

RANDOMIZED DYNAMIC TRICKLE TIMER ALGORITHM FOR INTERNET OF THINGS

Muneer Bani Yassein, Ansam Alnadi, Asmaa Bataineh

Computer Science Department, Jordan University of Science and Technology,
Irbid, Jordan

ABSTRACT

Routing Protocol for Low Power and Lossy Networks (RPL) is one of the most utilized routing protocols. It designed to adapt with thousands of nodes in energy-constrained networks. It is a proactive distance vector protocol which has two major components objective function and trickle algorithm. Our work focus on the trickle timer algorithm, it is used to control, maintain and follow the control messages over the network. Short listen problem is the main blot in trickle algorithm. Several studies focused on enlarging the listen period. However, as it was suffering from node starvation when the period is short, it suffers from time and energy wasting when the period is enlarged. Notice that the time and power consumption are sensitive factors in Low Power and Lossy Networks. In this paper, we propose a randomized dynamic trickle algorithm, it contributes in the improvement of trickle and solving the above-mentioned problems by controlling the t variable in a dynamic randomly way, where t is the border line between listening and transmitting period. The performance of the proposed algorithm is validated through extensive simulation experiments under different scenarios and operation conditions using Cooja 2.7 simulator. Simulation results compared with the standard trickle timer algorithm based on convergence time, packet delivery ratio (PDR) and power consumption performance metrics. The results of the simulations denote a high improvement in term of convergence time, power consumption and packet delivery ratio

KEYWORDS

Routing Protocol for Low Power and Lossy Networks(RPL), Internet of Things (IoT), Trickle Timer Algorithm.

1. INTRODUCTION

Internet of Things (IoT) is an enthusiastic topic in our era. The concept of IoT referred to a set of real world objects that communicate with each other through wireless sensors networks (WSN) [1, 5,16]. A communication means that we can transfer the information and knowledge between components through a media [3, 5]. Many routing protocols are used to achieve this goal [2,14,15]. In order to employ and use IoT functions, we need machine to machine protocols [3]. Daily, large number of devices emerges and creates a huge communication links between each other in every time unit [4]. The emergence of smart things makes our world smarter [5]. Using IoT, many applications in real life will be better in the future like education, healthcare, industry and others [6, 8]. In wireless sensor networks, there are two types of nodes: a source node that

transmits the data and a sink node that receives the data [1]. a source node sends a packet to the destination node through a path. There are many routing protocols used to explore the best path, each one has a specific algorithm to get a discovery path to arrive a destination node [1]. One of the most important protocols is the Routing Protocol for Low Power and Lossy networks (RPL) [7, 8]. RPL uses algorithm called Trickle Timer Algorithm to deliver a packet from a source to a destination [7, 8]. When we study the RPL protocol, we focus on three issues: power consumption, packet delivery ratio (PDR) and convergence time. Because of the limitations of the resources [8, 6] like a battery, always our target is minimize a power and time and to maximize the number of delivered packets [8]. The main problem in trickle timer algorithm is short listen period problem, so this leads some nodes to be starved or suffered from a long latency time [7, 1]. In trickle algorithm nodes are always on [9], so this consumes a high percentage of the power. a lot of researchers work to reduce the limitations of this algorithm. They proposed enhancement algorithms such as: Elastic Trickle Algorithm for Low- Power Networks and Internet of Things (Trickle-Plus) [7], fair broadcast suppression (Trickle-F) [8], a new elastic trickle timer algorithm [1], A New Dynamic Trickle Algorithm for Low Power and Lossy Networks [10], Adaptive-k algorithm [11], Trickle-D algorithm [12] and others.

In this paper, we propose an enhancement algorithm of standard trickle timer algorithm called randomized dynamic trickle timer algorithm (RD-Trickle) to meet the problems in standard trickle algorithm in term of three parameters: power consumption, packet delivery ratio (PDR) and convergence time. The rest of the paper is divided as the following: section 3 talks about literature review, section 4 talks about methodology, section 4 talks about conclusion, and section 5 talks about references.

