress its scheduled messages. After the completion of interval I, trickle doubles the size of the interval. This process continue until it reaches the end of main interval (I_{max}). Subsequently, if inconsistent message is detected trickle reset I to (I_{min}) and starts a new interval as previous steps. The problem encountered in trickle algorithm is the listen-only period in every interval. During this periods, a node only able to listen and receive messages that impose a delay at least $\frac{I}{2}$ in each interval. Therefore, a network consisting n-hops progressively increase the delay in
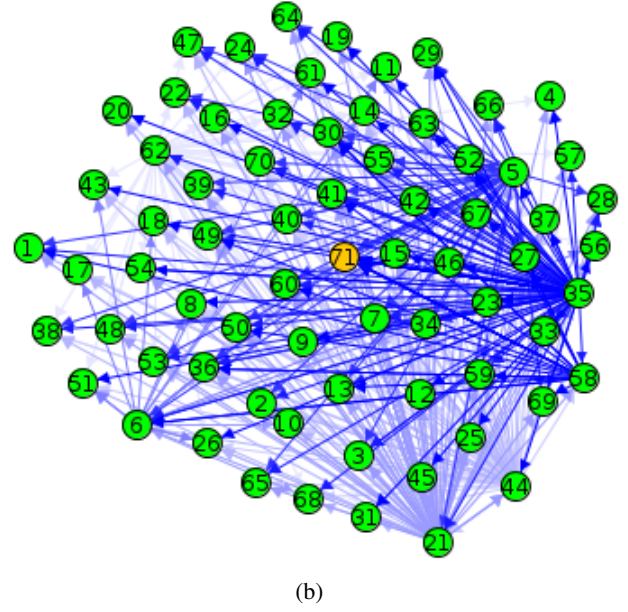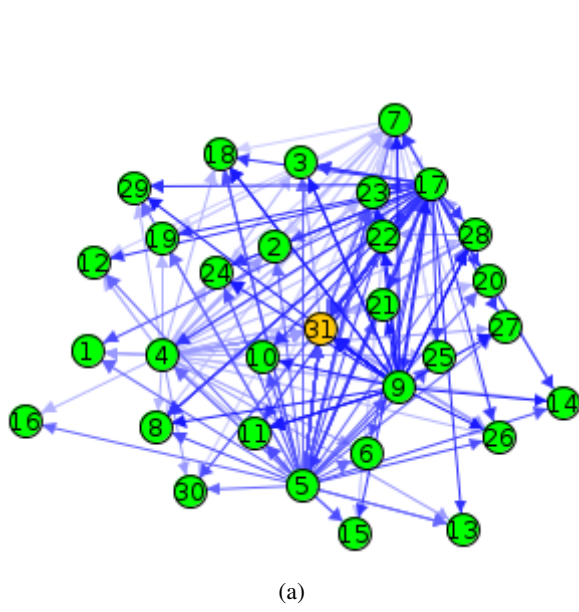
Figure 1: Network model with (a) 30 nodes and (b) 70 nodes

Algorithm 1: Proposed-Trickle Algorithm

1 `Initialization();`
2 $I \leftarrow I_{min}$;
3 `IntervalBegin();`
4 $c \leftarrow 0$;
5 $t \leftarrow random\,(0, I)$;
6 $I \leftarrow I \times 2$;
7 **if** $I \geq I_{max}$ **then**
8 | $I \leftarrow I_{max}$;
9 **end**
10 `ConsistentMessageTransmission();`
11 **if** $c < K$ **then**
12 | Receive message and increment message counter;
13 | $c \leftarrow c + 1$;
14 **end**
15 **if** $c = K$ **then**
16 | Stop message receive and transmit;
17 **end**
18 `IntervalEnd();`
19 $c \leftarrow 0$;
20 **if** $InconsistentTransmissionReceived$ **then**
21 | $I \leftarrow I_{min}$;
22 | $c \leftarrow 0$;
23 | $t \leftarrow random\left(0, \frac{I}{2}\right)$;
24 **else**
25 | $I \leftarrow I * 2$;
26 | **if** $I \geq I_{max}$ **then**
27 | **end**
28 | $I \leftarrow I_{max}$;
29 **end**

every hop as the delay is proportional to the number of hops. As a result the network will suffer from high convergence time

Table I: Simulation Parameters

Parameter Name	Values
Simulation time	10 min
Redundancy factor	3
Area of simulation	100m * 100m
TX range	50m
INT range	50m
Radio medium	Unit Disk Graph Medium (UDGM)
Reception success ratio	100%
Transmission success ratio	100%
Nodes	30, 50
Adaptation layer/ MAC	6LowPAN / ContikiMAC
Data packet rate	60 sec
Objective function	MRHOF

and high latency.

