

On the Performance of Two-Hop Message Spreading in DTNs

Antonis Panagakis, Athanasios Vaios, and Ioannis Stavrakakis

Dept. of Informatics, University of Athens, Athens, Greece

Abstract

In this paper, a Delay Tolerant Network environment is considered where the source is in full control of the two-hop spreading mechanism by setting key parameters such as the number of copies allowed to be spread in the network and the delay bound of the messages. The introduced analysis allows for a differentiation between the source of the message and the intermediate nodes (in terms of e.g. transmission power, speed or cooperation degree). Analytical expressions for the cumulative distribution function (*cdf*) of the delivery delay and the induced overhead are extracted, taking into account the fact that the source node may continue spreading copies after the message delivery. In addition, a fairly accurate approximate expression for the *cdf* of the delivery delay is also derived and validated through simulations.¹

Key words: Delay Tolerant Networks, Routing, Performance Analysis

1 Introduction

Delay Tolerant Networks (DTNs) are characterized by a relatively high delay in delivering the information to a destination node due to the fact that there is typically no contemporaneous path between a potential source-destination pair. In such networks, traditional ad hoc routing algorithms cannot be efficient, [1][2]. For instance, reactive algorithms would suffer from the fact

¹ This work has been supported in part by the: i) IST BIONETS program under contract FP6-027748; ii) Cyprus Research Promotion Foundation under contract ΠΑΗΡΟ/1104/10; and iii) IST-FET SOCIALNETS project (No. 217141). A shorter version of this paper has been accepted for publication in the Proceedings of the WiOpt 2007 conference.

Email address: {apan, avaios, ioannis}@di.uoa.gr (Antonis Panagakis, Athanasios Vaios, and Ioannis Stavrakakis).

that the route discovery procedure may lead to invalid paths (if any actually found) while proactive algorithms would spend too much overhead for information that would not be credible. Furthermore, source-based techniques are expected to be inappropriate since the selected path (if any discovered) will most likely be invalid before it is used. Thus, per-hop routing techniques seem to be the only viable option for DTNs due to the lack of guaranteed end-to-end connectivity.

The routing mechanisms considered within the context of DTNs rely on mobility as the main means to achieve connectivity and, thus, nodes should carry and relay data at each node encounter until data is eventually delivered to the destination, [3–10]. These routing mechanisms are categorized depending on the number of copies of the message they allow to be present in the network and the employed message relaying strategy. In case of single copy strategies – where only a single copy of each message exists in the network at any time – a lower number of transmissions is required and less bandwidth is wasted, [4]. On the other hand, in multiple-copy strategies, [14], multiple copies of a message may exist concurrently in the network leading, generally, to a lower delivery delay and higher robustness. The way the overhead may be limited for several routing variations is explored in [11], using the ODE (Ordinary Differentiation Equations) fluid model.

This work focuses on the two-hop relay algorithm, [12][13], where the source node relays a copy of the message to every node it encounters until it reaches the destination; the intermediate nodes that acquire a copy of the message are allowed to forward the copy only to the destination and, thus, one hop or, maximum, two hops of communication are employed for the message delivery. An important characteristic of the multiple-copy, two-hop relay algorithm is that the source itself fully controls the process of spreading the copies of the message, unlike having intermediate nodes assume part of this responsibility. This attribute may lead to the increased resilience of the copy-spreading process as it does not involve any potentially non-cooperative or misbehaving intermediate nodes that may decide to drop the copies, [15].

In the previous studies, the performance of the introduced routing algorithms has been evaluated basically through measuring two metrics: (i) the mean message delivery delay achieved; and (ii) the overhead spent, expressed as the total number of transmissions that are needed until the message delivery. First, although the mean message delivery delay is indicant of the performance of the algorithm employed, it still cannot fully illustrate the case in which one considers an upper bound on the tolerated delivery delay. Second, the overhead actually spent in the copy-spreading process is generally larger than the one measured until the message delivery and one should also take into account the case of a heterogeneous environment where, for instance, different transmission ranges may be employed.

Taking the above into account and in order to provide for a more general framework within which the two-hop relay algorithm may be studied, the setting that is considered in this work allows for a differentiation between the source of the message and the intermediate nodes (in terms of e.g. transmission power, speed or cooperation degree), as discussed in detail in Section 2. In addition, instead of allowing the source to relay the message to all the intermediate nodes that it encounters, the number of copies allowed to be spread in the network is treated here as a design parameter.

Based on the introduced setting, the delivery delay is fully characterized by extracting its *cdf* (cumulative distribution function); moreover, an approximate approach is proposed that leads to a fairly accurate and much simpler expression for the *cdf*, as shown in Section 3. In addition, the overhead required, in terms of both the number of transmissions and energy consumed, is considered not only upon the delivery of the message, but also until the actual termination of the algorithm, which takes place when the source becomes aware of the delivery, and is analytically derived in Section 4. (When the message is delivered to the destination by some intermediate node, the source continues to forward the rest of the copies of the message allowed to be spread until it becomes aware of the successful delivery and, thus, the number of transmissions until message delivery is only a fraction of the total number of transmissions that will eventually take place.)

For all the above derivations, we assume that each message has a certain delay bound upon the expiration of which the message is dropped. This delay bound might be considered either as an application-specific characteristic (e.g. a calendar-related message or a meeting reminder that are expected to be delay-bounded) or as a design parameter of the algorithm (in order, for instance, to limit the spreading of copies by determining an appropriate delay bound for the messages within which their delivery is expected).

2 Model Description

According to the two-hop relay algorithm employed here, [12], the source node is allowed to spread up to a maximum number of copies (K) within the network. Each time it encounters some other node with no copy of the message, it gives it one until it has only one copy available (for the destination node). The intermediate nodes are not allowed to spread the message copy they may have to any other node than the destination.

Let $N + 1$ be the total number of nodes moving within a square area of size L^2 . It is assumed that the node intermeeting times (i.e., the time elapsed between two consecutive encounters for a given pair of nodes) are exponentially

distributed. This assumption has been demonstrated to be fairly accurate in the case the communication range R is such that $R \ll L$ and that nodes move according to the random waypoint or the random direction model, [12]. As it has been shown in [12], the rate at which a given node encounters some other node, $\hat{\lambda}$, may be linked to the nodes' relative speed v , communication range R and network area L^2 :

$$\hat{\lambda} = c \frac{vR}{L^2}, \quad (2.1)$$

where c is a constant that depends on the mobility model used.

For the random direction model that we employ in our simulation, c is equal to 1. For two nodes with velocities v_1 and v_2 with $v_2 > v_1$, the expected value of the relative speed v is given by $v = \frac{1}{\pi v_1 v_2} \int_{v_2 - v_1}^{v_1 + v_2} \left(x^2 / \sqrt{1 - \left(\frac{v_1^2 + v_2^2 - x^2}{2v_1 v_2} \right)^2} \right) dx$. (For more details regarding both mobility models, see Section 4.2 in [12].)

The setting that is considered in this work allows both for the spreading of a restricted ($K \leq N$) number of copies and for diverse intermeeting times between the source and the other nodes. More specifically, from equation (2.1), it is become evident that a possible differentiation between the transmission range and/or the speed of the intermediate nodes and the source node leads to a differentiation between their corresponding parameters; in the sequel, we use the parameter λ to denote the parameter for the source node and λ_o to denote the parameter for the intermediate nodes, respectively. In addition, a differentiation between the above two parameters may be associated with other causes like a possible misbehaviour of the intermediate nodes by assuming, for instance, that an intermediate node discards a copy without further relaying it at its encounter with the destination with a certain probability $1 - d$ (d expresses the degree of cooperation of the intermediate nodes); assuming, for simplicity, a homogeneous network with respect to the transmission range and speed of the nodes, the network may be modeled by setting $\lambda_o = d\lambda$.

The above parameters that may lead to differentiation could either be considered as non-tunable (e.g., the capabilities of some nodes to transmit by default at a higher power level or a node's misbehavior or cooperation degree), or as tunable (e.g., adjustment of the transmission range of the nodes within the technological power limitations in order to satisfy specific performance goals).

3 Derivation of the *CDF* of the delivery delay

Under the aforementioned assumptions, the message spreading process according to the K -limited, two-hop relay algorithm may be modeled by the

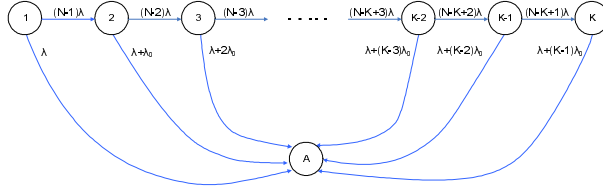


Fig. 1. The Markov chain for the two-hop relay algorithm.

continuous-time Markov chain depicted in Fig.1, consisting of $K + 1$ states (K states capturing the number of copies spread in the network and one absorbing state, A , assumed to be visited when the message is delivered to the destination; generally, $K \leq N$). When there are i copies ($i \leq K$) in the network, a new copy is forwarded either to one of the $(N - i)$ nodes which do not have a copy yet (at the rate of $\lambda(N - i)$ triggering a transition from i to $i + 1$) or to the destination (at the rate of $\lambda + (i - 1)\lambda_o$ triggering a transition from i to A). Thus, the transition rates of the Markov chain of Fig.1 are

$$q(i, j) = \begin{cases} \lambda(N - i), & i = 1, \dots, K - 1, j = i + 1; \\ \lambda + \lambda_o(i - 1), & i = 1, \dots, K, j = A; \\ 0, & \text{otherwise,} \end{cases}$$

where λ and λ_o denote the exponential distribution parameter of the inter-meeting times for the source and for the rest of the nodes, respectively.

In order to derive the *cdf* (cumulative distribution function) of the message delivery delay D , that is the time consumed until the message is eventually delivered to the destination, we first derive its *pdf* (probability distribution function), $f_D(t)$, and then use the properties of the Laplace transform. More specifically, based on the property of the Laplace transform concerning the integration in time domain, the *cdf* of the message delivery delay $Q(t) = P(D \leq t) = \int_0^t f_D(x)dx$ may be expressed as

$$Q(t) = \mathcal{L}^{-1} \left(\frac{F_D(s)}{s} \right), \quad (3.1)$$

where $F_D(s) = \mathcal{L}(f_D(t))$.

