Modeling the Message Count of the Trickle Algorithm in a Steady-State, Static Wireless Sensor Network

Hamidreza Kermajani, Carles Gomez, and Mostafa Hesami Arshad

***Abstract*—Trickle is a transmission scheduling algorithm developed for wireless sensor networks. The Trickle algorithm determines whether (and when) a message can be transmitted. Therefore, Trickle operation is critical for performance parameters such as energy consumption and available bandwidth. This letter presents an analytical model for the message count of a static Trickle-based network under steady state conditions, as a function of a parameter called the redundancy constant and the average node degree. The model presented is validated by simulation results.**

***Index Terms*—Trickle, analytical model, wireless sensor networks, message overhead, message count.**

I. Introduction

TRICKLE is a transmission scheduling algorithm which was initially proposed for code propagation and maintenance in Wireless Sensor Networks (WSNs) [1]. However, researchers have shown that it can be used for different purposes such as multicast propagation, route discovery and control traffic timing [2]. The Trickle algorithm has gained relevance in recent years, since it has been standardized by the IETF as the mechanism that regulates the transmission of the control messages used to create the network graph in the IPv6 Routing Protocol for Low power and lossy networks (RPL) [3].

Trickle is based on a consistency model. When a node receives a message that is not consistent with previously received messages, it communicates its own messages at a high rate in order to resolve the inconsistency rapidly. In contrast, as a node receives messages consistent with previously received messages, the node slows its message rate exponentially down to a minimum rate [2]. For example, in a steady-state, static network running RPL, the rate of the aforementioned messages remains constant once it reaches its minimum value.

Another important component of the Trickle algorithm is the redundant message suppression mechanism. This mechanism limits the message redundancy that may occur in a network due to the fact that a node may receive messages containing the same information from several of its neighbors.

The Trickle algorithm determines whether (and when) a message can be transmitted. Therefore, Trickle is critical for various network performance parameters including energy consumption and available bandwidth. However, to the best of our knowledge, the behavior of Trickle has not yet been modelled.

This letter presents an analytical model for the number of messages transmitted in a steady-state, static network that uses the Trickle algorithm, as a function of the redundancy constant (i.e. the main parameter of the Trickle suppression mechanism), and the average node degree. The model constitutes a tool that will be useful for network engineers and researchers in the planning, deployment and evaluation of Trickle-based networks. The model presented is validated by simulation results.

II. Trickle Algorithm Overview

The Trickle algorithm divides time into variable-size intervals. In each interval, each node attempts to send a message. Whether the message transmission takes place or not depends on rules defined by the Trickle algorithm. Trickle operates based on three parameters, three variables and six main rules. The three parameters used to configure the Trickle algorithm are the following:

- The minimum interval size, I_{min}.
- The maximum interval size, I_{max}.
- The redundancy constant, k.

In addition, the Trickle algorithm requires three variables which track the current status of the algorithm:

- The current interval size, I.
- The tentative transmission time within the current interval, t.
- The number of heard messages within the current interval, c.

The operation of Trickle is based on the following rules:

1) Initially, Trickle sets I to a value in the range [I_{min}, I_{max}].
2) At the beginning of each interval, Trickle resets c to 0 and sets t to a random instant within the range [$\frac{I}{2}$, I).
3) If a node hears a consistent transmission, Trickle increments its counter c.
4) At time t, if the counter c is smaller than the redundancy constant, k, Trickle allows the message transmission; otherwise, the transmission is suppressed.
5) On expiry of the interval I, Trickle doubles the interval length. If the new interval length is greater than I_{max}, Trickle sets the new interval length to I_{max} and executes step 2.
6) If Trickle hears an inconsistent transmission, it resets I to I_{min} and executes step 2.

Manuscript received June 5, 2012. The associate editor coordinating the review of this letter and approving it for publication was H.-P. Schwefel.

This work was supported in part by the Spanish Government through project TEC2009-11453 and FEDER.

H. Kermajani and C. Gomez are with the Entel Dept., Universitat Politècnica de Catalunya (UPC), and i2Cat Foundation, Castelldefels, 08860, Spain (e-mail: {hrkermajani, carlesgo}@entel.upc.edu). H. Kermajani is also with the Computer Dept., Islamic Azad University, Touyserkan branch, Touyserkan, Iran.

