Abstract

Wireless sensor networks require communication protocols for efficiently propagating data in a distributed fashion. The Trickle algorithm is a popular protocol serving as the basis for many of the current standard communication protocols. In this paper we develop a mathematical model describing how Trickle propagates new data across a network consisting of nodes placed on a line. The model is analyzed and asymptotic results on the hop count and end-to-end delay distributions in terms of the Trickle parameters and network density are given. Additionally, we show that by only a small modification of the Trickle algorithm the expected end-to-end delay can be greatly decreased. Lastly, we demonstrate how one can derive the exact hop count and end-to-end delay distributions for small network sizes.

Keywords: Analytical model, Markov renewal process, wireless communication, gossip protocol, end-to-end delay, Trickle algorithm

2010 MSC: 60K20, 90B18

1. Introduction

In recent years wireless sensor networks have been quickly growing in popularity. In these networks inexpensive autonomous sensor units, called nodes, gather data, which they exchange with each other through wireless transmissions. Wireless sensor networks have various applications, for example in health care, area- and industrial monitoring [1].

Typically, these networks are communication, computation, memory and energy constrained and therefore require appropriate distributed communication protocols. The requirements for a good communication protocol are threefold. First, it should be able to quickly disseminate and collect data within the network. Second, this should be done as efficiently as possible, meaning that the number of transmissions in the network should be as low as possible. Third, communication should be reliable; new
data should eventually be received by all the nodes in the network. Several communication protocols have been proposed in recent years for this purpose, see for example [4, 7, 17, 21, 23].

In [17] the Trickle algorithm has been proposed in order to effectively and efficiently distribute and maintain information in such networks. Trickle relies on a “polite gossip” policy to quickly propagate updates, while minimizing the number of redundant transmissions. Because of its broad applicability, Trickle has been documented in its own IETF RFC 6206 [16]. Moreover, it has been standardized as part of the IPv6 Routing Protocol for Low power and lossy networks [22] and the Multicast Protocol for Low power and lossy networks [9].

Because the Trickle algorithm has become a standard and is being widely used, it is crucial to understand how its parameters affect QoS measures like energy usage and end-to-end delay. However, not much work has yet been done in this regard. Most of the papers that evaluate Trickle try to give guidelines on how to set the Trickle parameters using simulations [6, 13, 15, 17].

The goal of this paper is to develop and analyze an analytical model describing how Trickle disseminates updates throughout a network. Our results serve as a first step towards understanding how Trickle’s parameters influence its performance; however, the main focus of this paper will be on the mathematical analysis. Additionally, our models are relevant for the analysis of protocols that build upon Trickle, such as Deluge [10] and Melete [23].

1.1. Key contributions of the paper

As key contributions of this paper, we thoroughly analyze a Markov renewal process which models a Trickle propagation event in networks of nodes arranged on a line. We show how the hop count and end-to-end delay distributions depend on the Trickle parameters and network density, as the size of the network grows large. These insights help us to better understand the impact of Trickle’s parameters on its performance and how to tune these parameters. Furthermore, we show that by a simple modification of the Trickle algorithm, the expected end-to-end delay can be significantly decreased. Finally, we show how to calculate the generating functions of the hop count and end-to-end delay distributions for any network size.

1.2. Related work

Some analytical results concerning the message count of the Trickle algorithm are provided in [14, 15, 19]. First, in [15] qualitative results are provided on the scalability of the algorithm. Specifically, it is conjectured that in single-hop networks the number of transmissions per time interval is bounded regardless of the network size, if a listen-only period is used. When no listen-only period is used, the message count scales as $O(\sqrt{n})$, where $n$ is the size of the network. These claims are proven in [19] and tight upper bounds and accompanying growth factors for the message count are provided. Additionally,
inter-transmission time distributions for large network sizes are derived and the concept of a listen-only period is generalized. Moreover, approximations for the message count in multi-hop networks are provided. Lastly, in [14], a different, slightly more accurate, approximation is given for the message count for the multi-cell network with specific parameter settings.

Analytic results on the speed at which the Trickle algorithm can propagate new data throughout a network are even fewer. In [3] the authors provide a method for deriving the Laplace transform of the distribution function of the end-to-end delay for any network topology. However, the method is computationally involved, limiting its practical use, and does not provide much insight. In this paper we provide more easily computable expressions for the Laplace transforms of the hop count and end-to-end delay in a line network.

1.3. Organization of the paper

The remainder of this paper is organized as follows. In Section 2 we give a detailed description of the Trickle algorithm and introduce a slight modification of the algorithm. In Section 3 we then analyze how fast the algorithm can propagate updates across a network consisting of nodes placed on a line. Additionally, we derive the limiting distributions for the hop count and end-to-end delay as the size of the network grows large and show that with our modified algorithm the expected end-to-end delay can be significantly decreased. This is followed in Section 4 by a derivation of the generating functions for the hop count and end-to-end delay for any network size. Finally, we compare our results with simulations and present our conclusions.

2. The Trickle Algorithm

We now provide a slightly modified brief description of the Trickle algorithm, as it is given in [19] (see also [17] for the original description). The Trickle algorithm has two main goals. First, whenever a new update enters the network, it must be propagated quickly. Secondly, when there are no updates, communication overhead has to be kept to a minimum.

The Trickle algorithm achieves this by using a “polite gossip” policy. Nodes divide time into intervals of varying length. During each interval a node will broadcast its current information, if it has not heard other nodes transmit the same information during that interval, in order to check if its information is up to date. If it has recently heard another node transmit the same information it currently has, it will stay quiet, assuming there is no new information to be received. Additionally, it will increase the length of its intervals, decreasing its broadcasting rate. Whenever a node receives an update or hears old information, it will reduce its interval size, increasing its broadcasting rate, in order to quickly resolve the inconsistency. This way inconsistencies are detected and resolved fast, while keeping the number of transmissions low.
2.1. Algorithm Description

The algorithm has four parameters:

- A threshold value $k$, called the redundancy constant.
- The maximum interval size $\tau_h$.
- The minimum interval size $\tau_l$.
- The listen-only parameter $\eta \leq \frac{1}{2}$, defining the size of a listen-only period.