It should be noted that the notation $\mathcal{L}(g)$ ($\mathcal{L}^{-1}(g)$) is used for the Laplace (inverse Laplace) transform of some function g .

a. Deriving $f_D(t)$.

By conditioning on the state at which the message delivery takes place, the

pdf of the message delivery delay D , may be expressed as

$$f_D(t) = \sum_{i=1}^K p_{d,i} f_{D_i}(t), \quad (3.2)$$

where $p_{d,i}$ denotes the probability to be in state i when the destination gets the message with

$$p_{d,i} = \begin{cases} \frac{q(i,A)}{q_i} \prod_{j=1}^{i-1} \left(1 - \frac{q(j,A)}{q_j}\right), & i = 1, \dots, K-1; \\ 1 - \sum_{j=1}^{K-1} p_{d,j}, & i = K; \\ 0, & \text{otherwise,} \end{cases}, \quad q_i \triangleq q(i, i+1) + q(i, A) \quad (3.3)$$

and $f_{D_i}(t)$ denotes the *pdf* of D_i , which is defined as the conditional delay until the message reaches the destination provided that the destination gets the message when the system is in state i . Obviously, it holds that $D_i = \sum_{j=1}^i T_j$, where T_j denotes the sojourn time in state j .

b. Deriving $F_D(s)$.

By applying the linear property of the Laplace transform to the formula of $f_D(t)$ in equation (3.2), the Laplace transform of the *pdf* of D may be expressed as

$$F_D(s) = \mathcal{L}(f_D(t)) = \mathcal{L}\left(\sum_{i=1}^K p_{d,i} f_{D_i}(t)\right) = \sum_{i=1}^K p_{d,i} \mathcal{L}(f_{D_i}(t)). \quad (3.4)$$

By the definition of the model, the sojourn times in states j , T_j , are independent random variables that are exponentially distributed with parameter q_j . Thus the *pdf* of T_j is $f_{T_j}(t) = q_j e^{-q_j t}$, while the Laplace transform of the *pdf* is $\mathcal{L}(f_{T_j}(t)) = \mathcal{L}(q_j e^{-q_j t}) = \frac{q_j}{q_j + s}$.

Now, $D_i = \sum_{j=1}^i T_j$ is the sum of the sojourn times at states j that precede state i . This implies that the Laplace transform of D_i , $\mathcal{L}(f_{D_i}(t))$, may be expressed as the product of the Laplace transform of T_j 's, that is

$$\mathcal{L}(f_{D_i}(t)) = \prod_{j=1}^i \mathcal{L}(f_{T_j}(t)). \quad (3.5)$$

Using equations (3.3)-(3.5) it is obtained that

$$F_D(s) = \sum_{i=1}^{K-1} B_i C_i(s) + B_K C_K(s), \quad (3.6)$$

where

$$\begin{aligned} B_i &= \frac{q(i, A)}{q_i} \left(\prod_{j=1}^{i-1} \left(1 - \frac{q(j, A)}{q_j} \right) \right) \prod_{j=1}^i q_j \\ &= (\lambda_d + (\lambda - \lambda_d)i) \lambda^{i-1} \frac{(N-1)!}{(N-i)!}, \quad 1 \leq i \leq K-1, \end{aligned}$$

$$\begin{aligned} B_K &= \left(1 - \sum_{i=1}^{K-1} \frac{q(i, A)}{q_i} \prod_{j=1}^{i-1} \left(1 - \frac{q(j, A)}{q_j} \right) \right) \prod_{j=1}^K q_j \\ &= \lambda^{K-1} (K\lambda - (K-1)\lambda_d) \frac{(N-1)!}{(N-K)!}, \end{aligned}$$

$$\lambda_d = \lambda - \lambda_o,$$

and

$$C_i(s) = \prod_{j=1}^i \frac{1}{q_j + s}.$$

c. Obtaining the cdf.

By substituting the expression derived in equation (3.6) in that of (3.1), the *cdf* of the message delivery delay may be expressed as

$$\begin{aligned} Q(t) &= \mathcal{L}^{-1} \left(\frac{F_D(s)}{s} \right) = \mathcal{L}^{-1} \left(\sum_{i=1}^{K-1} B_i \frac{C_i(s)}{s} + B_K \frac{C_K(s)}{s} \right) \\ &= \sum_{i=1}^{K-1} B_i \mathcal{L}^{-1} \left(\frac{C_i(s)}{s} \right) + B_K \mathcal{L}^{-1} \left(\frac{C_K(s)}{s} \right), \end{aligned}$$

or

$$Q(t) = \sum_{i=1}^{K-1} B_i m_i(t) + B_K m_K(t), \quad (3.7)$$

where $m_i(t) = \mathcal{L}^{-1}\left(\frac{C_i(s)}{s}\right)$. The derivation of $m_i(t)$ is lengthy and, thus, quoted in the Appendix.

3.1 Special cases.

The above expression for $Q(t)$ holds for the case $K < N$ and $\lambda_d \neq 0$. Following the same steps, it may be concluded that for $K < N$ and $\lambda_d = 0$, the *cdf* of the delivery delay, denoted as $Q_0(t)$, is

$$\begin{aligned} Q_0(t) = & \sum_{i=1}^{K-1} i \lambda^i \frac{(N-1)!}{(N-i)!} \left(\left(\frac{1}{\lambda N}\right)^i - \sum_{k=1}^i \frac{e^{-\lambda N t}}{(k-1)! \lambda N} t^{k-1} \right) \\ & + \lambda^K K \frac{(N-1)!}{(N-K)!} \left(\frac{1}{\lambda K (\lambda N)^{K-1}} - \frac{e^{-\lambda K t}}{\lambda K (\lambda(N-K))^{K-1}} \right. \\ & \left. - \frac{e^{-\lambda N t}}{\lambda K} \sum_{k=1}^{K-1} \frac{\left(\left(\frac{1}{\lambda N}\right)^{K-k} - \left(\frac{1}{\lambda(N-K)}\right)^{K-k}\right) t^{k-1}}{(k-1)!} \right). \end{aligned}$$

In the special case where $K = N$ and $\lambda_d \neq 0$ it is obtained that the *cdf* of the delivery delay, denoted as $Q_N(t)$, is

$$Q_N(t) = 1 - e^{-\lambda N t} \left(1 + \frac{\lambda}{\lambda_d} (e^{\lambda_d t} - 1) \right)^{N-1}. \quad (3.8)$$

For $K = N$ and $\lambda_d = 0$ the *cdf* may be obtained by taking the limit as $\lambda_d \rightarrow 0$,

$$\begin{aligned} \lim_{\lambda_d \rightarrow 0} Q_N(t) &= \lim_{\lambda_d \rightarrow 0} \left(1 - e^{-\lambda N t} \left(1 + \frac{\lambda}{\lambda_d} (e^{\lambda_d t} - 1) \right)^{N-1} \right) \\ &= 1 - e^{-\lambda N t} (1 + \lambda t)^{N-1}. \end{aligned}$$

The analysis was validated through simulations. Figure 2 depicts an indicative example for the cases of $K = 4, 8, 100$ when $N = 100$, $\lambda = 0.08$ and $\lambda_o = 0.04$. For the simulations, a network of 100 nodes that move according to the random direction mobility model with a speed of $11.2m/sec$ within a square area of a side of $16km$ has been considered. The source node has a transmission range of $200m$ and spreads up to 3, 7 and 99 copies of the message to the intermediate

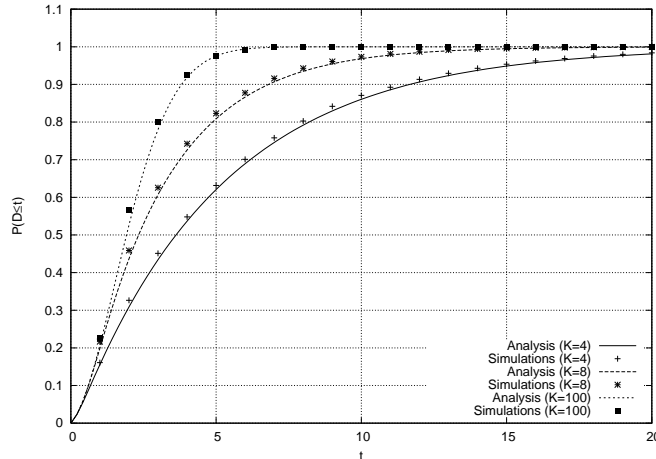


Fig. 2. Theoretical and simulation results for $N = 100$, $\lambda = 0.08$, $\lambda_o = 0.04$ and for the cases of $K = 4, 8, 100$.

nodes ($K = 4, 8, 100$, respectively) that have a transmission range of $100m$; the results refer to the average of 20000 runs.²

3.2 Approximate expression for the cdf of the delivery delay

Instead of using the equation (3.7) in order to calculate the *cdf* of the delivery delay, an approximate expression is derived here, giving a much simpler expression that approximates fairly accurately the exact one. Its derivation is based on the observation that the *cdf* for a specific number of copies K where $K \leq N$, named *accurate cdf* hereafter, is upper bounded by two *cdfs*.

The first one, named *maximum-copy cdf* hereafter, refers to the case where the number of copies employed in the network equals the number of nodes ($K = N$), is denoted as $Q_N(t)$ and is given by equation (3.8). This is because the algorithm has exactly the same behaviour up to the time point where the first K copies are spread in the network, while, afterwards, the setting corresponding to the maximum-copy *cdf* takes advantage of the surplus copies ($N - K$) allowed to be spread to enhance its performance.