M. H. Arshad is with the Mathematics Dept., Islamic Azad University, Touyserkan branch, Touyserkan, Iran (e-mail: mostaf.hesami@gmail.com).

Digital Object Identifier 10.1109/LCOMM.2012.111612.121232

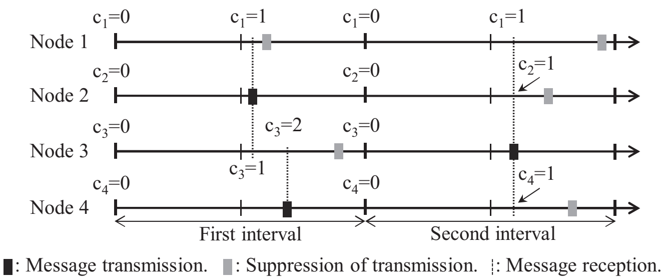

Fig. 1. Example of Trickle operation. Nodes 1, 2 and 3 can hear each other's transmissions, whereas node 4 is only in the coverage range of node 3. The redundancy constant, k, is equal to 1, and I=I_{max} in this example.

Fig. 1 illustrates an example of the Trickle behavior in a steady-state, static network composed of four nodes, where nodes 1, 2 and 3 are in the coverage range of each other, but node 4 is only in the coverage range of node 3. In this example, we assume link symmetry and a value of the redundancy constant, k, equal to 1. In the first interval, node 2 selects the smallest random value of t among the values selected by all four nodes and sends the first message. Nodes 1 and 3 hear this transmission and increase their counters. Since node 4 is beyond the coverage range of node 2, it does not hear this transmission and thus does not increment its counter. The second transmission time t is the one selected by node 1, but this node suppresses its message because the value of its counter (i.e. 1) is not smaller than the value of k (i.e. 1). Subsequently, node 4 is allowed to transmit its message because its counter, which is equal to 0, is smaller than k. Finally, node 3 suppresses its transmission, since its counter value (i.e. 2) is not smaller than the value of k at its scheduled transmission time. In the second interval, node 3 is the first to transmit a message. Hence, all other nodes increase their counters, since they hear this transmission. In the rest of this interval, transmissions are not allowed because the counters of nodes 1, 2 and 4 are not smaller than the value of k (i.e. 1).

III. TRICKLE MODEL

Next, we present an analytical model for the number of message transmissions allowed by Trickle in a WSN within a given time interval (note: by 'time interval' we refer to the time intervals defined by the Trickle algorithm). Our study assumes a static network which is in steady state, where the current interval size of all the nodes is the same and is equal to I_{max}. This assumption is consistent with the characteristics of WSN deployments with very stable links [5]. For more dynamic WSNs, the model provides a lower bound on the number of message transmissions.

For simplicity, our model assumes a synchronized network whereby the interval start time is the same for all the network nodes. Nevertheless, as shown in Section IV, the number of transmissions in a network when intervals have arbitrary start times is very close to the one in a synchronized network.

In our model, we assume a uniformly random spatial distribution for the nodes on a two-dimensional area of size A, and a total number of nodes equal to N. Let the area defined by the coverage range of a node be equal to a. Based on these definitions and assumptions, the probability that a node will be a neighbor of a given node, q, can be calculated as follows:

$$q = \frac{a}{A}. \tag{1}$$

The goal of the model is to calculate the total number of messages transmitted within a given interval in the whole network. In order to achieve this goal, in the following we calculate the average probability that a node belonging to the network, denoted by node x, will send a message in a given interval. This average probability is denoted by $P(tx)$.

Recall that k denotes the Trickle redundancy constant. In order to calculate $P(tx)$, we consider two possible situations: a) node x has fewer than k neighbors; b) node x has at least k neighbors.

Let y be the number of neighbors of node x. Also assume that $P(y < k)$ is the probability that the number of neighbors of node x will be smaller than k, and $P(tx|y < k)$ is the probability that node x will send a message in the current interval when its number of neighbors is smaller than k.