Furthermore, each node in the network has its own timer and keeps track of three variables:

- The current interval size $\tau$.
- A counter $c$, counting the number of messages heard during an interval.
- A broadcasting time $t$ during the current interval.

The behavior of each node is described by the following set of rules:

1. At the start of a new interval a node resets its timer and counter $c$ and, if $\tau = \tau_l$, sets $t$ to a value in $[\eta \tau, \tau]$ at random, otherwise in $[\frac{1}{2} \tau, \tau]$ at random.
2. When a node hears a message that is consistent with the information it has, it increments $c$ by 1.
3. When a node’s timer hits time $t$, the node broadcasts its message, if $c < k$.
4. When a node’s timer hits time $\tau$, it doubles its interval size $\tau$ up to $\tau_h$ and starts a new interval.
5. When a node hears a message that is inconsistent with its own information, then if $\tau > \tau_l$ it sets $\tau$ to $\tau_l$ and starts a new interval, otherwise it does nothing.

Here, rule 1 is slightly different compared to the original description of Trickle. The authors of [17] propose to always use a listen-only period of half an interval, i.e., $\eta = \frac{1}{2}$, because of the so-called short-listen problem, which is discussed in the same paper. When no listen-only period is used, i.e. nodes always pick $t$ in $[0, \tau]$, sometimes nodes will broadcast soon after the beginning of their interval, listening for only a short time, before anyone else has a chance to speak up. If we have a perfectly synchronized network this does not give a problem, because the first $k$ transmissions will simply suppress all the other broadcasts during that interval. However, in an unsynchronized network, if a node has a short listening period, it might broadcast just before another node starts its interval and that node possibly also has a short listening period. This possibly leads to a lot of redundant messages and is referred to as the short-listen problem.

However, always having a listen-only period greatly affects propagation speed. When a listen-only period of $\tau/2$ is used, newly updated nodes will always have to wait for a period of at least $\tau_l/2$, before
attempting to propagate the received update. Consequently, in an \( m \)-hop network, the end-to-end delay is at least \( m\tau_l/2 \).

Hence, as is also argued in [19], on the one hand long listen-only periods reduce the number of redundant transmissions, but on the other hand short listen-only periods increase propagation speed. Rule 1 of the Trickle algorithm as we propose it, tries to get the best of both worlds. When nodes have just received an update and reset \( \tau \) to \( \tau_l \), they are allowed to be impatient and transmit after listening for only \( \eta \tau \) time units. Probably, they are at the front of the propagation wave and have neighbors that are not yet up to date. When \( \tau > \tau_l \), the wave front probably has passed, and nodes should first listen to what their neighbors have to say, before deciding whether to broadcast or not. Finally, note that by setting \( \eta = \frac{1}{2} \) we have the Trickle algorithm as it was originally defined.

3. Propagation model

In this section we develop and analyze a model describing how fast the Trickle algorithm can propagate updates in a network consisting of nodes placed on a line. This is a common network topology in many applications, for example in intelligent street lighting. Moreover, simulation experiments suggest that Trickle’s qualitative behavior in this scenario is comparable with its performance in different topologies. We will analyze the asymptotic distributions of the hop count and end-to-end delay as the network grows large. We will concentrate on the commonly used setting \( k = 1 \), which permits a detailed mathematical analysis. However, we note that simulations suggest performance for other \( k \) is qualitatively the same as for \( k = 1 \) [6, 18]. Increasing \( k \) is generally used in order to deal with lossy transmissions. See also [13] for a simulation study on the effect of the redundancy constant \( k \) on propagation speed.

We first introduce the model and some notation. Assume we have \( n + 1 \) nodes arranged on a line and each node is separated by a distance of 1 from its neighbors. Furthermore, assume all the nodes are perfect receivers and transmitters, i.e. transmissions are instantaneous and there is no packet loss. We label the nodes 0, 1, \ldots, \( n \) from left to right. Without loss of generality we assume \( \tau_l = 1 \). Updates will be injected into the network at node 0. Nodes have a fixed transmission range \( R \), which means that when a node sends a message, only nodes within a distance \( R \) of the broadcaster will receive the message. Finally, assume initially that all the nodes have \( \tau = \tau_l \), and at time 0 an update is injected at node 0, which it starts to propagate. Let us denote the time node \( n \) gets updated by \( T(n) \), which is called the end-to-end delay, and the number of transmissions needed to reach node \( n \) by \( H(n) \), which is called the hop count.

Observe that because node 0 gets updated at time 0, this node will broadcast the update somewhere in the interval \( [\eta, 1] \), updating nodes 1 to \( R \). The newly updated nodes will reset their intervals, synchronize and set \( \tau = \tau_l = 1 \). Node 0 will double its interval length after its interval ends and will
have a listen-only period of length $\tau_i$ in its next interval. As a result, one of the newly updated nodes will be the next node to transmit. When node 1 is the first node to broadcast, it will only update node $R + 1$, which will be the next broadcaster. When node $R$ is the next node to broadcast, it will update nodes $R + 1$ to $2R$. After this step, nodes 1 to $R$ will double their interval length and the next transmission will again be done by one of the newly updated nodes. Let us formalize this process.

Let $U_m$ be the number of nodes that are updated by the $m$'th broadcast and let $U_0 = 1$. Then we can write

$$p_{ij} = \mathbb{P}[U_{m+1} = j \mid U_m = i] = \begin{cases} \frac{1}{i}, & R - i < j \leq R, \\ 0, & \text{otherwise}. \end{cases}$$

Hence, $\{U_i\}_{i=0}^\infty$ forms a Markov chain with states $\{1, \ldots, R\}$ and transition matrix $P = [p_{ij}]$, which allows us to analyze the hop count of a propagation event. Calculating its steady-state probability vector $\pi$ we find

$$\pi = \frac{2}{R(R + 1)}(1, \ldots, R).$$

Moreover, the expected number of nodes that get updated by each hop in steady state is given by

$$\mu_U = \mathbb{E}[U] = \lim_{i \to \infty} \mathbb{E}[U_i] = \sum_{j=1}^{R} j\pi_j = \frac{1}{3}(2R + 1).$$