² As it may be seen in equation (2.1), the parameters of λ and λ_o do not only depend on the transmission range of the source and the intermediate nodes but also on their speed and the area size. Although in this paper we only change the transmission ranges in order to derive different values for λ and λ_o , the same results are derived for other parameter sets as well (for instance, we obtained almost identical results for a transmission range of all nodes equal to $200m$ and a speed of the source node and the intermediate nodes (moving in an area of $16km^2$) of $13.632m/sec$ and $5.58m/sec$, respectively). It should also be noted that (although not illustrated) parameters λ and λ_o are here given in hrs^{-1} and, thus, time, t , is measured in hrs .

The second one, named *zero-spreadtime cdf* hereafter, refers to the *hypothetical ideal* case where upon the generation of a message, all K copies are assumed to have been spread *instantly* in the network; one copy is owned by the source node and $(K - 1)$ by some intermediate nodes. In that case, a transition to the state of the destination's having acquired the message takes place at a rate of $(\lambda + \lambda_o(K - 1))$ and, thus, the zero-spreadtime *cdf* of the delivery delay, denoted as $Q_{\hat{K}}(t)$, is given by

$$Q_{\hat{K}}(t) = 1 - e^{-(\lambda + \lambda_o(K-1))t}. \quad (3.9)$$

What may be intuitively expected and is indeed observed regarding the zero-spreadtime *cdf* is that when it is shifted to be tangent to the maximum-copy one, the part of the *cdf* from the contact point and afterwards is a fairly accurate approximation of the original *cdf*.

Based on the above observations, an approximate expression for the *cdf* of the delivery delay, denoted as $\hat{Q}_K(t)$, may be defined as a two-part function, consisting of the maximum-copy *cdf* until being tangent to the shifted zero-spreadtime one and of the latter afterwards, or

$$\hat{Q}_K(t) = \begin{cases} Q_N(t) = 1 - e^{-\lambda N t} \left(1 + \frac{\lambda}{\lambda_d} (e^{\lambda_d t} - 1)\right)^{N-1}, & 0 \leq t \leq t_{cr}; \\ Q_{\hat{K}}(t - t_0) = 1 - e^{-(\lambda + \lambda_o(K-1))(t-t_0)}, & t \geq t_{cr}, \end{cases}$$

where t_0 denotes the time shift of the zero-spreadtime *cdf* needed to be tangent to the maximum-copy one and t_{cr} corresponds to the contact point of the above two *cdfs*.

In order to calculate the contact point of the shifted zero-spreadtime *cdf* and the maximum-copy *cdf*, the equation $Q_N(t) = Q_{\hat{K}}(t - t_0)$ should be solved; by setting $y = \lambda t$ and $z = \lambda t_0$,

$$e^{-y} \left(1 + \frac{\lambda}{\lambda_d} \left(e^{\frac{\lambda_d y}{\lambda}} - 1\right)\right)^{\frac{N-1}{N}} = e^{-\frac{(\lambda + \lambda_o(K-1))(y-z)}{\lambda N}}.$$

To solve the above equation we expand each part in a Taylor series and keep the terms up to the second order. Thus, it is obtained

$$1 + a_1 y + a_2 y^2 = b_0 \left(1 + b_1 y + \frac{1}{2} (b_1 y)^2\right),$$

where $a_1 = -\frac{1}{N}$, $a_2 = \left(\frac{\lambda_d}{2\lambda} + \frac{1}{2N^2} + \frac{1}{2N} - \frac{\lambda_d}{2\lambda N} - \frac{1}{2}\right)$, $b_0 = e^{-b_1 z}$, $b_1 = \left(\frac{\lambda_d K}{\lambda N} - \frac{\lambda_d}{\lambda N} - \frac{K}{N}\right)$.

In order for the polynomial above to have a single root (contact point), denoted

as y_{cr} , with

$$y_{cr} = \frac{b_1 e^{-b_1 z} - a_1}{2a_2 - b_1^2 e^{-b_1 z}}$$

it is required that its discriminant be zero, or $(a_1 - b_1 e^{-b_1 z})^2 - 4(a_2 - \frac{1}{2}b_1^2 e^{-b_1 z})(1 - e^{-b_1 z}) = 0$, which, by using the terms up to the first order of its expansion in Taylor series, $(a_1^2 - 2a_1 b_1 + b_1^2) + (2a_1 b_1^2 - 4a_2 b_1)z + O(z^2) = 0$, leads to the calculation of the required time shift of the zero-spreadtime *cdf*, t_0 ,

$$t_0 = \frac{z}{\lambda} \approx \frac{2a_1 b_1 - b_1^2 - a_1^2}{2\lambda b_1 (a_1 b_1 - 2a_2)} = \frac{\lambda_o (K - 1)^2 N}{2\lambda (\lambda K - \lambda_d (K - 1)) ((N - 1)N - K + 1)}.$$

Then, t_{cr} may be obtained by

$$t_{cr} = \frac{1}{\lambda} \frac{b_1 e^{-b_1 \lambda t_0} - a_1}{2a_2 - b_1^2 e^{-b_1 \lambda t_0}}.$$

Figure 3 illustrates how the approximate *cdf* is constructed for the case of $N = 100$, $K = 8$, $\lambda = 0.08$ and $\lambda_o = 0.04$.

The approximate *cdf* may be used in order to obtain closed-form solutions to design problems for which the exact analysis allows only for a numerical solution. For example, given the characteristics of the network and the nodes, which would allow to determine the intermeeting parameters λ and λ_o , it might be desirable to estimate the number of copies K that the source should spread in the network in order to achieve a specific delivery ratio within some time t , referred to as Q_d .

Using the above approximation, the following estimation may be obtained:

$$K_{approx} = \frac{-\lambda_d (N + (N - 2)(N + 1)\lambda t) + \lambda (N + (N(N - 1) - 1)\lambda t + \ln(1 - Q_d)) \pm \sqrt{C}}{\lambda_o (N - 2\lambda t)},$$

where $C = \lambda (2\lambda \lambda_o (N - 1)N^2 + (\lambda_d (N - 1)N + \lambda(1 - N^2 + N))^2 \lambda t) + \lambda \ln(1 - Q_d) (-2\lambda_d (N - 1)N(N - \lambda t) + 2\lambda((N - 1)N(N - \lambda t) + \lambda t) + \lambda \ln(1 - Q_d))$, and $\lambda_d = \lambda - \lambda_o$. From the two values obtained for K_{approx} , the positive one that fulfils the condition $t_0 \leq t_{cr}$ should be selected. It should be noted that the derived K_{approx} is found to provide a performance close to the targeted one even if it suggests a lower number of copies.

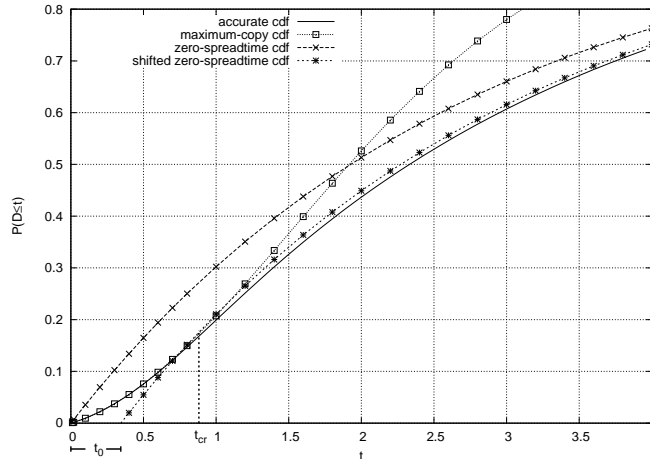


Fig. 3. Approximating the accurate *cdf* with the maximum copy *cdf* for $t \leq t_{cr}$ and the shifted zero-spreadtime *cdf* for $t > t_{cr}$.

4 Calculation of the overhead

Here, the overhead induced for the message delivery is calculated; this overhead is not only considered in terms of the number of transmissions that is required for the message to be delivered to the destination node but also in terms of energy consumption (in case of heterogeneous environments where not all the nodes employ the same transmission power). Moreover, every message is assumed to have a delay bound (that might be either an application-specific characteristic or a design parameter of the algorithm itself) after which it is dropped, no matter if it has been delivered or not to the destination.

In the sequel, the total overhead spent for the delivery of the delay-bounded messages is determined in two phases; first, the overhead until the delivery or drop of the message is extracted (both in terms of transmissions and energy consumption) and then the overhead that might occur after the delivery of the message until the source becomes aware of the message delivery, referred to as additional overhead, is calculated.

4.1 Overhead until the delivery or drop of the message

The expected overhead consumption until the delivery or drop of the message, $\overline{O}_{del|drop}$, may be expressed as

$$\overline{O}_{del|drop} = \sum_{i=1}^K P_{del|drop,i} \overline{O}_{del|drop,i},$$

where $P_{del|drop,i}$ denotes the probability that the system is in state i when the destination is reached or the message is dropped and $\overline{O}_{del|drop,i}$ denotes

the expected overhead consumption provided that the message is delivered or dropped when being in state i .