Let us define the term $P(y \geq k)$ as the probability that the number of neighbors of node x will be greater than or equal to k, and the term $P(tx|y \geq k)$ as the probability that node x will send its message in the current interval when it has k or more neighbors. Then, the average probability that a node will send a message in a given interval, $P(tx)$, can be written as follows:

$$P(tx) = P(tx|y < k) \cdot P(y < k) + P(tx|y \geq k) \cdot P(y \geq k). \tag{2}$$

Next we calculate each term of (2) separately. First, when the number of neighbors for node x is smaller than k, the node will surely transmit at time t, i.e. $P(tx|y < k)$=1, because it will hear at most k-1 transmissions during the interval (i.e. its counter c will never reach k). The probability that node x will have fewer than k neighbors, $P(y < k)$, can be obtained as:

$$P(y < k) = \sum_{i=0}^{k-1} \left(\binom{N-1}{i} \cdot q^i \cdot (1-q)^{N-1-i} \right). \tag{3}$$

We next derive the second addend of (2). The probability that node x will have at least k or more neighbors, $P(y \geq k)$, can be calculated as follows:

$$P(y \geq k) = \sum_{i=k}^{N-1} \left(\binom{N-1}{i} \cdot q^i \cdot (1-q)^{N-1-i} \right). \tag{4}$$

Note that each node can have at most N-1 neighbors. In order to calculate the probability of message transmission for node x in the current interval when it has at least k neighbors, $P(tx|y \geq k)$, we consider two possible cases: i) the time selected by node x for the transmission of its message is one of the first k transmission times selected by its neighbors and by itself; and ii) the time selected by node x is not one of those first k transmission times. In the first case, node x will surely send its message at the selected time t. The probability of selecting one of the first k transmission times, denoted by $P(i)$, can be obtained as:

$$P(i) = \frac{k}{y+1}. \tag{5}$$

In the second case, node x can send its message if, at time t, at most k-1 of the subset of neighbors that have selected smaller transmission times than that of node x actually transmit their messages. (Note that the remaining neighbors suppress their transmissions due to the influence of their own neighbors.) If these conditions are satisfied, node x will be able to send its message. The probability that node x will select any of the last y+1-k times, denoted by $P(ii)$ is:

$$P(ii) = 1 - \frac{k}{y+1}. \tag{6}$$

The probability that, in that case, at most k-1 nodes will transmit before node x, denoted Γ, can be expressed as:

$$\Gamma = \sum_{j=0}^{k-1} \left(\binom{n-1}{j} \cdot (P(tx))^j \cdot (1-P(tx))^{n-1-j} \right), \tag{7}$$

where n is the positive integer that denotes the position of node x transmission time in the set of increasingly ordered transmission times selected by node x and by its neighbors. Note that there are n-1 nodes with smaller selected transmission time than the one chosen by node x, and $n > k$.

By plugging (1) and (3)-(7) into (2), we obtain (8), which gives an expression for $P(tx)$, i.e. the average probability of sending a message for node x during a given time interval. Since (8) is a polynomial in $P(tx)$, $P(tx)$ can be obtained from (8) by applying a root-finding algorithm.

Finally, the average number of transmissions in a network composed of N nodes in a given time interval, denoted N_{Tx}, can be expressed as $N_{Tx} = P(tx) \cdot N$.

IV. Evaluation

A. Simulation environment and methodology

In order to validate the model presented, we have developed a simulator using Delphi programming language. In our simulation environment, 100 nodes (i.e. N=100) are placed according to a uniformly random spatial distribution on a square scenario of area 150 m · 150 m. The coverage range of a node is a circular area defined by the transmission radius of the node. Any node contained within the coverage range of another node receives correctly the messages transmitted by the latter. The coverage range (which is the same for all nodes) is varied in order to evaluate the results for different average node degrees. In order to cope with the border effect problem in the simulations, we use the toroidal distance for calculating the distances between nodes [4]. Nodes execute the Trickle algorithm, which has been implemented in the simulator according to the Trickle specification [2]. The simulator allows the network to be configured in two different modes: synchronous mode and asynchronous mode.

In the synchronous mode, all nodes start their first interval (of size I_{min}) at the same time. When the intervals of all nodes reach the I_{max} size (i.e. the network reaches the steady state), the simulator computes the number of transmitted messages for five consecutive intervals.