The inter-transmission time between hops depends on the number of newly updated nodes $U_i$. Let $T_i$ be the time of the $i$'th transmission and $\theta_i = T_i - T_{i-1}$ be the time between the $i - 1$'th and $i$'th transmission and let $T_0 = 0$. We then know that the time $\theta_{i+1}$ is the minimum of $U_i$ Trickle timers. Then, $\theta_{i+1} \sim \eta + (1 - \eta)\beta[1, U_i]$, since the minimum of $m$ uniform random variables follows a $\beta(1, m)$ distribution. More precisely

$$\mathbb{P}[\theta_{i+1} \leq t \mid U_i = u] = \begin{cases} 0, & t < \eta, \\ 1 - \left(\frac{1-t}{1-\eta}\right)^u, & \eta \leq t \leq 1, \\ 1, & \text{otherwise}. \end{cases}$$

Hence the expected time between transitions in steady state is given by

$$\mu_\theta = \mathbb{E}[\theta] = \lim_{i \to \infty} \mathbb{E}[\theta_i] = \sum_{j=1}^{R} \pi_j \left(\eta + \frac{1 - \eta}{j + 1}\right) = \eta + 2(1 - \eta)\left(\frac{R + 1 - \sum_{j=1}^{R+1} \frac{1}{j}}{R(R + 1)} \right).$$

The process $(U_i, T_i)_{i=0}^\infty$ is called a Markov renewal process, see [3]. We will now analyze this Markov renewal process to gain insight in the propagation speed of the Trickle algorithm.

First note that the hop count $H^{(n)}$ can be written as

$$H^{(n)} = \min \left\{ m : \sum_{i=1}^{m} U_i \geq n \right\}. $$
Also note that $H^{(n)}$ is a stopping time with respect to the Markov chain $\{U_i\}_{i=0}^{\infty}$. Furthermore, we can write the end-to-end delay $T^{(n)}$ as follows:

$$T^{(n)} = T_{H^{(n)}} = \sum_{i=1}^{H^{(n)}} \theta_i. \quad (7)$$

We will analyze the behavior of the Markov renewal process for large $n$. For simplicity, we shall assume stationarity of the underlying Markov chain in some of the arguments and no longer assume $U_0 = 1$. However, note that the asymptotic results for large $n$ also hold for the case $U_0 = 1$, since we have a finite-state Markov chain, which converges geometrically fast to its steady-state distribution. Assuming stationarity of the Markov chain, we have by Wald’s equation

$$E\left[ H^{(n)} \right] = E\left[ \sum_{i=1}^{H^{(n)}} U_i \right] / \mu_U. \quad (8)$$

Furthermore, since $n \leq E\left[ \sum_{i=1}^{H^{(n)}} U_i \right] \leq n + R - 1$, we find

**Proposition 1.** For $k = 1$, $\eta \in [0, 1]$ and $R \in \mathbb{N}^+$,

$$\lim_{n \to \infty} \frac{1}{n} E\left[ H^{(n)} \right] = \frac{1}{\mu_U} = \frac{3}{2R + 1}. \quad (8)$$

Hence, since $1/R$ can be seen as a measure for the density of the network, we find that the hop count decreases linearly with the network density.

Now using (7) and again applying Wald’s equation, we conclude

**Proposition 2.** For $k = 1$, $\eta \in [0, 1]$ and $R \in \mathbb{N}^+$,

$$\lim_{n \to \infty} \frac{1}{n} E\left[ T^{(n)} \right] = \frac{\mu_U}{\mu_U} = \frac{3}{2R + 1} \left( \eta + 2(1 - \eta) \frac{R + 1 - \sum_{j=1}^{R+1} \frac{1}{j}}{R(R + 1)} \right). \quad (9)$$

Proposition 2 reveals the impact of a listen-only period on the end-to-end delay. If $\eta > 0$, the end-to-end delay decreases linearly with the density of network, while for $\eta = 0$ it decreases quadratically. Hence, in large networks the algorithm can benefit greatly from an appropriate choice for $\eta$.

In addition to a law of large numbers for the hop count and end-to-end delay, we now provide results concerning their limiting distribution. First of all, we provide results on the asymptotics of the variance of $H^{(n)}$ and $T^{(n)}$.

**Theorem 1.** Let $H^{(n)}$ be as defined by (6). Then

$$\lim_{n \to \infty} \frac{\text{Var}\left[ H^{(n)} \right]}{n} = \sigma_H^2 \frac{\mu_U^2}{\mu_U} = \frac{R^2 + R - 2}{16R^3 + 24R^2 + 12R + 2}, \quad (10)$$

where

$$\gamma_U^2 = \lim_{m \to \infty} \frac{1}{m} \text{Var}\left[ \sum_{i=0}^{m} U_i \right] = \text{Var}[U_0] + 2 \sum_{j=1}^{\infty} \text{Cov}[U_0, U_j]. \quad (11)$$
Proof. A result for first passage times of Markov renewal processes in \([8]\) implies (see Theorem 3.4)

\[
\text{Var} \left[ H^{(n)} \right] \sim n \gamma^{2}_U / \mu^3_U, \text{ as } n \to \infty.
\]

Therefore we need to show that \( \gamma^{2}_U = \frac{1}{18}(R^2 + R - 2) \). To simplify the analysis we will assume stationarity of the Markov chain \( \{U_i\}_{i=0}^{\infty} \), but again note that the final result will also hold for the non-stationary case \( U_0 = 1 \). We show by induction that \( \text{Cov}[U_0, U_j] = (-\frac{1}{2})^j \frac{1}{18}(R^2 + R - 2) \). First note that

\[
\text{Cov}[U_0, U_j] = \mathbb{E}[U_0 U_j] - \mu^2_0 = \mathbb{E}[U_0 U_j] - \frac{1}{9}(2R + 1)^2.
\]

For \( j = 0 \) we find

\[
\mathbb{E}[U_0^2] = \sum_{k=1}^{R} \pi_k k^2 = \frac{2}{R(R+1)} \sum_{k=1}^{R} k^3 = \frac{1}{2}R(1+R).
\]