2. TRICKLE TIMER ALGORITHM

Each node in Trickle timer algorithm has many intervals [1]. Major interval begins from I_{min} value and ends with I_{max} value [7]. Both I_{min} and I_{max} values are determiners variables of the major interval. Major interval contains subintervals; each interval begins from I_{start} value and ends with I_{end} value. Both I_{start} and I_{end} values are determiners variables of each subinterval. Subinterval begins with I_{start} value, so $I_{start} = I_{min}$ and ends with $I_{end} = I_{start} * 2$ [1]. At starting point, the first subinterval is executed until the end, then a next subinterval starts and the same thing for all subintervals until reaches I_{max} value [10], which indicates that a major interval is ended. The standard trickle algorithm consists of following parameters:

1. I_{min} : minimum length of the interval
2. I_{max} : maximum length of the interval
3. K: redundancy factor

Also, the following parameters are maintained in trickle algorithm [13]:

1. I: current interval length.
2. C: counter.
3. T: random time within the current interval.

3. RELATED WORK

In IoT, the RPL protocol primarily uses a trickle algorithm to control a flow of messages in term of power consumption, packet delivery ratio and convergence time [7]. Problems in the standard trickle algorithm are considered by multiple researchers. They applied many improvements to enhance trickle limitations. In [7] proposed an enhancement algorithm over a standard trickle algorithm to meet a problem of getting high convergence time with lower power consumption and vice versa. They proposed an Elastic Trickle Algorithm for Low- Power Networks and Internet of Things (Trickle-Plus). Their simulations approved that trickle-plus algorithm made a protocol more flexible. They improved a convergence time and power consumption in an observable way. Advanced Metering Infrastructures (AMIs) consist of a huge number of devices like home devices and education devices, that spread in our real word neither in urban nor in rural environments [8]. C. Vallati et al. [8] proposed a new algorithm named fair broadcast suppression (F-Trickle) to meet the limitations of standard trickle algorithm. F-Trickle chooses a better route from the available routes with insurance of preserving the same number of packets in standard algorithm. Their simulations showed that F-Trickle is effective to choose an efficient route with preserving of the same power consumption value in standard algorithm. M. Bani Yassein et al. [10] proposed a new algorithm called new Dynamic trickle timer algorithm. This algorithm deals with listen only period problems in standard trickle algorithm that effects on convergence time and power consumption [1]. When they applied a new Dynamic trickle timer algorithm through random topology simulations, an enhancement results are observed in term of convergence time, power saving and performance. Also, M. Yassein et al. [1] proposed an algorithm called A new elastic trickle timer algorithm for Internet of Things to deal with listen only period problems in standard trickle algorithm and their effects on power consumption and convergence time[10]. They applied simulations with different number of nodes, and the results indicate that a new algorithm is better than the original algorithm in term of performance, power saving and convergence time. T. Meyfroyt et al. [11] proposed an enhancement algorithm above standard trickle called adaptivek algorithm. Their algorithm made every node adapted its suppression mechanism to local node density. The Adaptive-k algorithm has many advantages over a standard trickle: distribute a load between nodes, adapting the suppression mechanism among different network densities in network topologies and clearly guarantee of the functionality of applying suppression mechanism. A result of simulations showed that a proposed algorithm leads to better performance, easily discovery process of the routes by curb control messages that are considered as redundant. M. Vućinić et al. [12] proposed a Trickle-D algorithm to distribute a load between nodes in fairly way and reduce the number of messages, so reduce the transmissions through a network. By simulations, Trickle-D achieved its goal in term of load distribution in good manner and improvements on performance in an observable way.

