Performance Analysis of Self Limiting Epidemic Forwarding

Technical Report LCA-REPORT-2006-127

Alaeddine El Fawal, Jean-Yves Le Boudec, and Kave Salamatian

EPFL, I&C
CH-1015 Lausanne, Switzerland
{alaeddine.elfawal,jean-yves.leboudec,kave.salamatian}@epfl.ch

Self limiting epidemic forwarding is a method of epidemic information dissemination in wireless ad-hoc networks that achieves congestion control by limiting spread (i.e. the number of nodes that receive a given message) and injection rate in order to preserve a meaningful service. We analyze the performance of various methods for spread control: on one hand, the classical method, which consists in decrementing the TTL of a packet when it is transmitted, on the other hand, two methods based on "aging", where the TTL of a packet may be decremented while it is waiting for transmission in the epidemic buffer. The aging methods are: (selective aging) decrement TTL of a waiting packet when a duplicate is received and (global aging) decrement when any packet is received. The performance metrics are based on injection rates of messages, on amount of redundant information and on spread. We use detailed, realistic simulation for medium scale networks (up to 800 nodes); for networks of any size, we use an analytical method based on fluid approximation and solution of a fixed point problem. We find that the classical method does not perform well. Selective aging improves the performance, and global aging performs much better; it manages to control the spread so that the rate of injection remains good with fixed parameters across a wide range of settings.

I. Introduction

Epidemic forwarding has been proposed for use in wireless ad-hoc (i.e. infrastructure less) networks to disseminate information in broadcast mode [8], [11], [19], [20]. The principle is that nodes repeat with some probability the information the heard from others, thus propagating fresh information around. It is designed to operate in quickly varying environments, where mobility and self-organization make the classical methods based on distributions trees non practical. It can be used to support routing, resource discovery protocols, or applications such as chatting in a traffic jam, coupon advertisements [10], or pocket switched networks [18].

The performance of epidemic forwarding depends heavily on the control of the forwarding factor (i.e. the probability that a node repeats a given message). There is a tradeoff between amount of redundant information and delivery ratio. Various mechanisms have been proposed in [11], [19], [20] to reach satisfactory trade-offs; they differ in the assumptions they make on the information required about one node's neighborhood. These references assume that the semantics of the service requires that all nodes receive all information, as is typical for example for disseminating routing information.

In contrast, we follow the approach in [8] and are interested in an epidemic service where the number of sources (and recipients) is large, such as the population of a highway system. Broadcasting to all in such a setting is simply not possible as the broadcast capacity does not scale with the population [14]. Thus, we must limit the spread of epidemic forwarding if we want sources to have a constant throughput as the network scale. One might argue that this defeats the purpose of an epidemic service, but this is not always true, as there are applications (e.g. coupons, chatting in a jam) for which, in contrast, the semantics of local scope broadcast makes sense. It also follows that we care more for spread (absolute number of nodes that receive a message) than delivery ratio (see Section II for a definition of our performance metrics).

A natural way ("classical TTL") to limit the spread of packets is by means of the TTL field naturally present in IP packets: the TTL of a packet is decremented when it is transmitted, and a packet with TTL==0 cannot be relayed; thus setting a small value at the source (the "max TTL" value) limits the spread. However, as we show later, it is not easy, or even feasible, to find a good value for max TTL, when the number of one hop neighbors varies. Further, in extremely dense cases (such as a traffic jam on highway, with 802.11 range values) even the smallest possible value of max TTL is too large. We consider alternative methods, based on "aging", i.e. decrementing the TTL of a packet while it is waiting for transmission in the epidemic buffer. We are in particular interested in two methods: the former (selective aging) decrements the TTL of a waiting packet by some amount K_1 for every received duplicate of this packet; the latter (global aging) decrements the TTL of *all* waiting packets by some (smaller) amount K_2 for *any* received packet. Note that the TTL field of a transmitted IP packet is an 8 bits integer, but this constraint does not have to apply when the packet is stored in a node, in particular, the aging decrement can be less than 1 (but the decimal part of the TTL is lost when the packet is transmitted). See Section III for more details.