Hence \( \text{Cov}[U_0, U_0] = \frac{1}{2}R(1+R) - \frac{1}{9}(2R + 1)^2 = \frac{1}{18}(R^2 + R - 2) \), which is our induction basis. Now let \( p^{(k)}_{ij} \) be the probability that starting from state \( i \) the Markov chain is in state \( j \) after \( k \) steps. Then we can write

\[
\mathbb{E}[U_0 U_j] = \sum_{k=1}^{R} \pi_k \left( \sum_{l=1}^{R} p^{(j)}_{kl} \right) = \sum_{k=1}^{R} \pi_k \left( \sum_{m=1}^{R} p^{(j-1)}_{km} \left( \sum_{l=1}^{R} \pi_l k l \right) \right)
\]

\[
= \sum_{k=1}^{R} \pi_k \left( \sum_{m=1}^{R} p^{(j-1)}_{km} \left( \sum_{l=R-m+1}^{R} m k l \right) \right) = \sum_{k=1}^{R} \pi_k \left( \sum_{m=1}^{R} p^{(j-1)}_{km} \frac{1}{2}(2R + 1 - m)k \right)
\]

\[
= \frac{1}{2}(2R + 1) \sum_{k=1}^{R} \pi_k k - \frac{1}{2} \sum_{k=1}^{R} \pi_k \left( \sum_{m=1}^{R} p^{(j-1)}_{km} m k \right) = \frac{1}{6}(2R + 1)^2 - \frac{1}{2} \mathbb{E}[U_0 U_{j-1}].
\]

Consequently, we find

\[
\text{Cov}[U_0, U_j] = -\frac{1}{2} \mathbb{E}[U_0 U_{j-1}] + \frac{1}{18}(2R + 1)^2 = \left(-\frac{1}{2}\right) \text{Cov}[U_0, U_{j-1}].
\]

Using this result it is easy to see that

\[
\gamma^{2}_U = \frac{1}{18}(R^2 + R - 2) + 2 \sum_{j=1}^{\infty} \left(-\frac{1}{2}\right)^j \frac{1}{18}(R^2 + R - 2) = \frac{1}{54}(R^2 + R - 2),
\]

finishing the proof. \( \square \)

Similarly, we have the following asymptotic result for the variance of \( T^{(n)} \).

**Theorem 2.** Let \( T^{(n)} \) be as defined by (7). Then

\[
\lim_{n \to \infty} \frac{\text{Var} \left[ T^{(n)} \right] \mu_U}{n} = \sigma_T^2 = \frac{\gamma^{2}_U}{\mu_U},
\]

where \( \gamma_U \) is as defined in (11).

\[
\gamma_T^2 = \lim_{m \to \infty} \frac{1}{m} \text{Var} \left[ \left( \sum_{i=0}^{m} U_i \right) \mu_\theta - T_{m+1} \mu_U \right],
\]

(13)
\[
\Delta = (1 - \eta) \frac{(4R + 8) \left( \sum_{j=1}^{R+1} \frac{1}{j} \right) - (R^2 + 9R + 8)}{9R^2 + 9R},
\]

and

\[
\gamma_0^2 = \text{Var}[\theta_1] + 2\pi MZM \mathbf{1} - 2\mu_0^2,
\]

with \( Z = (I - P + 1\pi)^{-1} \) the fundamental matrix and \( M = \left[ p_{ij} \left( \eta + \frac{1 - \eta}{\tau + 1} \right) \right] \).

**Proof.** A result for stopped functionals of Markov renewal processes implies (see [2], Theorem 2)

\[
\text{Var}\left[T^{(n)}\right] \sim n\gamma_T^2/\mu_T^2, \quad \text{as } n \to \infty.
\]

Now, rewriting (13) we have

\[
\gamma_T^2 = \text{Var}[U_0\mu_\theta - \theta_1\mu_U] + 2 \sum_{i=1}^{\infty} \text{Cov}[U_0\mu_\theta - \theta_1\mu_U, U_i\mu_\theta - \theta_{i+1}\mu_U] = \mu_\theta^2\gamma_T^2 + \mu_U\gamma_\theta^2 - 2\mu_U\mu_\theta\Delta.
\]

Here \( \gamma_\theta \) and \( \Delta \) are defined as

\[
\gamma_\theta^2 = \lim_{m \to \infty} \frac{1}{m} \text{Var}[T_{m+1}] = \text{Var}[\theta_1] + 2 \sum_{j=1}^{\infty} \text{Cov}[\theta_1, \theta_{j+1}],
\]

\[
\Delta = \text{Cov}[\theta_1, U_0] + \sum_{j=1}^{\infty} \text{Cov}[U_0, \theta_{j+1}] + \sum_{j=1}^{\infty} \text{Cov}[\theta_1, U_j].
\]

For simplicity of the analysis, assume again stationarity of the Markov chain \( \{U_i\}_{i=0}^{\infty} \). An expression for \( \gamma_\theta^2 \) in terms of the matrix \( M \) and the fundamental matrix \( Z = (I - P + 1\pi)^{-1} \) is given in [11], that is

\[
\gamma_\theta^2 = \text{Var}[\theta_1] + 2\pi MZM \mathbf{1} - 2\mu_\theta^2.
\]

Here,

\[
\text{Var}[\theta_1] = 4(1 - \eta)^2 \left( \frac{6 + R}{8 + 4R} - \frac{2 + R}{2R} - \frac{\sum_{j=1}^{R+1} \frac{1}{j}}{R(1 + R)} \right).
\]

Finally, analogous to the proof of Theorem [1] we can show that \( \text{Cov}[\theta_1, U_j] = (-\frac{1}{2})^j \text{Cov}[\theta_1, U_0] \) and

\[
\text{Cov}[\theta_1, U_0] = (1 - \eta) \frac{(4R + 8) \left( \sum_{j=1}^{R+1} \frac{1}{j} \right) - (R^2 + 9R + 8)}{3R^2 + 3R}.
\]

Now, since \( \pi_ip_{ij} = \pi_jp_{ji} \) for all \( i \) and \( j \), the Markov chain \( U \) is reversible (see [12], Theorem 1.2). This implies \( (U_0, \theta_{j+1}) \sim (\theta_1, U_j) \) and hence we have \( \text{Cov}[U_0, \theta_{j+1}] = \text{Cov}[\theta_1, U_j] \). This then yields

\[
\Delta = \text{Cov}[\theta_1, U_0] + \sum_{j=1}^{\infty} \text{Cov}[U_0, \theta_{j+1}] + \sum_{j=1}^{\infty} \text{Cov}[\theta_1, U_j] = \text{Cov}[\theta_1, U_0] + 2 \sum_{j=1}^{\infty} \text{Cov}[\theta_1, U_j]
\]

\[
= (1 - \eta) \frac{(4R + 8) \left( \sum_{j=1}^{R+1} \frac{1}{j} \right) - (R^2 + 9R + 8)}{9R^2 + 9R}.
\]

\[\square\]
Finally using results from [2], we get the following results for the limiting distributions of $H^{(n)}$ and $T^{(n)}$ (see [2], Theorem 1).