4. RANDOMIZED DYNAMIC TRICKLE TIMER ALGORITHM

According to standard trickle algorithm, if a subinterval does not have enough period for listening and transmitting the data a new subinterval doubled to achieve the remaining work. In trickle, always a double value ($I_double = 2$) regardless to the situation of the network, node density, number of direct one hop neighbor or others. We must know if the node places in dense network, so it requires a high double value. When a node has low density because of doubled subinterval by 2 problems appeared like low utility problem and loss of time and power. From this point, a problem appeared in standard trickle algorithm. In order to suppress this problem and give each node the required time to complete all its work, after doing simulations, we proposed an

enhancement above a standard trickle algorithm that is related to t value. As we observed, there is random time implemented in standard trickle algorithm inside each subinterval. In our proposed algorithm, we determine a time for listening period and transmitting period based on t value within a subinterval. t variable is random time used as listening and transmitting determiner. So our work primarily focuses on choosing t value. We follow a mechanism for choosing t value according to the network density. In order to achieve this, we implement four cases to choose better t value in subinterval as explain in Fig.1 below. Every time, we check a number of neighbors (C) to determine in which range we should choose a random time (t). As we observe from these four cases, if the density is low the range of choosing t value is small, but if the density is high the range of choosing t value is large. The Same thing applies to all cases. As a rule, whenever a density increases a range of choosing t value also increases.

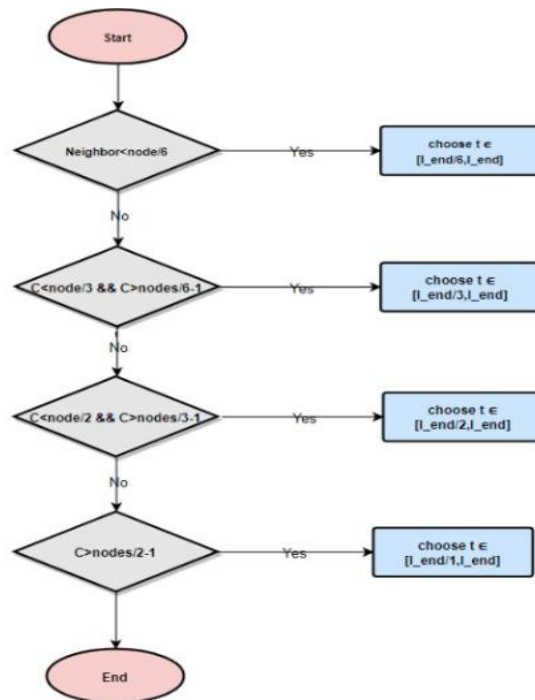


Figure 1. our proposed method to choose (t) value.

Below is the proposed randomized dynamic trickle timer algorithm:

Randomized Dynamic Trickle Timer Algorithm for each nodeInput: I_{min}, I_{max}, threshold value (K), nodes.

Output: control message flow

1	I_min = 212, I_max = 220, I_double = 2, K=1, total_nodes = 20,
2	40 or 80
3	I_start = I_min
4	nbr_count = 0
5	
6	for (I_min; I_min <= I_max ; I_min = I_min * I_double)
7	// main I = [I_min , I_max]
8	{
9	I_end = I_start * I_double
10	for (I_start ; I_start <= [I_end / 2] ; I_start + 1)
11	//sub interval I_sub = [I_start , I_end]
12	{
13	Receiving ()
14	}
15	
16	for (I_end / 2 ; [I_end / 2] <= I_end ; [I_end / 2] + 1 <= I_end)
17	{
18	Receiving ()
19	
20	If ((neighbor_count > -1) && (neighbor_count <
21	[total_nodes / 6]))
22	{
23	time = random number chosen over [I_end / 6 , I_end].
24	}
25	
26	Else If ((neighbor_count > [total_nodes / 6] + 1
27	&& (neighbor_count < [total_nodes / 3]))
28	{
29	time = random number chosen over [I_end / 3 , I_end].
30	}
31	
32	Else If ((neighbor_count > [total_nodes / 3] + 1) &&
33	(neighbor_count < [total_nodes / 2]))
34	{
35	time = random number chosen over [I_end / 2 , I_end].
36	}
37	
38	Else If (neighbor_count > [total_nodes / 2] + 1)
39	{
40	time = random number chosen over [I_end / 1 , I_end].
41	}
42	
43	
44	Sending ()
45	}
46	I_start = I_end
47	}
48	

```

49 Receiving ()
50 {
51     if a message is the newest
52     {
53         C=C+1
54     }
55     else
56         C=0
57     Break
58 }
59 -----
60 Sending ()
61 {
62     if (C < K)
63     {
64         Transmit
65     }
66     else
67         Suppress
68         C=0
69 }
70 }