a. *Deriving $P_{del|drop,i}$.*

For $1 \leq i < K$,

$$P_{del|drop,i} = \left(1 - (1 - P_{drop,i}) \frac{q(i, i+1)}{q_i} \right) \prod_{j=1}^{i-1} (1 - P_{drop,j}) \frac{q(j, j+1)}{q_j},$$

where $P_{drop,i}$ denotes the probability that the message is dropped provided that the system is in state i . For $i = K$, we use the complementary probability

$$P_{del|drop,K} = 1 - \sum_{i=1}^{K-1} P_{del|drop,i}.$$

Let D_b denote the delay bound of the message. The probability $(1 - P_{drop,i})$ that the message is not dropped provided that the system is in state i may be expressed as

$$1 - P_{drop,i} = \begin{cases} \int_0^{D_b} f_{D_1}(t) dt, & i = 1; \\ \frac{\int_0^{D_b} f_{D_i}(t) dt}{\int_0^{D_b} f_{D_{i-1}}(t) dt}, & 1 < i \leq K - 1, \end{cases}$$

where D_i depicts the total delay until the message reaches the destination provided that the destination gets the message when the system is in state i and $f_{D_i}(t)$ denotes the *pdf* of D_i . More specifically, since the system is in state i , it holds that $\sum_{j=1}^{i-1} T_j \leq D_b$, where T_j denotes the sojourn time in state j . In addition, the message is not dropped only if $\sum_{j=1}^i T_j \leq D_b$. Furthermore, $\int_0^t f_{D_i}(x) dx = \left(\prod_{j=1}^i q_j \right) m_i(t)$. Thus,

$$1 - P_{drop,i} = \begin{cases} q_1 m_1(D_b), & i = 1; \\ \frac{q_i m_i(D_b)}{m_{i-1}(D_b)}, & 1 < i \leq K - 1. \end{cases}$$

b. *Deriving $\bar{O}_{del|drop,i}$.*

The expected overhead consumption provided that the message is delivered or dropped when the system is in state i , $\bar{O}_{del|drop,i}$ can be expressed as

$$\bar{O}_{del|drop,i} = (i - 1)E_s + P_{dels,i}E_s + P_{delo,i}E_o,$$

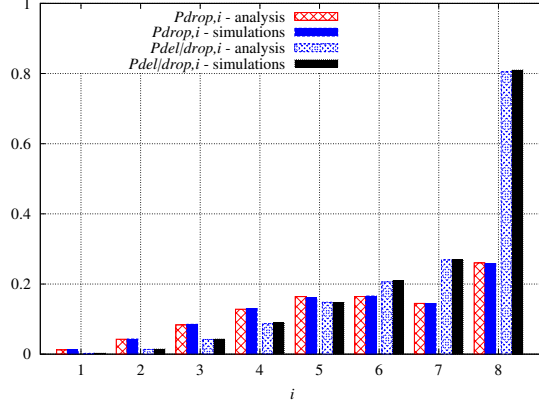


Fig. 4. $P_{del|drop,i}$ and $P_{drop,i}$ as a function of state i , for $N = 100$, $K = 8$, $\lambda = 0.02$ and $\lambda_o = 0.04$.

where $P_{dels,i}$ ($P_{delo,i}$) denotes the probability that the source (some intermediate node) delivers the message to the destination provided that the message is delivered or dropped when the system is in state i . To allow for the calculation of the consumed energy when different power levels are used for the transmissions of the source and the intermediate nodes, the parameters E_s and E_o are used respectively. In order to calculate the overhead only in terms of the number of transmissions, E_s and E_o should be assumed to be equal to 1.

When being in state i , the message is dropped with probability $P_{drop,i}$, it is delivered to the destination by some of the $i - 1$ intermediate nodes with probability $(1 - P_{drop,i}) \frac{(i-1)\lambda_o}{q_i}$ and it is delivered to the destination by the source with probability $(1 - P_{drop,i}) \frac{\lambda}{q_i}$. Taking the above into consideration, it is obtained that

$$P_{dels,i} = \frac{(1 - P_{drop,i}) \frac{\lambda}{q_i}}{P_{drop,i} + (1 - P_{drop,i}) \frac{\lambda + (i-1)\lambda_o}{q_i}}, 1 \leq i \leq K,$$

and

$$P_{delo,i} = \frac{(1 - P_{drop,i}) \frac{(i-1)\lambda_o}{q_i}}{P_{drop,i} + (1 - P_{drop,i}) \frac{\lambda + (i-1)\lambda_o}{q_i}}, 1 \leq i \leq K.$$

The analysis was validated through simulations, as shown in Fig.4, where $P_{del|drop,i}$ and $P_{drop,i}$ are calculated as a function of state i . The simulation setting refers to a network of 100 nodes that move according to the random direction mobility model with a speed of $11.2m/sec$ within a square area of a side of $16km$; the transmission range of the source node is $50m$ while that of the intermediate nodes is $100m$.

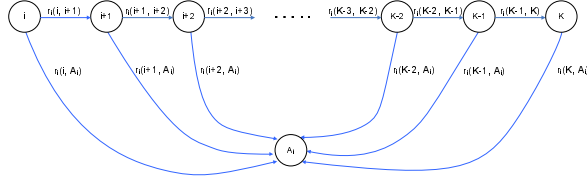


Fig. 5. The Markov chain for the additional overhead calculation.

4.2 Calculation of the additional overhead

When the message is delivered to the destination by an intermediate node, the source node will continue spreading the rest (if any) of the predefined number of copies (K); the overhead that is unnecessarily spent in this case is referred to as additional overhead. Here, the additional overhead is calculated for two distinct cases: (a) until the source node meets the destination (referred to as *single notification*); and (b) until it meets either the destination or the intermediate node that delivered the message to the destination (referred to as *double notification*).

The single notification may be considered as the straightforward notification procedure of the two-hop relay algorithm, since the source node is obviously expected to end the message copy spreading process after meeting the destination node. On the other hand, the double notification is the simplest notification procedure engaging intermediate nodes and requires that the node which has delivered the message to the destination keep the source in its memory and notify it after their encounter; this procedure is used as an indicative example of how a little more sophisticated mechanism may limit the overhead spent.

The expected additional overhead may be expressed as

$$\bar{O}_{add} = \sum_{i=2}^K P_{del|drop,i} P_{delo,i} \bar{O}_{add,i},$$

where $\bar{O}_{add,i}$ denotes the expected additional overhead provided that the message is delivered by some intermediate node when the system (as modeled by the Markov chain of Fig.1) is in state i . In order to derive $\bar{O}_{add,i}$ the system may be modeled as a Markov chain, depicted in Fig.5, starting from state i (where there are i copies of the message, one for the source node and the rest for the intermediate ones) up to state K and having an absorbing state A_i that corresponds to the case that the source node has been informed of the delivery success.

The transition rates for the Markov chain of Fig.5 are

$$r_i(j, k) = \begin{cases} \lambda(N - j), & j = i, \dots, K - 1, k = j + 1; \\ \lambda_{n,i} + \lambda_{n,d}, & j = 1, \dots, K, k = A_i; \\ 0, & \text{otherwise,} \end{cases}$$

where $\lambda_{n,i}$ and $\lambda_{n,d}$ denote the intermeeting time parameters for the intermediate node and the destination node when they notify the source, respectively. (In the case where this parameter is assumed to be the same as for message transmissions, it holds that $\lambda_{n,i} = \lambda_{n,d} = \lambda_o$.)

The expected additional energy consumption provided that the message is delivered by an intermediate node when the system is in state i , $\bar{O}_{add,i}$, may be expressed as

$$\bar{O}_{add,i} = \sum_{j=i}^K \bar{O}_{add,i,j} P_{del|drop,i,j},$$

where $\bar{O}_{add,i,j}$ denotes the expected overhead provided that the source is notified or the message is dropped when the system is in state j (Fig. 5) and $P_{del|drop,i,j}$ denotes the probability that the system is in state j when the source is notified or the message is dropped. For $i \leq j < K$, the latter term is given by

$$P_{del|drop,i,j} = \left(1 - (1 - P_{drop,i,j}) \frac{r_i(j, j+1)}{r_{i,j}} \right) \prod_{k=1}^{j-1} (1 - P_{drop,i,k}) \frac{r_i(k, k+1)}{r_{i,k}},$$

where $P_{drop,i,j}$ denotes the probability that the message is dropped provided that the system is in state j and $r_{i,j} = r_i(j, j+1) + r_i(j, A_i)$. (In case $j = K$, there is no additional overhead spent.)

Now, the message will not be dropped when the system is in state j if and only if $D_{i,j} = \sum_{k=1}^i T_k + \sum_{k=i}^j T_{i,k} \leq D_b$, where T_k denotes the sojourn time in state k of the chain in Fig.1 and $T_{i,k}$ denotes the sojourn time in state k of the chain in Fig.5. Provided that the system is in state j in Fig.5, it holds that $D_{i,j-1} = \sum_{k=1}^i T_k + \sum_{k=i}^{j-1} T_{i,k} \leq D_b$. Thus,

$$1 - P_{drop,i,j} = \begin{cases} F_{D_{i,1}}(D_b), & j = 1; \\ \frac{F_{D_{i,j}}(D_b)}{F_{D_{i,j-1}}(D_b)}, & 1 < j \leq K - 1, \end{cases}$$

with

$$F_{D_{i,j}}(t) = \int_0^t f_{D_{i,j}}(x) dx,$$

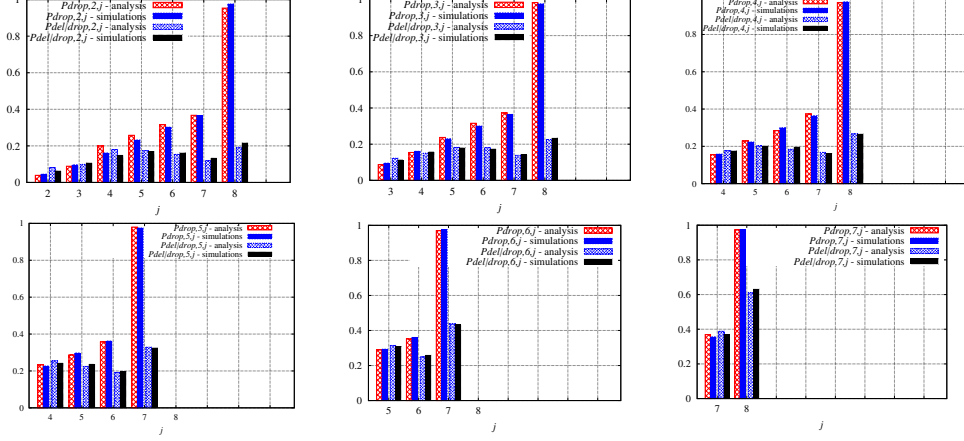


Fig. 6. $P_{del|drop,i,j}$ and $P_{drop,i,j}$ as a function of state j , for $N = 100$, $K = 8$, $\lambda = 0.02$ and $\lambda_0 = 0.04$.

where $f_{D_{i,j}}(t)$ denotes the *pdf* of $D_{i,j}$. $F_{D_{i,j}}(t)$ is derived in the Appendix.