**Theorem 3.** Let $H^{(n)}$ and $T^{(n)}$ be as defined in (6) and (7) respectively. Then

$$\frac{H^{(n)} - \mu_H n}{\sigma_H n^{1/2}} \xrightarrow{d} \mathcal{N}[0, 1]$$

and

$$\frac{T^{(n)} - \mu_T n}{\sigma_T n^{1/2}} \xrightarrow{d} \mathcal{N}[0, 1],$$

where $\sigma_H$ is as defined in (10) and $\sigma_T$ as defined in (12).

### 3.1. Simulation results

We now look at some simulation results on the end-to-end delay in a network of $n = 250$ nodes with $R = 4$ for different values of $\eta$. For each value of $\eta$ we run $10^5$ simulations. In Figure 1 we compare the obtained histogram for the end-to-end delay with the asymptotic result from Theorem 3. We find that the end-to-end delay distribution is approximated well by the normal distribution. Furthermore, setting $\eta = 0$ as opposed to $\eta = \frac{1}{2}$ more than halves the expected delay, while the variance does not seem to change significantly. Hence, setting $\eta = 0$ seems preferable to $\eta = \frac{1}{2}$.

![Simulation results](image1.png)

Figure 1: Density of $T^{(n)}$ for $n = 250$ and $R = 4$ for different $\eta$: analysis vs. simulation.

Finally, we investigate how the variance of the end-to-end delay depends on $\eta$ and $R$. For this we plot $\sigma_T^2$ as a function of $\eta$ for $R = 5$, $R = 10$ and $R = 30$ in Figure 2. We find that for $R = 5$ the variance is minimized for $\eta \approx 0.56$. For $R = 10$, the minimum is achieved at $\eta \approx 0.26$ and for $R = 30$ at $\eta = 0$. For small $R$, the variance of the hop count is small compared to the variance of inter-transmission times and hence it pays to increase $\eta$ in order to decrease the variance of inter-transmission times. For large $R$, the hop count has high variability, while inter-transmission times have small variance, hence it pays to decrease $\eta$ in order to decrease the impact of the hop count on the end-to-end delay variance. Therefore, for dense networks ($R$ large), setting $\eta = 0$ is always preferred to $\eta > 0$, since it minimizes both the expected value and variance of the propagation delay. For sparse networks ($R$ small), one can
reduce the variance by setting $\eta > 0$, however this greatly increases the expected delay, hence $\eta = 0$ is probably more desirable.

4. Hop count and end-to-end delay distribution

We now focus on deriving the probability generating function of $H^{(n)}$ and the moment generating function of $T^{(n)}$ for finite $n$. These allow us to calculate the exact moments of the hop count and end-to-end delay for small finite $n$, which is not covered by the asymptotic results for large $n$ in the previous section.

Let us denote by $M_X(s)$ the moment generating function of a continuous variable $X$ and by $G_Y[z]$ the probability generating function of a discrete random variable $Y$. Additionally, for a pair $(X_1, X_2)$ of discrete variables, write

$$G_{(X_1, X_2)}[z_1, z_2] = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} z_1^n z_2^m \mathbb{P}[X_1 = n, X_2 = m],$$

and for a pair $(X, Y)$, with $X$ a discrete and $Y$ a continuous random variable, write

$$G_{(X,Y)}[z, s] = \int_0^\infty \sum_{n=0}^{\infty} z^n e^{st} d\mathbb{P}[X = n, Y \leq t].$$

We first analyze the probability generating function of $H^{(n)}$.

4.1. Hop count

Let $A(m) = \sum_{i=1}^{m} U_i$, i.e. $A(m)$ is the total number of updated nodes after $m$ transmissions, not including node 0. Furthermore, let $\eta_{ij}$ be the number of transitions in the Markov chain $\{U_m\}_{m=0}^{\infty}$ between entering state $i$ and entering state $j$ for the first time. Denote by $A(\eta_{ij})$ the number of nodes updated during that time.
Theorem 4.

\[ \mathcal{H}[z_1, z_2] = \sum_{n=0}^{\infty} \left( \sum_{m=0}^{\infty} \mathbb{P}[H^{(n)} = m] z_1^m \right) z_2^n = \sum_{n=0}^{\infty} G_{H^{(n)}}[z_1] z_2^n \]

\[ \frac{(z_1 - 1)z_2}{1 - z_2} \left( 1 + \sum_{j=1}^{R} \frac{G(A(\eta_{j-1}, \eta_j))[z_1, z_2]}{1 - G(A(\eta_{j-1}, \eta_j))[z_1, z_2]} \right) + \frac{1}{1 - z_2}. \]

Remark. Note that the probability generating function for \( H^{(n)} \) can be obtained by differentiating \( \mathcal{H}[z_1, z_2] \):

\[ G_{H^{(n)}}[z] = \left. \frac{d^n}{dz_2^n} \mathcal{H}[z, z_2] \right|_{z_2 = 0}. \] (16)

Proof. In [20] results are provided which lead to explicit expressions for relevant Laplace transforms of general reward functions for Markov renewal and semi-Markov processes. The following proof closely follows the steps of their proof, but is slightly customized for the case at hand, simplifying the analysis.