The goal of this paper is to evaluate the performance of these different options for controlling the spread. i.e., when to kill packets. We are in particular interested in knowing whether it is possible to assign constant values to the parameters such that the system works well in a wide range of situations.

We evaluate the performance on networks up to 800 nodes using a Java implementation of the epidemic forwarding system, over a simulated mobile ad-hoc network, using JIST-SWANS (Section V). We also use an analysis method that applies to networks of any size. The method is inspired by the analysis of particle systems. We write the system evolution equations for a Markov process that represents the states of all nodes; we cannot solve this system exactly, (as is classical in such situations), but we use a fluid approximation, which, in this case, can be justified as a scaling limit. The resulting system for the stationary regime is a set of non linear equations. For networks up to 800 nodes we solve the equations numerically, compare the results to those of the simulation, and find a reasonably good match. We also apply the method to analyze networks of unlimited size; here we assume that nodes are regularly spaced over an infinite grid, in 1 or 2 dimensions. We reduce the system of evolution equations to a system of smaller dimensions (order of 100), and we find its stationary point by reducing it to a convergent fixed point problem.

We find the following results. First, we verify that spread control is necessary; without it, a message propagates to the entire network and thus all sources compete for a fixed non scalable capacity. Second, spread control with classical TTL does not perform well. Third, selective aging improves over classical TTL, but global aging performs much better and manages to control the spread so that the rate of injection remains good with fixed parameters across a wide range of settings.

The rest of the paper is organized as follows. In Section II gives our performance metrics. In Section III we describe in detail the options and mechanisms for spread control and other system assumptions. In Section IV we explain our analytical method. The results of our performance analysis are presented in Section V; the state of the art is discussed in Section VI, and Section VII concludes the paper.

II. Performance Metrics

The primary performance metrics we use are:

- *spread* (unitless): the number of nodes that receive a given message
- *spread factor* (unitless : the number of times that a given message is transmitted (cumulated over all nodes)
- *rate of successful injection* (in messages / second): the number of messages per second that a node considers successful (using the implicit acknowledgement method described in Section III)
- *buffer usage* (in messages): number of messages in epidemic buffer, i.e., the transmission of which is pending.

We also use the following derived metrics:

- *geo-coverage* (in m or m^2): the ratio spread / node density. This measures the geographical extent of the epidemic service. The unit depends on the node distribution model. In linear scenarios, the density is in nodes per meter and geo-coverage is in meters; otherwise it is in m^2.

III. System Model

In this section we describe the hypotheses we make about the epidemic forwarding system.

A. General Setting

Every node that participates in the self limiting epidemic forwarding service has an application that generates fresh packets, and an "epidemic buffer", in which packets received from other nodes, or issued by the application at this node, are stored before retransmission. Data is also consumed by the application (e.g. displayed on a screen), but this is not modeled here.

In the context of this paper, the functions of the epidemic forwarding system can be classified as (1) spread control (2) control of the forwarding factor, i.e. scheduling packets from the epidemic buffer for transmission by the MAC layer and (3) adaptation of the rate of injection of fresh packets by the applications. We describe our options for (1) in detail in the next section. Though we focus on the mechanisms for limiting the spread, we still need to make assumptions about (2) and (3); this is done in Section III-C.

B. Spread control

We consider the following alternative methods for limiting the spread of packets.

1) Classical TTL: This is the method that comes by default with the Internet Protocol (IP). When a packet is created by a source and placed into the epidemic buffer, it receives a TTL value equal to some positive constant "max TTL". When the packet is accepted for transmission by the MAC layer, the TTL field of the *transmitted* packet is equal to the value of the TTL field in the packet in the epidemic buffer, minus 1. The TTL field in the packet stored in the epidemic buffer is unchanged.

When a packet created by some other node is received for the first time at this node, the packet is delivered to the application, and the value of the TTL is screened. If it is equal to 0, it cannot be retransmitted and the packet is discarded. Else (TTL$\geq$ 1), the packet is stored in the epidemic buffer, with TTL equal to the value present in the received packet. When and if the packet is later accepted for transmission by the MAC layer, the transmitted TTL field is equal to the stored TTL minus 1, and the stored TTL is unchanged, as above.