The term $\bar{O}_{add,i,j}$ may be expressed as

$$\bar{O}_{add,i,j} = (j - 1)E_s + P_{nd,i,j}E_{nd} + P_{ni,i,j}E_{ni},$$

where $P_{nd,i,j}$ ($P_{ni,i,j}$) denotes the probability that the source is notified by the destination (the intermediate node that delivered the message) provided that the source is notified or the message is dropped when being in state j ; E_{nd} and E_{ni} denote the energy consumed for the transmission of the notification message by the destination node and an intermediate node, respectively.

Finally, $P_{nd,i,j}$ and $P_{ni,i,j}$ may be obtained by

$$P_{nd,i,j} = \frac{(1 - P_{drop,i,j}) \frac{\lambda_{n,d}}{r_{i,j}}}{P_{drop,i,j} + (1 - P_{drop,i,j}) \frac{\lambda_{n,i} + \lambda_{n,d}}{r_{i,j}}},$$

and

$$P_{ni,i,j} = \frac{(1 - P_{drop,i,j}) \frac{\lambda_{n,i}}{r_{i,j}}}{P_{drop,i,j} + (1 - P_{drop,i,j}) \frac{\lambda_{n,i} + \lambda_{n,d}}{r_{i,j}}}.$$

It should be noted that for single notification $\lambda_{n,i} = 0$; thus, $P_{ni,i,j}$ becomes equal to zero.

Analytical results are validated through simulations as shown in Fig.6, where $P_{del|drop,i,j}$ and $P_{drop,i,j}$ are depicted as a function of state j for the same simulation setting as that in Fig.4. (For $i = 8$, the results are similar.)

5 Results

For all the simulations that are presented in the sequel, a network of 100 nodes that move according to the random direction mobility model with a speed of $11.2m/sec$ within a square area of a side of $16km$ has been considered; the results refer to the average of 20000 runs.

5.1 The effect of the parameters K and D_b

Both the number of copies that are allowed in the network (K) and the delay bound that may be set to the messages (D_b) constitute two basic parameters of the two-hop relay algorithm and affect the performance of message spreading. In Fig.7, the effect of the delay bound (D_b) on the message delivery ratio (Fig.7(a)), on the number of transmissions until delivery (Fig.7(b)) and on the number of transmissions after delivery when applying single (Fig.7(c)), or double notification (Fig.7(d)), as a function of the number of copies (K) is depicted for the case of $\lambda = 0.08$ and $\lambda_o = 0.04$. As it may be seen, both the parameters K and D_b seem to significantly affect the message delivery ratio and the number of transmissions until the message delivery, up to some specific values (approximately $K = 20$ and $D_b = 10$). By allowing an extra number of copies to be spread in the network, or by increasing the delay bound, the achieved increase in the delivery ratio (and the number of transmissions until delivery) is negligible; however, there is a significant effect on the additional transmissions that take place (especially in the case of single notification).

Figures 8 and 9 illustrate the fact that the parameters K and D_b may be used either interchangeably or in combination with each other to allow for efficient message dissemination.³ More specifically, the minimum delay bound, for which a specific delivery ratio (0.5, 0.7, 0.9, 0.99) may be achieved, as a function of K is depicted in Fig.8. It can be seen that increasing K beyond a certain value does not lead to a reduction in the required value of D_b . Figure 9 illustrates the number of transmissions (until the delivery and after the delivery of the message for the case of single and double notification) as a function of K for the case where no delay bound is used (leading to a delivery ratio equal to 1) and for the case where for each K the minimum value of D_b required to achieve a delivery ratio equal to 0.99 is applied. It can be seen that: (i) K and D_b may be used interchangeably to achieve similar performance; however, (ii) when used in combination with each other, a more significant decrease of the additional overhead is observed (at the cost of a negligible reduction in the delivery ratio).

³ Although K takes only discrete values, it is treated as continuous for illustration purposes.

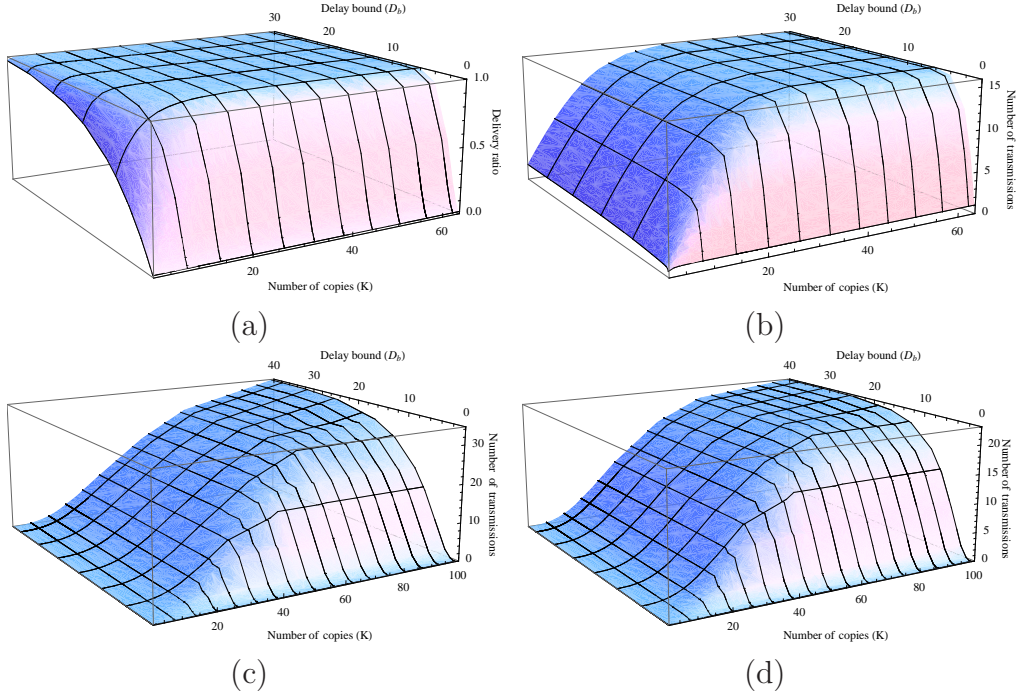


Fig. 7. The effect of the delay bound (D_b) (a) on the message delivery ratio, (b) on the number of transmissions until delivery, and on the number of transmissions after delivery when applying (c) single, or (d) double notification, as a function of the number of copies (K) for the case of $\lambda = 0.08$ and $\lambda_o = 0.04$.

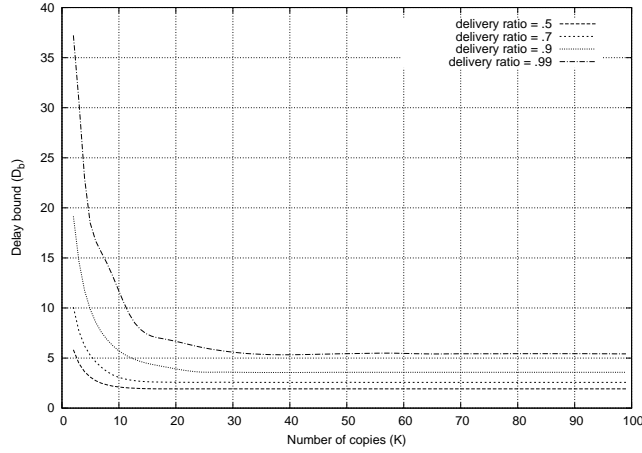


Fig. 8. The minimum delay bound, for which a specific delivery ratio (0.5, 0.7, 0.9, 0.99) may be achieved, as a function of K .

5.2 The effect of the parameters λ and λ_o

Figure 10 illustrates the number of transmissions as a function of λ for the case of $K = N = 100$ and (a) $\lambda_o = 0.08$, or (b) $\lambda_o = 0.04$. (The transmission range of the source varies between 25m and 400m corresponding to a value of λ between 0.01 and 0.16, respectively; a transmission range of (a) 200m

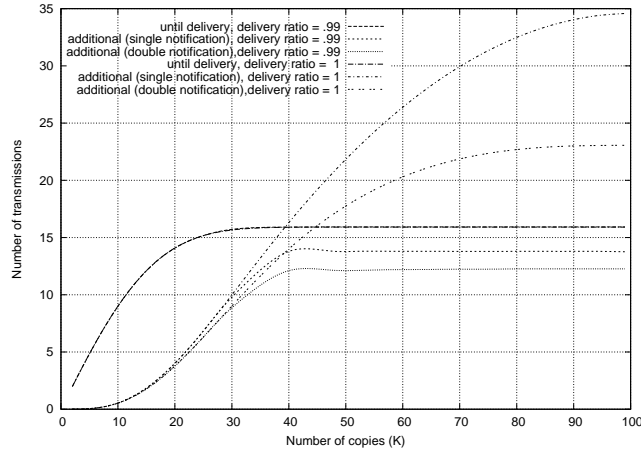


Fig. 9. The number of transmissions (until the message delivery and for single and double notification) as a function of K for the case where no delay bound is used (leading to a delivery ratio equal to 1) and for the case where for each K the minimum value of D_b required to achieve a delivery ratio equal to 0.99 is applied.

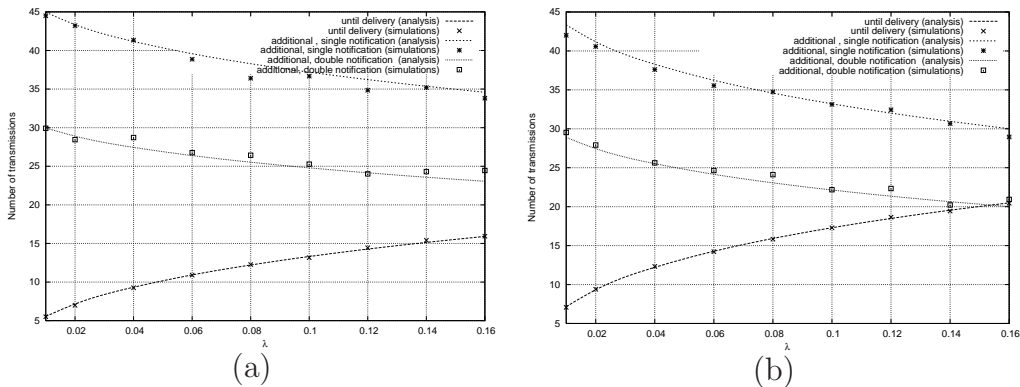


Fig. 10. The number of transmissions as a function of λ , where $K = N = 100$ and (a) $\lambda_o = 0.08$, or (b) $\lambda_o = 0.04$.

and (b) $100m$ is used instead for the intermediate nodes leading to $\lambda_o = 0.08$ and $\lambda_o = 0.04$.) It turns out that the transmissions until the message delivery time are just a small portion of the additional ones actually made; the double notification scheme achieves a noticeable decrease in the wasted overhead.