In the asynchronous mode, each node randomly chooses a start time for its first interval (of size I_{min}) in the range between 0 and I_{max}. Thus, the last Trickle execution start can occur while the node that selected the first start time is already in its first interval of I_{max} size. When the last node starts its second interval of I_{max} size, the network reaches steady state (note that, during the previous interval of I_{max} size, the neighbors of the last node may have a decreased transmission probability, since the last node's interval of 0.5·I_{max} size favored its own transmissions). Then, the simulator computes the number of transmitted messages for five consecutive intervals.

The procedures described for both synchronous and asynchronous modes are repeated 2000 times, using a new randomly generated node spatial distribution instance for each procedure and set of conditions. I_{min} and I_{max} parameters are set to 8 ms and 2.3 hours, respectively, which are the default values established by the RPL specification [3].

B. Synchronous network simulation results vs analytical model results

In this subsection we compare the number of messages transmitted in a given interval, as predicted by the analytical model presented in the previous section, with results obtained by simulation, for values of both k and average node degree between 1 and 15, respectively. The range of values considered for k includes the default value determined by the RPL specification (i.e. k=10) [3]. This subsection focuses on the synchronous mode, since the model assumes a synchronous network. The slightly different performance of an asynchronous network is discussed in the next subsection. Fig. 2 compares simulation results (which show the average results obtained from 10000 intervals for each set of parameter values) with analytical results. The largest 95% confidence interval size obtained for the simulation results is 4.73% of the corresponding average result. For the sake of clarity, Fig. 2 does not show the confidence intervals. As shown in Fig. 2, the analytical model approximates well the results obtained by simulation. When the average node degree is comparable to or smaller than k, the model is highly accurate. As the difference between the average node degree and k increases, the model accuracy decreases slightly. This occurs because the model assumes in (7) that message transmissions of node x neighbors are independent events. However, a transmission done by a node x neighbor within the coverage range of another node x neighbor will lead to an increment of the counter c and a decrease in the transmission probability of the latter.

$$\begin{aligned} P(tx) = &\sum_{i=0}^{k-1} \left(\binom{N-1}{i} \cdot (\frac{a}{A})^i \cdot (1-\frac{a}{A})^{N-1-i} \right) + \sum_{i=k}^{N-1} \left((\frac{k}{i+1}) \cdot \left(\binom{N-1}{i} \cdot (\frac{a}{A})^i \cdot (1-\frac{a}{A})^{N-1-i} \right) \right) \\ &+ \sum_{i=k}^{N-1} \left[(1-\frac{k}{i+1}) \cdot \left(\binom{N-1}{i} \cdot (\frac{a}{A})^i \cdot (1-\frac{a}{A})^{N-1-i} \right) \cdot \sum_{j=0}^{k-1} \left(\binom{i}{j} \cdot (P(tx))^j \cdot (1-P(tx))^{i-j} \right) \right] \end{aligned} \tag{8}$$

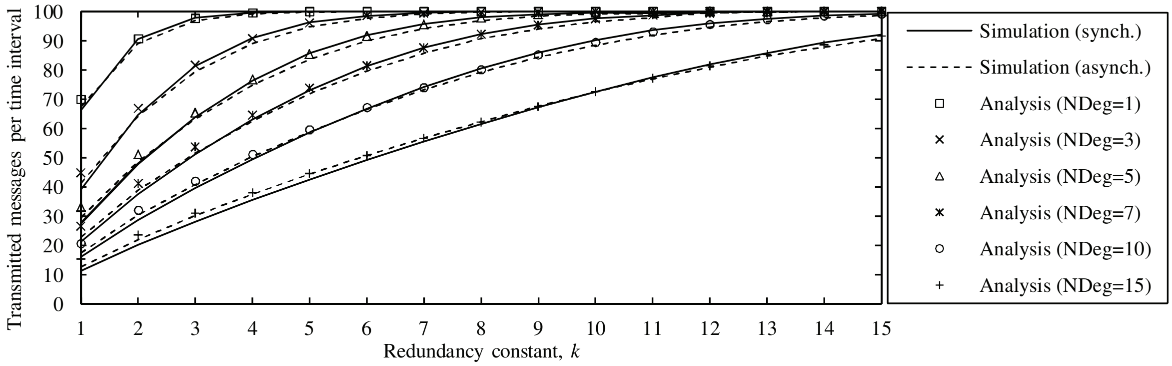

Fig. 2. Number of messages transmitted per time interval in a network composed of 100 nodes for different values of the redundancy constant, k, and different average node degrees, denoted $NDeg$: analysis (symbols) vs simulation (lines).