First, we write

\[ \mathbb{P}[H^{(n+1)} > m] = \sum_{j=1}^{R} \mathbb{P}[A(m) \leq n, U_m = j \mid U_0 = 1]. \]

Now, let \( \eta_j^{(k)} \) be the time of the \( k \)'th entry into state \( j \) of the Markov chain \( U \), with \( \eta_j^{(0)} = 0 \), that is,

\[ \eta_j^{(k)} = \inf \{ i > \eta_j^{(k-1)} : U_i = j \}. \]

Then

\[ \mathbb{P}[H^{(n+1)} > m] = \mathbb{1}[m = 0] + \sum_{j=1}^{R} \sum_{k=1}^{\infty} \sum_{n=0}^{\infty} \mathbb{P} \left[ A \left( \eta_j^{(k)} \right) \leq n, \eta_j^{(k)} = m \mid U_0 = 1 \right]. \]

Substituting and using the fact that \( \frac{1}{1-z} \sum_{n=0}^{\infty} a_n z^n = \sum_{n=0}^{\infty} \sum_{m=0}^{n} a_m z^n \), we find

\[ \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \mathbb{P}[H^{(n+1)} > m] z_1^m z_2^n \]

\[ = \frac{1}{1 - z_2} + \sum_{m=1}^{\infty} \sum_{n=0}^{\infty} \sum_{j=1}^{R} \sum_{k=1}^{\infty} \mathbb{P} \left[ A \left( \eta_j^{(k)} \right) \leq n, \eta_j^{(k)} = m \mid U_0 = 1 \right] z_1^m z_2^n \]

\[ = \frac{1}{1 - z_2} + \frac{1}{1 - z_2} \sum_{m=1}^{\infty} \sum_{n=0}^{\infty} \sum_{j=1}^{R} \sum_{k=1}^{\infty} \mathbb{P} \left[ A \left( \eta_j^{(k)} \right) = n, \eta_j^{(k)} = m \mid U_0 = 1 \right] z_1^m z_2^n \]

\[ = \frac{1}{1 - z_2} \left( 1 + \sum_{j=1}^{R} \sum_{k=1}^{\infty} G(A(\eta_j^{(k)}), \eta_j^{(k)})[z_1, z_2] \right). \]

Since the consecutive entry times to a fixed state form a sequence of regeneration times, we obtain for \( k \geq 1 \) and \( j = 1 \)

\[ G(\mathbf{A}(\eta_{j}^{(k)}), \eta_{j}^{(k)})[z_1, z_2] = (G(\mathbf{A}(\eta_{j-1}, \eta_j))[z_1, z_2])^k, \]

and for \( j \neq 1 \) we get

\[ G(\mathbf{A}(\eta_{j}^{(k)}), \eta_{j}^{(k)})[z_1, z_2] = G(\mathbf{A}(\eta_{j-1}, \eta_j))[z_1, z_2] \left( G(\mathbf{A}(\eta_{j-1}, \eta_j))[z_1, z_2] \right)^{k-1}. \]
Therefore
\[
\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \mathbb{P}[H^{(n)} > m] z_1^m z_2^n = \frac{z_2}{1 - z_2} \left( 1 + \sum_{j=1}^{R} \sum_{k=1}^{\infty} G_{(A(n_{i,j}),\eta_{i,j})}[z_1, z_2] \left( G_{(A(n_{i,j}),\eta_{i,j})}[z_1, z_2] \right)^{k-1} \right)
\]
\[
= \frac{z_2}{1 - z_2} \left( 1 + \sum_{j=1}^{R} \frac{G_{(A(n_{i,j}),\eta_{i,j})}[z_1, z_2]}{1 - G_{(A(n_{i,j}),\eta_{i,j})}[z_1, z_2]} \right).
\]
Finally we obtain
\[
\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \mathbb{P}[H^{(n)} = m] z_1^m z_2^n = \frac{(z_1 - 1)z_2}{1 - z_2} \left( 1 + \sum_{j=1}^{R} \frac{G_{(A(n_{i,j}),\eta_{i,j})}[z_1, z_2]}{1 - G_{(A(n_{i,j}),\eta_{i,j})}[z_1, z_2]} \right) + \frac{1}{1 - z_2}.
\]

Note that the functions \(G_{(A(n_{i,j}),\eta_{i,j})}[z_1, z_2]\) can be derived by solving the following system of linear equations:
\[
G_{(A(n_{i,j}),\eta_{i,j})}[z_1, z_2] = \sum_{k \neq j} p_{ik} z_1^k z_2 G_{(A(n_{k,j}),\eta_{k,j})}[z_1, z_2] + p_{ij} z_1^i z_2, \text{ for all } i \text{ and } j.
\] (17)

5. End-to-end delay

Similarly, we will now consider the moment generating function for the end-to-end delay \(T^{(n)}\). Let \(B(t) = \sum_{i=1}^{m} U_i \) for \(t \in [T_m, T_{m+1})\). Then \(B(t)\) is the total number of updated nodes at time \(t\), not including node 0. Again let \(\eta_{ij}\) be the number of transitions in the Markov chain \(\{U_m\}_{m=0}^{\infty}\) between entering state \(i\) and entering state \(j\) for the first time and denote by \(B(T_{\eta_{ij}})\) the number of nodes updated during that time. Denote by \(\nu_j\) the random time the Markov chain stays in state \(j\) before transitioning.