When the source node increases its parameter λ the transmissions needed until message delivery increase as well. For the case of single notification, the total number of transmissions spent is approximately 50.5, irrespectively of the values of λ or λ_o . This is rather expected since however high the parameter λ might be, the source node will anyway meet, on average, half of the nodes before reaching the destination, leading to a constant number of total transmissions.⁴

⁴ More specifically, consider a network of $N + 1$ nodes: whichever the value of λ and λ_o may be, spreading will be terminated only when the source node reaches the destination (after having encountered on average $(N + 1)/2$ nodes – including the

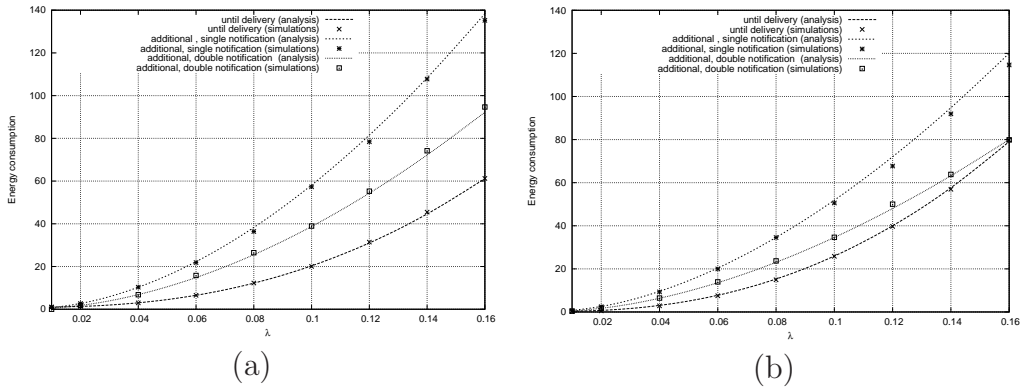


Fig. 11. The energy consumption as a function of λ , where $K = N = 100$ and (a) $\lambda_o = 0.08$, or (b) $\lambda_o = 0.04$.

In the case where not all the nodes have the same transmission range, it might be useful to calculate the induced overhead in terms of the energy spent and not only the number of transmissions. We here assume the case where the source node transmits at a different power level than the intermediate nodes, leading to different energy consumption per transmission. In particular, the energy consumed for a transmission of a node with parameter equal to 0.08 is assumed to be equal to 1. The energy consumption for all other transmissions are calculated in terms of this unit, based on the relation between the energy consumed and the parameters λ_o and λ ; we assume a quadratic relation, thus, for instance, doubling the latter parameters would imply doubling the transmission range that would, eventually, quadruple the energy consumed.

The energy consumed is depicted in Fig.11 as a function of λ for $K = N = 100$ and (a) $\lambda_o = 0.08$, or (b) $\lambda_o = 0.04$. The difference between the energy spent until the message delivery and the additional actually spent when single or double notification is applied is more noticeable as λ increases. As it may be expected, and contrary to the total transmissions, the total energy spent depends on both λ and λ_o .

destination); when the source reaches the destination, it either gives it a copy (if the message has not been delivered yet) or it does not (in case an intermediate node has already delivered the message). In the first case, the total transmissions will be equal to the average encounters of the source node till it meets the destination node, that is $(N + 1)/2$; in the second case, the total transmissions are equal to the average number of the encounters of the source node (except the one with the destination node since the latter has already been given the message) added by the intermediate node's transmission to the destination, that is $[(N + 1)/2 - 1] + 1$. Thus, the total transmissions are equal to $(N + 1)/2$ and do not depend on λ or λ_o .

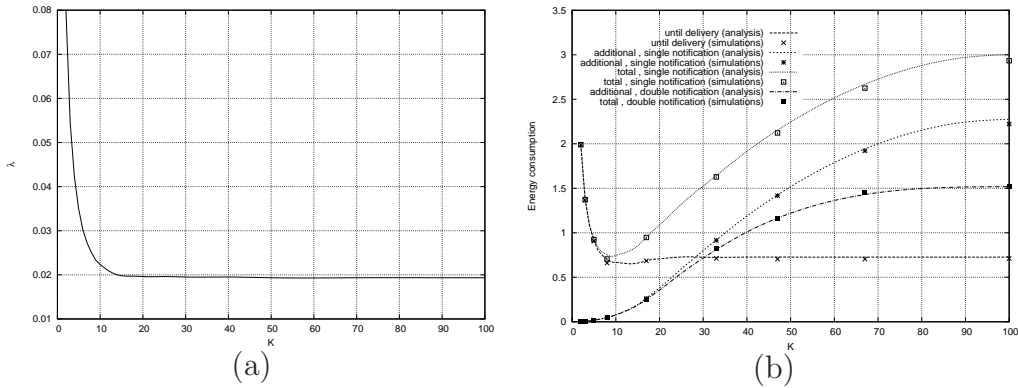


Fig. 12. (a) Pairs of λ and K that achieve the same mean delay ($\bar{D} = 6,3125$), where $\lambda = \lambda_o$, $N = 100$; (b) The energy consumption as a function of K for the pairs (K, λ) of (a).

5.3 Achieving a specific mean delay

In paragraph 5.1, it was shown that the parameters K and D_b can be used interchangeably in order to control the wasted overhead without noticeably deteriorating the performance of the two-hop relay algorithm. In this paragraph, we consider the case of delay unbounded messages ($D_b = \infty$) and use, as a performance metric, the achieved mean delay (denoted as \bar{D}) by using different pairs of K and λ . More specifically, we consider a network of $N = 100$ nodes where $\lambda = \lambda_o$. Several pairs of values for λ and K , for which the achieved mean delay is the same ($\bar{D} = 6,3125$), are depicted in Fig.12(a). The energy consumed for all these pairs of values is depicted in Fig.12(b) for the time until the message delivery or until the source node is notified (single or double notification).

From Fig.12(a), it may be concluded that for the same achieved mean delay a significant reduction in the per-transmission energy consumption (due to lower λ) may be achieved by increasing K , when K is small. Beyond a relatively small threshold value of K (less than $K = 20$ in Fig.12(a)), a further increase of K does not yield any reduction in the energy consumption due to λ . Consequently, the pair (K, λ) that achieves a given mean delay and minimizes the total energy consumption should be associated with a K below the aforementioned threshold value.

This indeed turns out to be the case as illustrated in Fig.12(b), showing that the total energy consumption is minimized for a given (K, λ) pair below that threshold value of K ; this optimal pair could be used to effectively set key design parameters of the two-hop relay algorithm.

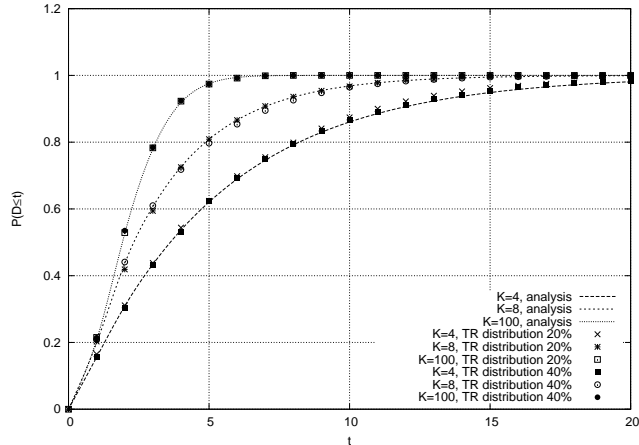


Fig. 13. Cdf of the message delivery delay for $K = 4, 8$ and 100 and for $N = 100$, $\lambda = 0.08$. In the simulations, the transmission ranges (TR) of the intermediate nodes vary 20% and 40% (distribution) around the mean value of $100m$; in the analysis, a value of $\lambda_o = 0.04$ is used (that corresponds to the transmission range of $100m$).

5.4 Differentiated behavior of the intermediate nodes

In this paragraph, we investigate the applicability of the model in two scenarios where a differentiation in the behavior of the intermediate nodes is introduced. This differentiation leads to a different value of $\hat{\lambda}$ (see equation (2.1)) for each intermediate node. We use as λ_o in our model the mean value of $\hat{\lambda}$ for the intermediate nodes in order to investigate the capability of the introduced model to also capture the case where the intermediate nodes have a differentiated behavior.

In the first scenario, the transmission range of the intermediate nodes differ; more specifically, the source node's transmission range is $200m$ (leading to $\lambda = 0.08$), while the intermediate nodes' transmission ranges vary 20% and 40% (distribution) around the mean value of $100m$. In the analysis, we use a value of $\lambda_o = 0.04$ equal to the mean value of $\hat{\lambda}$ of the intermediate nodes. The analytical and simulation results for the first scenario are illustrated in Fig.13 for the cases of $K = 4, 8$ and 100 .

In the second scenario, all the intermediate nodes have the same transmission range of $200m$ with the source node but they do not fully cooperate in message delivery; when an intermediate node with a message copy encounters the destination, it delivers the copy only with a specific probability (denoted as cooperation degree, d). In simulation results, a cooperation degree of $d = 0.2, 0.4, 0.6$ and 0.8 is applied for $\lambda = \lambda_o = 0.08$ while the analytical results are derived for fully cooperative intermediate nodes with $\lambda_o = \lambda d$. Both analytical and simulation results for the second scenario are illustrated in Fig.14.