There is an issue when a duplicate of an existing packet is received, as is common with epidemic forwarding. In theory, there are 4 possible strategies: (1) keep the existing stored TTL (2) adopt the received TTL (3) keep the largest of the received and stored TTLs and (4) keep the smallest. However strategies (2) to (4) are not practical as they are not resilient to simple bugs or attacks: with (2) and (3) a packet could keep a maximum TTL value for ever; with (4) a system could inject packets with TTL=1 and destroy existing packets. Therefore, only strategy (1) is implemented in practice and this is the one we consider in the rest of this paper.

Classical TTL is used in existing discovery protocols. It has the interesting feature that, if all nodes use the same value for max TTL, they can infer their distances in number of hops.

Used with epidemic forwarding, classical TTL control the spread by limiting the number of retransmissions at successive nodes. Limitation of the number of retransmissions at the same node is performed by the control of the forwarding factor (see Section III-C). We will also see that we assume that nodes control the retransmission of their own packets by examining whether they receive a copy back from some neighbor. For this to work, we need to impose max TTL≥ 2 in the rest of this paper (i.e. every packet should be able to be relayed at least once).

A potential problem, confirmed by our simulation and analysis, is that Classical TTL does not adapt to the degree of connectivity of a node. Without further controls, a packet is multiplied exponentially with a basis roughly proportional to the degree of a node. This degree can vary dramatically without notice (for example on a highway).

2) Classical TTL + Receive Count: This is essentially the method proposed in [17]. In addition to its TTL field, a packet stored in the epidemic buffer has a counter "Receive Count" incremented by 1 when a duplicate of this packet is received. Initially, i.e. when the packet is created by the application or received for the first time, the counter is 0. When the counter reaches a maximum value, the packet is discarded from the epidemic buffer. There is no relation between TTL and Receive Count. In particular, when a packet is transmitted, the value of Receive Count is lost.

This method improves on classical TTL by discarding packets that are probably not worth transmitting. We assume in Section III-C that some mechanism is in place to limit the forwarding factor, so, in practice, a packet that was received a number of times has a very small chance of being accepted by the scheduler. As a result, the only benefit we expect from this method is to reduce the buffer occupancy compared to Classical TTL.

3) Stored TTL: This method is a variant of Classical TTL, which consists in decrementing the stored TTL when there is an indication that some node received the transmission (see Section III-C). This might improve on Classical TTL when nodes are very mobile.

4) Aging: This method was proposed in [8] in a different but essentially equivalent form. We give here a presentation that combines different options in one single framework. The method uses the TTL field like Classical TTL and its variants above, but the TTL of a packet may be decremented while it is stored in the epidemic buffer, depending on receive and send events. Formally, we have the following.

1) Every packet in the epidemic buffer has an "age" field, which is a fixed decimal positive number less than 256. The age field can be coded as a 32 or 64 bit fixed point decimal number; it replaces the stored TTL field. When a packet is created by the application, age $= 0$.
2) When a packet, created by some other node, is received by this node for the first time, its age is set to the complement to 255 of the received TTL:
   age$=255-$ TTL.
3) When a packet is transmitted, its TTL (which must fit in 8 bits) is set to the rounded value of the complement to 255 of the stored age, decremented by a fixed amount K_0:
   $$\text{transmitted TTL} = \lfloor 255 - \text{age} - K_0 \rfloor \tag{1}$$
   When there is an indication that some node received the transmission (see Section III-C), the stored age is incremented in the same way: age = age $+K_0$
4) A packet can be transmitted only if (1) returns a positive TTL; it follows that a packet is discarded from the epidemic buffer whenever its age reaches or exceed 256 $-K_0$.
5) When a duplicate packet is received, the received TTL is ignored but the stored age is incremented:
   age = age$+K_1$
6) When *any* packet is received, the stored age of *all* packets in the epidemic buffer is incremented by K_1:
   age = age$+K_2$

The method has three nonnegative constants K_0, K_1 and K_2. In order to understand perhaps more easily what this method does, assume for a moment that $K_1 = K_2 = 0$, so that only items 1 to 4 above are relevant. In this case, the transmitted TTL decremented by K_0 (and the stored age is incremented by the same amount) when a transmission occurs, and the maximum

TTL set at the source is 255; this is the same as “Classical TTL + Receive Count”, with max TTL approximately equal to $\frac{255}{K_0}$ and the stored TTL equivalent to age.