**Theorem 5.**
\[
\mathbb{T}[z, s] = \sum_{n=0}^{\infty} \left( \int_{t=0}^{\infty} e^{zt} \mathbb{P}[T^{(n)} \leq t] dt \right) z^n = \sum_{n=0}^{\infty} M_{T^{(n)}}[s] z^n
\]
\[
= \frac{z}{z - 1} \left( 1 - M_{\nu_1}(s) + \sum_{j=1}^{R} (1 - M_{\nu_j}(s)) \frac{G_{(B(T_{\eta_{nj}}),\eta_{nj})}[z, s]}{1 - G_{(B(T_{\eta_{nj}}),\eta_{nj})}[z, s]} \right) + \frac{1}{1 - z}.
\]

**Remark.** Note that the moment generating function for \(T^{(n)}\) can be obtained by differentiating \(\mathbb{T}[z, s]\):
\[
M_{T^{(n)}}[s] = \frac{1}{n!} \frac{d^n}{dz^n} \mathbb{T}[z, s] \bigg|_{z=0}.
\] (18)

**Proof.** Again, our proof closely resembles the proof given in [20]. Let \(U(t) = U_m \) for \(t \in [T_m, T_{m+1})\). We write
\[
\mathbb{P} \left[ T^{(n+1)} > t \right] = \sum_{j=1}^{R} \mathbb{P} [B(t) \leq n, U(t) = j \mid U_0 = 1].
\]
Again let $\eta_j^{(k)}$ be the time of the $k$’th entry into state $j$ of the Markov chain $U$, with $\eta_j^{(0)} = 0$, that is,

$$\eta_j^{(k)} = \inf\{i > \eta_j^{(k-1)} : U_i = j\}.$$

Then

$$\mathbb{P}[T(n+1) > t] = \sum_{j=1}^{R} \sum_{k=0}^{\infty} \mathbb{P}[B \left( T_{\eta_j^{(k)}} \right) \leq n, T_{\eta_j^{(k)}} \leq t < T_{\eta_j^{(k)+1}}, U(t) = j \mid U(0) = 1].$$

For convenience we write

$$b_j^k(n, t) = \mathbb{P}[B \left( T_{\eta_j^{(k)}} \right) \leq n, T_{\eta_j^{(k)}} \leq t < T_{\eta_j^{(k)+1}}, U(t) = j \mid U(0) = 1].$$

Note first that for $k = 0$ we have

$$b_j^0(n, t) = \begin{cases} \mathbb{P}[\nu_1 > t], & \text{if } j = 1, \\ 0, & \text{otherwise}. \end{cases}$$

For $k \geq 1$ we can write,

$$b_j^k(n, t) = \int_0^\infty \sum_{l=0}^\infty \mathbb{P}[\nu_j > t - u] \mathbb{I}[n - l \geq 0] d\mathbb{P} \left[ B \left( T_{\eta_l^{(k)}} \right) = l, T_{\eta_l^{(k)}} \leq u \right],$$

which can be written as a convolution of a function $\phi(u, l)$ and a probability measure:

$$b_j^k(n, t) = \int_0^\infty \sum_{l=0}^\infty \phi_j(t - u, n - l) d\mathbb{P} \left[ B \left( T_{\eta_l^{(k)}} \right) = l, T_{\eta_l^{(k)}} \leq u \right],$$

where

$$\phi_j(u, l) = \mathbb{P}[\nu_j > u] \mathbb{I}[l \geq 0].$$

Consequently,

$$\int_0^\infty \sum_{n=0}^\infty b_j^k(n, t) z^n e^{zt} dt = \frac{1 - M_{\nu_j}(s)}{s(1 - z)} G_{(B(T_{\eta_j^{(k)}}, T_{\eta_j^{(k)}}))[z, s]}.$$

Hence,

$$\int_0^\infty \sum_{n=0}^\infty \mathbb{P}[T(n+1) > t] z^n e^{zt} dt = \int_0^\infty \sum_{n=0}^\infty \mathbb{P}[\nu_1 > t] z^n e^{zt} dt + \int_0^\infty \sum_{n=0}^\infty \sum_{j=1}^{R} \sum_{k=1}^{\infty} b_j^k(n, t) z^n e^{zt} dt$$

$$= \frac{1}{s(1 - z)} \left( 1 - M_{\nu_j}(s) + \sum_{j=1}^{R} \sum_{k=1}^{\infty} (1 - M_{\nu_j}(s)) G_{(B(T_{\eta_j^{(k)}}, T_{\eta_j^{(k)}}))[z, s]} \right).$$

Since the consecutive entry times to a fixed state form a sequence of regeneration times, we obtain for $k \geq 1$ and $j = 1$

$$G_{(B(T_{\eta_1^{(k)}}, T_{\eta_1^{(k)}}))[z, s]} = \left( G_{(B(T_{\eta_1^{(1)}}, T_{\eta_1^{(1)}}))[z, s]} \right)^k,$$

and for $j \neq 1$ we get

$$G_{(B(T_{\eta_j^{(k)}}, T_{\eta_j^{(k)}}))[z, s]} = G_{(B(T_{\eta_1^{(1)}}, T_{\eta_1^{(1)}}))[z, s]} \left( G_{(B(T_{\eta_j^{(1)}}, T_{\eta_j^{(1)}}))[z, s]} \right)^{k-1}.$$
Therefore,

\[
\int_0^\infty \sum_{n=0}^\infty P[T^{(n)} > t] z^n e^{st} dt = \frac{z}{s(1-z)} \left( 1 - M_\nu(s) + \sum_{j=1}^R \sum_{k=1}^\infty (1 - M_\nu(s)) G(B(T_{n,j}, T_{n,j})[z, s]) \left( G(B(T_{n,j}, T_{n,j})[z, s])^{k-1} \right) \right)
\]

\[
= \frac{z}{s(1-z)} \left( 1 - M_\nu(s) + \sum_{j=1}^R (1 - M_\nu(s)) \frac{G(B(T_{n,j}, T_{n,j})[z, s])}{1 - G(B(T_{n,j}, T_{n,j})[z, s])} \right).
\]

Finally we obtain

\[
\sum_{k=0}^\infty \left( \int_0^\infty e^{st} dP[T^{(k)} \leq t] \right) z^k = \frac{z}{z - 1} \left( 1 - M_\nu(s) + \sum_{j=1}^R (1 - M_\nu(s)) \frac{G(B(T_{n,j}, T_{n,j})[z, s])}{1 - G(B(T_{n,j}, T_{n,j})[z, s])} \right) + \frac{1}{1 - z}.
\]

Using (19) one can deduce that \( M_\nu(s) \) is the moment generating function of a \( \beta(1, j) \) random variable, which can be expressed in terms of the incomplete gamma function \( \Gamma[s, x] \) as follows

\[
M_\nu(s) = j! e^s \left( \frac{1}{s(1-\eta)} \right)^j \left( 1 - \frac{\Gamma[j, s(1-\eta)]}{(j-1)!} \right),
\]

(19)

Furthermore, analogous to Equation (17), the functions \( G(B(T_{n,i}, T_{n,j})[z, s]) \) can be derived by solving the following system of equations:

\[
G(B(T_{n,i}, T_{n,j})[z, s]) = \sum_{k \neq j} p_{ik} z^k M_\nu[s] G(B(T_{n,k}, T_{n,k})[z, s]) + p_{ij} z^j M_\nu[z], \text{ for all } i \text{ and } j.
\]

(20)

As illustrating examples, in Figure 3 we have plotted the density functions of \( H^{(20)} \) and \( T^{(20)} \) for \( R = 4 \) and both \( \eta = 0 \) and \( \eta = \frac{1}{2} \) obtained by inverting Equations (16) and (18) using Mathematica. Note that the hop count distribution is the same for both settings.