```

5. PERFORMANCE EVALUATION

This section will show the performance evaluation of RD-Trickle algorithm in terms of packet delivery ratio (PDR), convergence time and power consumption, comparing with the standard trickle algorithm. Simulation experiments were executed using cooja 2.7 simulator based on Contiki operating system. We perform a randomly deployed topology with two different network densities. For more accuracy result, we repeat each experiment around 10 times taking in the account the average of these experiments with getting rid of some of the thumping experiences. Table I shows the detailed simulation parameters used in our experiments.

Table 1. Simulation parameters.

Parameter	value
Simulation tool	Cooja 2.7
Operating System (OS)	Contiki
Computer specifications	8 RAM, 64 bit
Simulation experiment Time	15 minute
Total nodes	40,20
I_max	2 ²⁰
I_min	2 ¹²
Rx (Reception Success Ratio)	100
Tx (Transmission Success Ratio)	100
Transmission Rang	30
Interference Range	30
Network Topology	Random distribution
Radio Medium	UDGM

5.1 CONVERGENCE TIME

5.1.1 RANDOM TOPOLOGY

Fig. 2 shows the average convergence time of our proposed RD- trickle algorithm and the standard trickle algorithm on a different number of nodes deployed on random topology. As the figure shows, our RD- trickle algorithm significantly enhances the time comparing with the standard algorithm on both number of nodes 20 and 40.

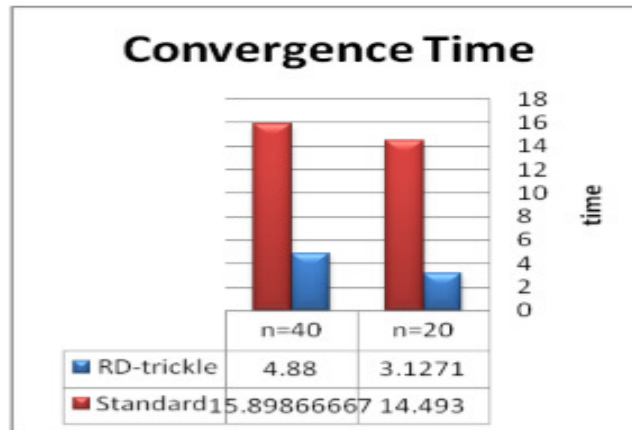


Figure 2. convergence time of randomly deployed nodes: 20 and 40.

5.1.2 GRID TOPOLOGY

Fig. 3 shows the average convergence time of our proposed RD- trickle algorithm and the standard trickle algorithm on a different number of nodes deployed on a grid topology. As shown, our RD- trickle algorithm also significantly enhances the time comparing with the standard algorithm when the number of nodes =20. But when the number of nodes increased to 40, RD- trickle goes on a worse way which increases the amount of consumed time comparing with the standard algorithm.