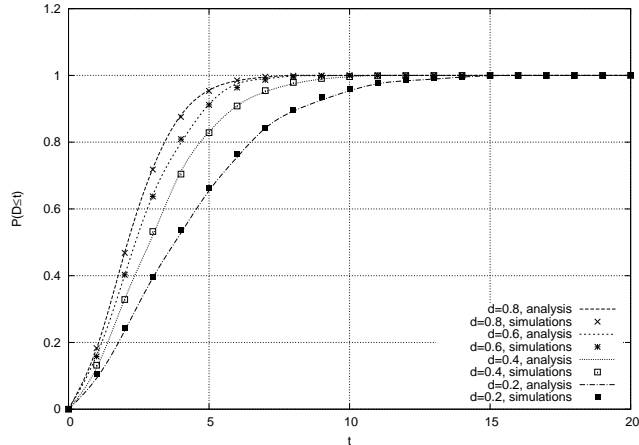


Fig. 14. Cdf of the message delivery delay for $K = N = 100$. In the simulations, a cooperation degree of $d = 0.2, 0.4, 0.6$ and 0.8 is applied for $\lambda = \lambda_o = 0.08$; the corresponding analytical results are derived for fully cooperative intermediate nodes with $\lambda_o = \lambda d$.

6 Conclusions

In this work, a model to capture the performance of the two-hop message spreading mechanism within the framework of a DTN environment is introduced, which allows for: (i) the differentiation on the characteristics between the source and the intermediate nodes – either intentionally (e.g. adjusting transmission range, misbehaving), or unintentionally (e.g., transmission power limitations), expressed by the parameters λ and λ_o respectively, and (ii) handling both the number of copies that are permitted to be spread in the network and the delay bound of a message. The analytical expressions for the message delivery ratio as well as the number of transmissions and consumed energy not only until the delivery of the message but also until the actual termination of the message copy spreading process were derived, assuming two message delivery notification procedures. Moreover, an approximation for the delivery ratio was extracted that fairly accurately captures the performance of the two-hop relay algorithm.

Based on the introduced model, the effect of the basic parameters of the algorithm (number of copies or delay bound) and the nodes (parameters λ and λ_o) was investigated. It was illustrated that the overhead wasted after the message delivery (number of transmissions or energy consumed) may be rather significant; even a simple message delivery notification procedure was proved to be a valuable mechanism when the number of copies to be spread in the network is high. In addition, it has been demonstrated that the parameters K and D_b may be used either interchangeably, or in combination with each other to allow for efficient dissemination. Finally, simulation results indicate that the model also captures well the cases of non-homogeneous networks where the

intermediate nodes may have distinct characteristics as far as their capability or willingness to forward a message is concerned (different transmission range, or cooperation degree).

7 Appendix

7.1 Derivation of $m_i(t)$

It holds that $m_i(t) = \mathcal{L}^{-1}(M_i(s))$ where

$$M_i(s) \triangleq \frac{C_i(s)}{s} = \frac{1}{s} \prod_{j=1}^i \frac{1}{q_j + s} = \begin{cases} \frac{1}{s} \prod_{j=1}^i \frac{1}{\lambda N - \lambda_d(j-1) + s}, & 1 \leq i \leq K-1; \\ \frac{1}{s} \frac{1}{\lambda + (\lambda - \lambda_d)(K-1) + s} \prod_{j=1}^{K-1} \frac{1}{\lambda N - \lambda_d(j-1) + s}, & i = K. \end{cases}$$

A.1 For $1 \leq i < K$ the poles of $M_i(s)$ are $s_{i,0}=0$ and $s_{i,k} = \lambda_d(k-1) - \lambda N$, $k = 1, \dots, i$, which are all single poles. Thus,

$$m_i(t) = \sum_{k=0}^i \bar{m}_{i,k} e^{s_{i,k}t},$$

where $\bar{m}_{i,k} = (s - s_{i,k})M_i(s)|_{s=s_{i,k}}$. For $0 < k \leq K$,

$$\bar{m}_{i,k} = \frac{1}{\lambda_d(k-1) - \lambda N} m_{i,k},$$

where

$$\begin{aligned} m_{i,k} &= \prod_{j=1, j \neq k}^i \frac{1}{\lambda N - \lambda_d(j-1) + \lambda_d(k-1) - \lambda N} \\ &= \frac{1}{\lambda_d^{i-1}} \prod_{j=1, j \neq k}^i \frac{1}{k-j} = \frac{1}{\lambda_d^{i-1}} \frac{1}{(k-1)!} \frac{(-1)^{i-k}}{(i-k)!}. \end{aligned}$$

For $k=0$,

$$\bar{m}_{i,0} = \prod_{j=1}^i \frac{1}{\lambda N - \lambda_d(j-1)} = \sum_{k=1}^i \frac{1}{\lambda N - \lambda_d(k-1)} \prod_{j=1, j \neq k}^i \frac{1}{\lambda_d(k-1) - \lambda_d(j-1)} = \sum_{k=1}^i -\bar{m}_{i,k}.$$

Thus,

$$\begin{aligned} m_i(t) &= \sum_{k=1}^i \overline{m}_{i,k} (e^{-\lambda N t} e^{\lambda_d(k-1)t} - 1) = \sum_{k=1}^i \frac{1}{\lambda_d(k-1) - \lambda N} \overline{m}_{i,k} (e^{-\lambda N t} e^{\lambda_d(k-1)t} - 1) \\ &= \sum_{k=1}^i \frac{1}{\lambda_d(k-1) - \lambda N} \frac{1}{\lambda_d^{i-1}} \frac{1}{(k-1)!} \frac{(-1)^{i-k}}{(i-k)!} (e^{-\lambda N t} e^{\lambda_d(k-1)t} - 1). \end{aligned}$$

A.2 For $i = K$ the poles of $M_K(s)$ are $s_{K,0} = 0$, $s_{K,K} = -(\lambda + (\lambda - \lambda_d)(K-1)) = (K-1)\lambda_d - K\lambda$ and $s_{K,k} = \lambda_d(k-1) - \lambda N$, $k = 1, \dots, K-1$. The above poles are all single for $-(\lambda + (\lambda - \lambda_d)(K-1)) \neq \lambda_d(j-1) - \lambda N$, $\forall j$, $1 \leq j \leq K-1$ (it can be easily concluded that $\lambda(N-K) + j\lambda_d \neq 0$, $\forall j$, $1 \leq j \leq K-1$, so none of the following denominators is zero), $s_{K,K} \neq 0$ and $s_{K,k} \neq 0$, $k = 1, \dots, K-1$. The case of poles with multiplicity greater than 1 is not considered in order to avoid unnecessary complexity; however, in the numerical results the aforementioned condition is technically never met by adding some “noise” digits. Thus,

$$m_K(t) = \sum_{k=0}^K \widetilde{m}_{K,k} e^{s_{K,k} t},$$

where $\widetilde{m}_{K,k} = (s - s_{K,k})C_K(s)|_{s=s_{K,k}}$. For $0 < k < K$,

$$\begin{aligned} \widetilde{m}_{K,k} &= \frac{1}{\lambda_d(k-1) - \lambda N} \frac{1}{\lambda + (\lambda - \lambda_d)(K-1) + \lambda_d(k-1) - \lambda N} \\ &\quad \prod_{j=1, j \neq k}^{K-1} \frac{1}{\lambda N - \lambda_d(j-1) + \lambda_d(k-1) - \lambda N} \\ &= \frac{1}{\lambda_d(k-1) - \lambda N} \frac{-1}{\lambda(N-K) + \lambda_d(K-k)} m_{K-1,k} \\ &= \frac{1}{\lambda_d(k-1) - \lambda N} \widehat{m}_{K,k}. \end{aligned}$$

where

$$\begin{aligned} \widehat{m}_{K,k} &= \frac{1}{\lambda + (\lambda - \lambda_d)(K-1) + \lambda_d(k-1) - \lambda N} \\ &\quad \prod_{j=1, j \neq k}^{K-1} \frac{1}{\lambda N - \lambda_d(j-1) + \lambda_d(k-1) - \lambda N} \\ &= \frac{-1}{\lambda(N-K) + \lambda_d(K-k)} m_{K-1,k}. \end{aligned}$$

For $k = K$,

$$\begin{aligned}\widetilde{m}_{K,K} &= \frac{1}{\lambda_d(K-1) - \lambda K} \prod_{j=1}^{K-1} \frac{1}{\lambda N - \lambda_d(j-1) + (K-1)\lambda_d - K\lambda} \\ &= \frac{1}{\lambda_d(K-1) - \lambda K} \widehat{m}_{K,K},\end{aligned}$$

where

$$\begin{aligned}\widehat{m}_{K,K} &= \prod_{j=1}^{K-1} \frac{1}{\lambda N - \lambda_d(j-1) + (K-1)\lambda_d - K\lambda} \\ &= \prod_{j=1}^{K-1} \frac{1}{\lambda(N-K) + j\lambda_d} \\ &= \sum_{k=1}^{K-1} \frac{1}{\lambda(N-K) + k\lambda_d} \prod_{j=1, j \neq k}^{K-1} \frac{1}{\lambda_d j - \lambda_d k} \\ &= \sum_{k=1}^{K-1} \frac{1}{\lambda(N-K) + k\lambda_d} \frac{1}{\lambda_d^{K-2}} \frac{(-1)^{k-1}}{(k-1)!} \frac{1}{(K-k-1)!}.\end{aligned}$$

For $k = 0$,

$$\begin{aligned}\widetilde{m}_{K,0} &= \frac{1}{\lambda + (\lambda - \lambda_d)(K-1)} \prod_{j=1}^{K-1} \frac{1}{\lambda N - \lambda_d(j-1)} \\ &= \frac{1}{\lambda K - \lambda_d(K-1)} \sum_{k=1}^{K-1} \frac{1}{\lambda N - \lambda_d(k-1)} \frac{1}{\lambda_d^{K-2}} \frac{1}{(k-1)!} \frac{1}{(K-k-1)!} \frac{(-1)^{K-k-1}}{(k-1)!}.\end{aligned}$$