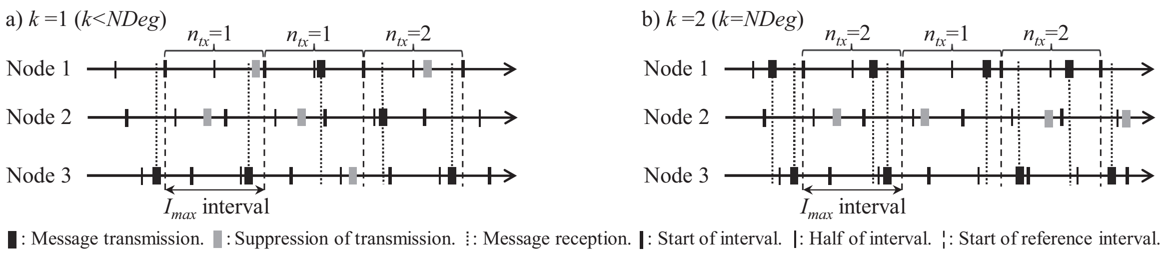

Fig. 3. Example of Trickle operation in an asynchronous, steady-state network composed of three nodes where all nodes can hear each other's transmissions: a) the value of k is set to 1 (i.e. k is smaller than $NDeg$); b) k is set to 2 (i.e. k is equal to $NDeg$). The total number of transmissions per interval is denoted n_{tx}.

C. Synchronous vs asynchronous modes

As can be seen in Fig. 2, for low values of k in comparison with the average node degree, the asynchronous mode yields a slightly greater number of message transmissions than the synchronous one. Otherwise, an asynchronous network leads to a message count that is equal to or slightly smaller than that of a synchronous network. We next explain the reasons for the different performance of these two types of network modes.

In an asynchronous network, when the value of k is small, results are influenced by the non-negligible probability that more than k transmissions will occur within the coverage range of a node during an interval. This happens because the start times of the nodes' intervals do not coincide. For example, Fig. 3.a) depicts the behavior of a network composed of three nodes, all of which can listen to each other, whereby k=1. As a consequence of the asynchronous interval start times of each node, 2 (i.e. more than k) transmissions occur in the last interval. As the average node degree grows in comparison with k, this phenomenon is more likely to occur. However, the same network in synchronous mode (e.g. nodes 1, 2 and 3 in Fig. 1) would yield k transmissions per interval.

On the other hand, in an asynchronous network, when k is not small in terms of the average node degree, results are affected by the fact that less than k transmissions can occur within the coverage range of a node during an interval. This also happens as a consequence of the network asynchronicity. Fig. 3.b) shows an example of the behavior of the same network referred to in Fig. 3.a), albeit with k set to 2 (i.e. k is equal to the number of neighbors of all the nodes). In Fig. 3.b), only one message is transmitted within the second interval. In fact, Node 2 suppresses its transmission in that interval since it has already heard k messages in its own interval. On the other hand, Node 3's next transmission is scheduled out of the second interval, while Node 1 is allowed to carry out its transmission. In the second interval, since there are no further neighbors that can transmit, the total number of transmissions is finally smaller than k. (Note that as the node degree increases in comparison with k, more nodes have the opportunity of transmitting.) Remarkably, the same network in synchronous mode would yield k transmissions per interval.

V. Conclusion

In this letter we present an analytical model for predicting the number of transmitted messages in a Trickle-based steady-state, static WSN, as a function of the redundancy constant and the average node degree. We validate the model by simulation for synchronous and asynchronous networks.

References

[1] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proc. 2004 NSDI*.

[2] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm," RFC 6206, Mar. 2011.

[3] T. Winter (editor) *et al.*, "RPL: IPv6 Routing Protocol for low power and lossy networks," RFC 6550, Mar. 2012.

[4] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proc. 2002 ACM MobiHoc*.

[5] O. Gnawali *et al.*, "Collection Tree Protocol," in *Proc. 2009 SenSys*.