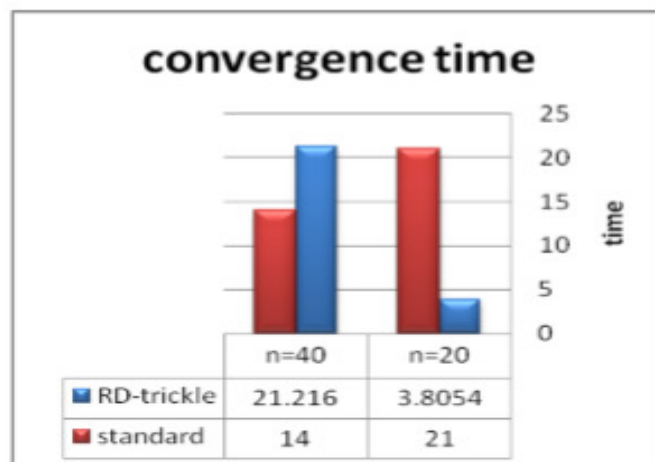


Figure 3. convergence time of grid deployed nodes: 20 and 40.

5.2 PACKET DELIVERY RATIO (PDR)

5.2.1 RANDOM TOPOLOGY

Fig. 4 shows the average packet delivery ratio (PDR) of our proposed RD- trickle algorithm and the standard trickle algorithm on a different number of nodes deployed on random topology. The figure shows that PDR is not affected when the number of node equals to 20, however, it is enhanced when the nodes increased to 40.

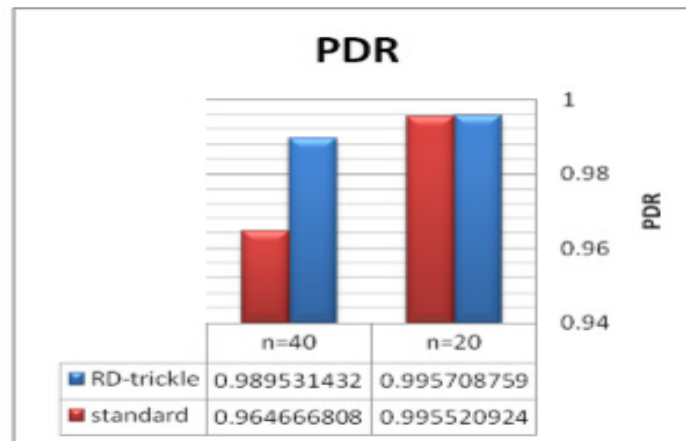


Figure 4. PDR of randomly deployed nodes: 20 and 40.

5.2.2 GRID TOPOLOGY

Fig. 5 shows the average packet delivery ratio (PDR) of our proposed RD- trickle algorithm and the standard trickle algorithm on a different number of nodes deployed on a grid topology. A noticeable packet delivery ratio improvement appears in grid topology on both 20 and 40 nodes comparing with the standard trickle.

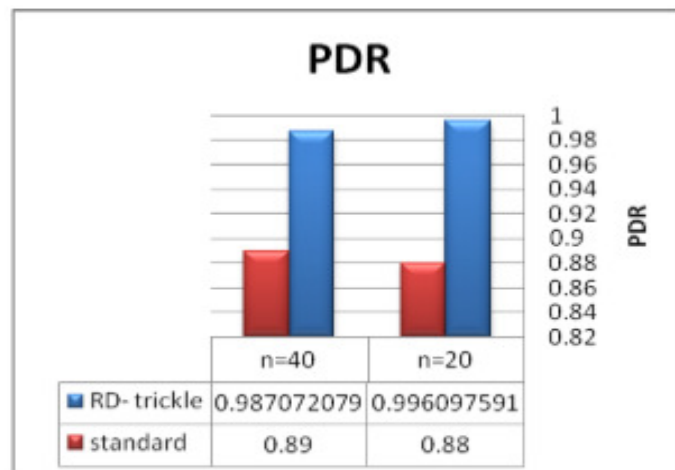


Figure 5. PDR of grid deployed nodes: 20 and 40.