![Figure 3: Hop count and end-to-end delay density for \( n = 20 \) and \( R = 4 \).](image-url)
6. Conclusion

In this paper, we presented a generalized version of the Trickle algorithm with a parameter $\eta$, which allows us to set the length of a listen-only period for newly updated nodes. We show that this parameter can greatly increase the speed at which the Trickle algorithm can disseminate data, while retaining scalability.

First, we provided an analysis of the hop count and end-to-end delay distribution for line networks consisting of $n$ nodes. We derived formulas for the mean and variance of the hop count and end-to-end delay as a function of $R$, $n$ and $\eta$, giving insight into the performance of the Trickle algorithm. Additionally, we showed that both distributions converge to a normal distribution as $n$ goes to infinity.

Secondly, we demonstrated how to derive explicit expressions for the probability and moment generating functions of the hop count and end-to-end delay. As was shown, these functions can be used to determine the respective density functions for small network sizes explicitly.

From our analysis we can conclude that the generalized version of Trickle as presented in this paper with $\eta = 0$ allows for better performance, compared to the original description of Trickle. It greatly decreases end-to-end delay, while having only a small effect on its variability and the energy consumption in the network.

Finally, we note that our analysis is only a first step to a complete understanding of Trickle's propagation performance. For example, the impact of network topology and MAC-layer interactions on the performance remain as interesting topics.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. Computer Networks, 38(4):393–422, March 2002.

[2] G. Alsmeyer and A. Gut. Limit theorems for stopped functionals of Markov renewal processes. Annals of the Institute of Statistical Mathematics, 51(2):369–382, 1999.

[3] M. Becker, K. Kuladinithi, and C. Görg. Modelling and simulating the Trickle algorithm. In Proceedings of MONAMI, pages 135–144. Springer, 2011.

[4] Y. Busnel, M. Bertier, E. Fleury, and A.-M. Kermarrec. GCP: gossip-based code propagation for large-scale mobile wireless sensor networks. In Proceedings of the 1st International Conference on Autonomic Computing and Communication Systems, Autonomics '07, pages 11:1–11:5, 2007.

[5] E. Çinlar. Markov renewal theory. Advances in Applied Probability, 1(2):123–187, 1969.
[6] T. Clausen, A. C. de Verdiere, and J. Yi. Performance analysis of Trickle as a flooding mechanism. In *Proceedings of the 15th IEEE International Conference on Communication Technology (ICCT)*, pages 565–572, Nov 2013.

[7] W. Dong, Y. Liu, X. Wu, L. Gu, and C. Chen. Elon: enabling efficient and long-term reprogramming for wireless sensor networks. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS ’10, pages 49–60, 2010.

[8] C.-D. Fuh and T. L. Lai. Asymptotic expansions in multidimensional Markov renewal theory and first passage times for Markov random walks. *Advances in Applied Probability*, 33(3):652–673, 2001.

[9] J. Hui and R. Kelsey. Multicast Protocol for Low power and Lossy Networks (MPL). *IETF, Internet-Draft draft-ietf-roll-trickle-mcast-07.txt*, February 2014.

[10] J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys ’04, pages 81–94, 2004.

[11] J. Keilson and D. G. M. Wishart. Addenda to processes defined on a finite Markov chain. *Mathematical Proceedings of the Cambridge Philosophical Society*, 63:187–193, 1967.

[12] F. P. Kelly. *Reversibility and Stochastic Networks*. Cambridge University Press, New York, 2011.

[13] H. Kermajani and C. Gomez. On the network convergence process in RPL over IEEE 802.15.4 multihop networks: Improvement and trade-offs. *Sensors*, 14(7):11993–12022, 2014.

[14] H. Kermajani, C. Gomez, and M. H. Arshad. Modeling the message count of the Trickle algorithm in a steady-state, static wireless sensor network. *Communications Letters, IEEE*, 16(12):1960–1963, December 2012.

[15] P. Levis, E. Brewer, D. Culler, D. Gay, S. Madden, N. Patel, J. Polastre, S. Shenker, R. Szewczyk, and A. Woo. The emergence of a networking primitive in wireless sensor networks. *Communications of the ACM*, 51(7):99–106, July 2008.

[16] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko. The Trickle algorithm. *Internet RFC 6206*, March 2011.

[17] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation*, pages 15–28, 2004.
[18] T. M. M. Meyfroyt. Modeling and analyzing the Trickle algorithm. Master’s thesis, Eindhoven University of Technology, Eindhoven, The Netherlands. http://alexandria.tue.nl/extra1/afstversl/wsk-i/meyfroyt2013.pdf, August 2013.

[19] T. M. M. Meyfroyt, S. C. Borst, O. J. Boxma, and D. Denteneer. Data dissemination performance in large-scale sensor networks. *SIGMETRICS Performance Evaluation Review*, 42(1):395–406, June 2014.

[20] V. T. Stefanov. Exact distributions for reward functions on semi-Markov and Markov additive processes. *Journal of Applied Probability*, 43(4):1053–1065, 2006.

[21] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy. Pump-slowly, fetch-quickly (PSFQ): A reliable transport protocol for sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):862–872, April 2005.

[22] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 routing protocol for low-power and lossy networks. *Internet RFC 6550*, March 2012.

[23] Y. Yu, L. J. Rittle, V. Bhandari, and J. B. Lebrun. Supporting concurrent applications in wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, pages 139–152, 2006.