Thus,

$$\begin{aligned}m_K(t) &= \sum_{k=0}^K \widetilde{m}_{K,k} e^{s_{K,k}t} = \frac{1}{\lambda K - \lambda_d(K-1)} \\ &+ \sum_{k=1}^{K-1} \frac{1}{\lambda N - \lambda_d(k-1)} \frac{1}{\lambda_d^{K-2}} \frac{1}{(k-1)!} \frac{1}{(K-k-1)!} \frac{(-1)^{K-k-1}}{(k-1)!} \\ &+ \frac{1}{\lambda_d^{K-2}} \sum_{k=1}^{K-1} \frac{1}{(k-1)!} \frac{1}{(K-1-k)!} \\ &\left(\frac{(-1)^{k-1} e^{((K-1)\lambda_d - K\lambda)t}}{(\lambda_d(K-1) - \lambda K)(\lambda(N-K) + k\lambda_d)} \right. \\ &\left. + \frac{(-1)^{K-k} e^{-\lambda N t} e^{\lambda_d(k-1)t}}{(\lambda_d(k-1) - \lambda N)(\lambda(N-K) + \lambda_d(K-k))} \right).\end{aligned}$$

7.2 Derivation of $F_{D_{i,j}}(t)$

$F_{D_{i,j}}(t)$ is derived starting from

$$\mathcal{L}\left(F_{D_{i,j}}(t)\right) = \frac{1}{s} \left(\prod_{k=1}^i \frac{q_k}{s + q_k} \right) \left(\prod_{k=i}^j \frac{r_{i,k}}{s + r_{i,k}} \right).$$

The poles $\mathcal{L}\left(F_{D_{i,j}}(t)\right)$ are

$$s_{i,j,k} = \begin{cases} 0, & k = 0; \\ -q_k, & 1 \leq k \leq i - 1; \\ -r_{i,k}, & i \leq k \leq j; \\ -q_i, & k = j + 1. \end{cases}$$

(We consider $-q_i$ as the $j + 1$ pole.) We consider only the case where the multiplicity of each pole is equal to one; this requires that:

$$\begin{aligned} q_k &\neq q_{k'}, & 1 \leq k, k' \leq i, k \neq k'; \\ r_{i,k} &\neq r_{i,k'}, & i \leq k, k' \leq j, k \neq k'; \\ q_k &\neq r_{i,k'}, & 1 \leq k \leq i, i \leq k' \leq j; \\ q_k &\neq 0, & 1 \leq k \leq i; \\ r_{i,k} &\neq 0, & i \leq k \leq j. \end{aligned}$$

(In the numerical results it is ensured that these conditions are met by using some additional “noise” digits for the values of the parameters.) Then,

$$F_{D_{i,j}}(t) = \sum_{k=0}^{j+1} c_{i,j,k} e^{s_{i,j,k} t},$$

where

$$c_{i,j,k} = (s - s_{i,j,k}) \mathcal{L}\left(F_{D_{i,j}}(t)\right) \Big|_{s=s_{i,j,k}}, \quad 0 \leq k \leq j + 1.$$

Thus,

$$c_{i,j,k} = \begin{cases} 1, & k = 0; \\ - \left(\prod_{m=1}^{k-1} \frac{q_m}{q_m - q_k} \right) \left(\prod_{m=k+1}^{i-1} \frac{q_m}{q_m - q_k} \right) \left(\prod_{m=i}^j \frac{r_{i,m}}{r_{i,m} - q_k} \right), & 1 \leq k \leq i - 1; \\ - \left(\prod_{m=1}^{i-1} \frac{q_m}{q_m - r_{i,k}} \right) \left(\prod_{m=i}^{k-1} \frac{r_{i,m}}{r_{i,m} - r_{i,k}} \right) \left(\prod_{m=k+1}^j \frac{r_{i,m}}{r_{i,m} - r_{i,k}} \right), & i \leq k \leq j; \\ - \left(\prod_{m=1}^{i-1} \frac{q_m}{q_m - q_i} \right) \left(\prod_{m=i}^j \frac{r_{i,m}}{r_{i,m} - q_i} \right), & k = j + 1. \end{cases} \quad (7.1)$$

7.3 Table of notations

Description of notations	
K	Maximum number of copies allowed to be spread in the network
$N + 1$	Total number of nodes
L^2, R, v	Size of the square area, communication range, expected value of relative speed (see equation (2.1))
λ	Rate at which the source node encounters some other node
λ_o	Rate at which an intermediate node encounters some other node (not the source)
d	Cooperation degree (see Sections 2 and 5.4)
λ_d	$\lambda - \lambda_o$
$q(i, j), q_i$	Transition rates for the continuous-time Markov chain of Fig.1
T_i	Sojourn time in state i of the Markov chain of Fig.1
D	Message delivery delay
D_i	Conditional message delivery delay, given that the message is delivered to the destination when exactly i copies of the message have been spread in the network
D_b	Delay bound of a message
$p_{d,i}$	Probability that i copies of the message have been spread in the network when the destination gets the message
$f_x(t)$	Probability distribution function (<i>pdf</i>) of the random variable x
$F_x(s)$	Laplace transform of the function $f_x(t)$
$Q(t)$	Cumulative distribution function (<i>cdf</i>) of the message delivery delay (<i>accurate cdf</i>)
$Q_0(t)$	Cdf of the message delivery delay in case $K < N$ and $\lambda = \lambda_o$
$Q_N(t)$	Cdf of the message delivery delay in case $K = N$ and $\lambda \neq \lambda_o$ (<i>maximum-copy cdf</i>)
$Q_{\hat{K}}(t)$	Cdf of the message delivery delay in the ideal case of all copies having being instantly spread in the network (<i>zero-spreadtime cdf</i>)
$\hat{Q}_K(t)$	Approximate expression for $Q(t)$
$B_i, m_i(t), c_i(t)$	Auxiliary functions to calculate $Q(t)$ (see section 3)
$M_i(s), C_i(s)$	Laplace transform of $m_i(t), c_i(t)$
K_{approx}	Estimation of the number of copies that the source should spread in the network in order to achieve a specified delivery ratio Q_d

$\overline{O}_{del drop}$	Expected overhead consumption until the delivery or drop of the message
$\overline{O}_{del drop,i}$	Expected overhead consumption provided that the message is delivered or dropped when exactly i copies of the message have been spread in the network
$P_{del drop,i}$	Probability that exactly i copies of the message have been spread in the network when the destination is reached or the message is dropped
$P_{drop,i}$	Probability that the message is dropped provided that exactly i copies of the message have been spread in the network
$P_{dels,i}$ ($P_{delo,i}$)	Probability that the source (some intermediate node) delivers the message to the destination provided that the message is delivered or dropped when exactly i copies of the message have been spread in the network
E_s (E_o)	Energy consumed for a single transmission of the source node (an intermediate node)
\overline{O}_{add}	Expected additional overhead
$\overline{O}_{add,i}$	Expected additional overhead provided that the message is delivered to the destination by some intermediate node when exactly i copies of the message have been spread in the network
$r_i(j, k), r_{i,j}$	Transition rates of the Markov chain of Fig.5
$T_{i,k}$	Sojourn time in state k of the Markov chain of Fig.5
$\lambda_{n,d}$ ($\lambda_{n,i}$)	Rate at which the destination node (an intermediate node) encounters the source node, notifying it of the message delivery
$\overline{O}_{add,i,j}$	Expected overhead provided that the source is notified or the message is dropped when exactly j message copies have been spread in the network after the delivery of the message when i copies of the message were spread in the network
$P_{del drop,i,j}$	Probability that exactly j copies of the message have been spread in the network when the source is notified or the message is dropped after the delivery of the message when i copies of the message were spread in the network
$D_{i,j}$	Delay until j copies of the message have been spread in the network, provided that the message was delivered to the destination when there were i copies of the message
$P_{nd,i,j}$ ($P_{ni,i,j}$)	Probability that the source node is notified by the destination (the intermediate node that delivered the message) provided that the source is notified or the message is dropped when exactly j message copies have been spread in the network after the message was delivered to the destination when exactly i copies were spread
E_{nd} (E_{ni})	Energy consumed for a single notification message transmission by the destination node (intermediate node)

References

- [1] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: An approach to interplanetary internet," in *Communications Magazine*, June 2004, vol.41, pp. 128-136.
- [2] K. Fall, "A delay tolerant networking architecture for challenged internets," in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003.
- [3] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad-hoc networks," Tech. Rep. Duke CS-2000-06, Duke University, April 2000.
- [4] K. Psounis, T. Spyropoulos and C. Raghavendra, "Single-copy routing in intermittently connected networks," in *Proceedings of IEEE SECON: Society Conference on Sensor and Ad Hoc Communications and Networks*, October 2004.

- [5] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proceedings of SIGCOMM 2005 workshop on Delay Tolerant Networking*, August 2005.
- [6] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, 2004.
- [7] J. Byers and G. Nasser, "Utility-based decision-making in wireless sensor networks," Tech. Rep. BUCS-2000-014, Boston University, June 2000.
- [8] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *Proceedings of the IEEE 6th International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM)*, 2005.
- [9] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, 2004.
- [10] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay tolerant network," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2005.
- [11] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," in *Computer Networks*, July 2007, vol.51, pp. 2867-2891.
- [12] R. Groenevelt, "Stochastic models in mobile ad hoc networks," Ph.D. dissertation, University of Nice Sophia Antipolis, April 2005.
- [13] R. Groenevelt, P. Nain, and G. Koole, "Message delay in mobile ad hoc networks," in *Proceedings of Performance 2005*, Juan-les-Pins, France, October 2005. Published in *Performance Evaluation*, vol. 62, Issues 1-4, October 2005, pp. 210-228.
- [14] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multi-copy Case," in *IEEE/ACM Transactions on Networking*, February 2008, vol. 16, no. 1, pp. 77-90.
- [15] A. Panagakis, A. Vaios, and I. Stavrakakis, "On the effects of cooperation in dtns," in *The Second IEEE/Create-Net/ICST International Conference on COMMunication System softWARE and MiddlewaRE (COMSWARE)*, 2007.