5.3 POWER CONSUMPTION

5.3.1 RANDOM TOPOLOGY

Fig. 6 shows the average power consumption of our proposed RD- trickle algorithm and the standard trickle algorithm on a different number of nodes deployed on a random topology. When the number of nodes equals to 20, RD- trickle almost consumes the same power that consumed in the standard algorithm. Power consumption enhancement arises when the number of nodes equals to 40.

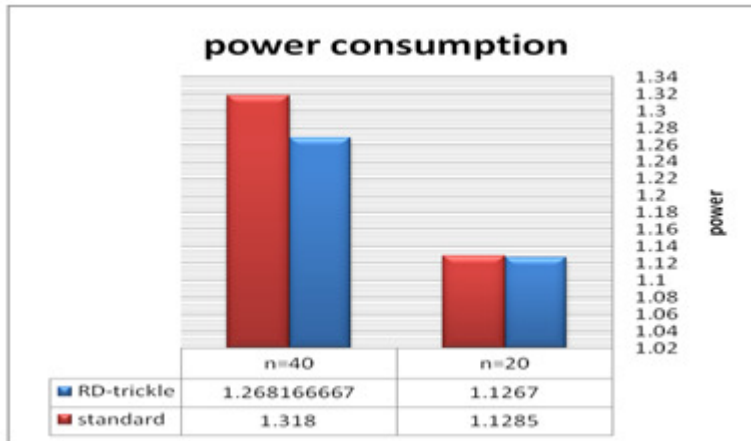


Figure 6. power consumption of randomly deployed nodes: 20 and 40.

5.3.2 GRID TOPOLOGY

Fig. 7 shows the average power consumption of our proposed RD- trickle algorithm and the standard trickle algorithm on a different number of nodes deployed on a grid topology. RD- trickle enhances the power consumption by reducing the amount of consumed power when number of nodes =20, but when the number of nodes = 40 in grid topology RD- trickle consumes power more than the standard trickle algorithm.

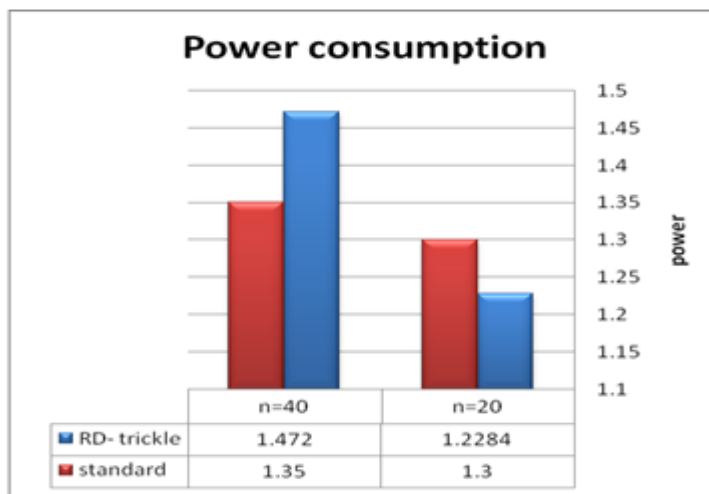


Fig. 7. power consumption of grid deployed nodes: 20 and 40.

6. CONCLUSION AND FUTURE WORK

To control and follow the messages over the Internet of Things (IoT), there is a need for routing protocols. Routing protocol for low-power and lossy network (RPL) is most commonly used routing protocol which is put on the network layer. Trickle timer algorithm is the major component in RPL which is interested in the time of the flow control messages. Trickle algorithm is suffering from short listen only period problem and latency leading to reduce its performance. Popular performance metrics used like packet delivery ratio (PDR), convergence time and power consumption for performance evaluation. If the listen only period set to be long, it will lead to resource wasting and force the nodes to wait with no ability to transmit. On the other hand, short listen only period lead to node starvation and load balancing problems. From this point, the need of a dynamically controlled period has appeared. This study proposed a randomized dynamic trickle timer algorithm (RD- trickle), RD- trickle contributes in solving the above mentioned problems. PDR, convergence time and power consumption used to evaluate its performance. Experiments disclosed that it performs better than the standard trickle in many cases especially in term of convergence time. The best performance appeared on a random topology in both 20 and 40 nodes. Our experiments have executed on cooja 2.7 simulator, Contiki OS. Moreover, some improvements appear in a grid topology, in terms of power consumption and convergence time just when the number of nodes =20, in term of PDR on both 20 and 40 nodes