III. PROPOSED ALGORITHM

In previous section, we analyzed the behavior of trickle algorithm and noticed that the listen only period of enhance the network delay. Moreover it increases the network latency due to the increment number of counter message than the redundancy constant that ignores all the received control messages from the previous interval. To overcome this short-comings and make trickle more efficient we include a simple but very effective message transmission control mechanism. Based on the observation of transmission control messages in trickle scheme, we select the value of random transmit time, t from the range $[0, I]$ instead of $\left[\frac{I}{2}, I\right]$. In the first phase, the interval time has initialized from the ranging of $\left(I_{min}, \left(2^{I_{max}} \times I_{min}\right)\right)$ and the message counter c to 0. The random transmit time t has selected from the range $[0, I]$. In consistent message transmission phase, for every message receiving has increased the value of counter and check with the value of redundancy constant. When the value of message
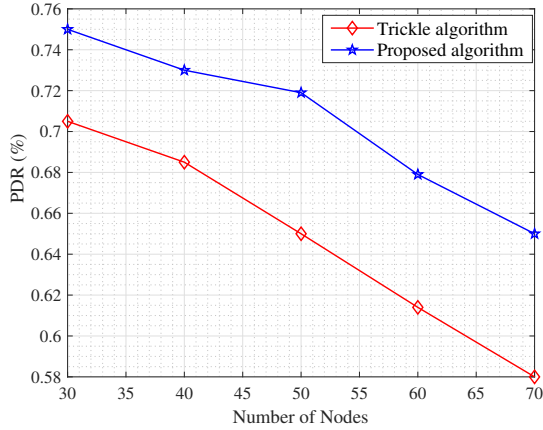
Figure 2: Packet delivery ratio with various densities

counter reaches the threshold value of redundancy constant then the node will stop to receive the message. Concurrently the transmit time t will be selected and transmits the received messages. The concept behind to check always the value of message counter and redundancy constant is to make update the system to transfer and receive the message till to the threshold value of k. As a results, all nodes can equally participate to transmit the consistent messages and the system can enhance the network latency as avoid the failure of message transmission. After the completion of one interval the system reset the counter to 0. It is worth noting that, if system found any inconsistent transmission, at that case the solving time t will be randomly selected from $\left[0, \frac{I}{2}\right]$ that allow the system more faster by reducing the transmission delay and extra cost.

IV. PERFORMANCE EVALUATION AND DISCUSSION

In this section, we evaluate the performance of our proposed algorithm and compare to the existing trickle algorithm in terms of packet delivery ratio. The simulation has been carried out by means of ContikiOS over the Cooja simulator which is particularly designed for low-power and lossy devices. The network scenarios' has designed with random topology consisting 30 and 70 nodes and placed the sink in middle of the network has shown in Fig. 1. In order to proper DoDAG formation, the network has chosen the minimum rank with hysteresis objective function (MRHOF). The simulation parameters has shown in Table I. Fig. 2 illustrates the comparative analysis of trickle algorithm and proposed algorithm in terms of packet delivery ratio (PDR). Basically, PDR is the ratio of total number of successfully delivered packets to the total numbers of sent packets. It can be inferred from the Fig. 2 that in proposed algorithm scheme the target nodes receives higher number of packets than that of trickle algorithm. It is worth noting that proposed algorithm outperforms the trickle even in the high density network. However, increment number of received control messages rises the probability of collisions and decreased the successfully packet delivery ratio.

V. CONCLUSION

Low power lossy network (RPL) is implemented in network layer which is utilized in the IoT. Consequently to maintain the timing of the flow of control messages is an challenging issue that may lead to load balancing and high network latency problem. The prolonged listen only period forces the nodes to receive for a long time while being incapable of sending messages before completion of that particular time slot. Moreover, increment number of received messages in listen only period enhance the probability of failure ratio to send messages. As a result the nodes may wait for a long time and network latency is increased. Therefore, in this paper an optimized version of trickle algorithm has developed where listen only period has minimized and handled the flow of control messages by fairly maintaining the message counter and the redundancy constant. The performance evaluation revealed that proposed algorithm offered better results in terms of flow of control message that offer high PDR compare to the existing trickle algorithm.

VI. ACKNOWLEDGEMENT

REFERENCES

[1] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 10–16, June 2017.

[2] N. B. Long, H. Tran-Dang, and D.-S. Kim, "Energy-aware real-time routing for large-scale industrial internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2190–2199, 2018.

[3] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Mar. 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6550.txt

[4] T. Clausen, U. Herberg, and M. Philipp, "A critical evaluation of the ipv6 routing protocol for low power and lossy networks (rpl)," in *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2011, pp. 365–372.

[5] N. Accettura, L. A. Grieco, G. Boggia, and P. Camarda, "Performance analysis of the rpl routing protocol," in *2011 IEEE International Conference on Mechatronics*, April 2011, pp. 767–772.

[6] P. Levis, T. H. Clausen, O. Gnawali, J. Hui, and J. Ko, "The Trickle Algorithm," RFC 6206, Mar. 2011. [Online]. Available: https://rfc-editor.org/rfc/rfc6206.txt