In the future, we want to pursue study RD- trickle in different RX values noting its behavior and performance. Also, we will examine it to a real service application.

REFERENCES

- [1] M. Yassein, S. Aljawarneh and E. Masa'deh, "A new elastic trickle timer algorithm for Internet of Things", *Journal of Network and Computer Applications*, vol. 89, pp. 38-47, 2017.
- [2] N. Kumar, Y. Singh and P. Singh, "An Energy Efficient Trust Aware Opportunistic Routing Protocol for Wireless Sensor Network", *International Journal of Information System Modeling and Design*, vol. 8, no. 2, pp. 30-44, 2017.
- [3] V. Kalyani, P. Gaur, and S. Vats, "IoT: 'Machine to Machine' Application A Future Vision", *Journal of Management Engineering and Information Technology (JMEIT)*, Vol.2, No.4, pp. 15-20, 2015.
- [4] A. Niruntasokrat, C. Issariyapat and P. Pongpaibool, "Authorization Mechanism for MQTT-based Internet of Things", *Communications Workshops (ICC)*, 2016 IEEE International Conference on, 2016.
- [5] S. Madakam, "Internet of Things: Smart Things", *International Journal of Future Computer and Communication*, Vol. 4, No. 4, pp. 250-253, 2015.
- [6] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey", *Computer Networks*, Vol.54, No.15, pp. 2787-2805, 2010.
- [7] M. Qasem, H. Altwassi, M. Bani Yassein, Ahmed Al-Dubai "Performance Evaluation of RPL Objective Functions", *International Conference on Ubiquitous Computing and Communications (IUCC 2015)*, 2015.

- [8] C. Vallati, and E. Mingozzi, "Trickle-F: Fair broadcast suppression to improve energy-efficient route formation with the RPL routing protocol", Sustainable Internet and ICT for Sustainability (SustainIT), pp. 1-9, Palermo, 2013.
- [9] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks", Computer Science Division, University of California, 2003.
- [10] M. Bani Yassein, S. Aljawarneh, E. Masa'deh, B. Ghaleb, R. Masa'deh, "A New Dynamic Trickle Algorithm for Low Power and Lossy Networks", International Conference on Engineering & MIS (ICEMIS), November 2016.
- [11] T. Meyfroyt, M. Stolikj, and J. Lukkien, "Adaptive Broadcast Suppression for Trickle-Based Protocols", IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM), pp. 1-9, Boston, 2015.
- [12] M. Vućinić, M. Król, B. Jonglez, T. Coladon and B. Tourancheau, "Trickle-D: High Fairness and Low Transmission Load with Dynamic Redundancy", IEEE Internet of Things Journal, 2017.
- [13] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm", RFC 6206, Internet Engineering Task Force (IETF), 2011.
- [14] M. Bani Yassein, W. Mardini, A. Khalil, " Smart Homes Automation using Z-wave Protocol", 2016 IEEE International Conference on Internet of Things and Pervasive System, 2016.
- [15] M. Charalambous, C. Mavromoustakis, M. Bani Yassein, " A resource intensive traffic-aware scheme for cluster-based energy conservation in wireless devices", 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012
- [16] S. Manaseer, M. Bani Yassein, "Pessimistic backoff for mobile ad hoc networks", Al-Zaytoonah University, the International Conference on Information Technology (ICIT'09),2009.