Back to the general case, the effect of K_1 and K_2 in items 5 and 6 is to increment the age when a packet is received. This is a fundamental difference with Classical TTL and its variants: the age of a packet is affected not only by transmissions of this packet at this node, but also by transmissions from other nodes (this is called “homeostasis” in [8], in reference to biological processes that aim at keeping the size of a population bounded).

We set $K_0 = 25$ in the rest of the paper, unless otherwise specified, which corresponds to a maximum hop count of 8. For K_1 and K_2 we consider in the sequel three special cases:

- *selective aging*: $K_1 > 0$ and $K_2 = 0$. The age of a packet increases as duplicates are received. We consider values of K_1 in the range $25 - 75$, so a packet is eliminated from the epidemic buffer after a few transmissions *or* receptions.
- *global aging*: $K_1 = 0$ and $K_2 > 0$. The age of a packet increases for any reception event. We consider much smaller values for K_2, since we expect the epidemic buffer to hold much more than 1 packet. We consider values of K_2 in the range $0.3 - 1.5$. Note that it is because K_2 is small that we need to introduce the age attribute instead of simply using the stored TTL.
- *hybrid aging*: this is a combination of selective and global aging ($K_1 > 0$ and $K_2 > 0$). We take the same ranges for K_1 and K_2 as above.

We expect aging to be more adaptive than Classical TTL and its variants; for example, in a very dense case, aging should discard packets well before Classical TTL (since transmissions have a large coverage, nodes receive many packets), and the opposite should be true in sparse situations. This is indeed confirmed by our results in Section V. It is less obvious how the three aging options compare. We will see in Section V that they perform similarly, in the sense that for a given network condition and a given value of K_1 there is a corresponding value of K_2 such that selective and global aging performs about the same. However, the optimal correspondence changes with the network condition, and this can be exploited by hybrid aging to perform even better.

Another aging method based on elapsed time is proposed in [8], with the aim to avoid keeping packets forever in situations where connectivity may be delayed a lot (hours or more). We leave the study of this delay tolerance aspect outside the scope of this paper.

C. Other functions

In this section we describe functions that are not the main focus of our analysis but that need to be implemented. We use the model in [8] as a basis as it is the only one, to the best of our knowledge, that was designed for large scales, and other models can be approximately mapped to this one.

1) Control of Forwarding Factor: A packet in the epidemic buffer is retransmitted with a probability that depends on its “virtual rate”; it is equal to $c_0 a^R b^S$ where c_0 is a constant (inverse of a time), R [resp. S] is the number of times this packet or a duplicate was received [resp. sent] and a and b are unitless constants less than 1. Thus the virtual rate of a packet decreases exponentially with any receiving or sending event of the same packet. A scheduler decides which packet is selected next for transmission by the MAC layer; it serves packets per IP source fairly but with a rate not exceeding its virtual rate. The constant c_0 is equal to ηR_0 where R_0 is the nominal bit rate of the MAC layer and η is the fraction of time that the MAC layer spends serving epidemic packets (as opposed to packets of other, non epidemic applications). As a first approximation, the parameter η allows to separate the services; it is outside the scope of this paper to evaluate the impact of self-limiting epidemic forwarding on other services and vice-versa, and therefore, without loss of generality, we take $\eta = 1$. The parameters a and b control the probability that a packet is retransmitted (the “forwarding factor”). We take $a = b = 0.15$ as in [8].

2) Control of Injection Rate: The packets generated by the source at this node are placed into the epidemic buffer, where they compete with the other packets for transmission (but with a high probability of being transmitted since their virtual rate is larger, having $R = S = 0$). The source rate is controlled by a windowing system : the number of outstanding packets the source is allowed to have in the epidemic buffer at this node is limited to at most 2 [8]; a packet is deleted from the epidemic buffer when a duplicate is received, which serves as implicit acknowledgement.

3) MAC Layer Assumptions: We assume the MAC layer is 802.11; in its standard configuration, its broadcast performance is poor, as the RTS/CTS mechanisms (that aims to avoid collisions) cannot be used. We assume that we use the pseudo-broadcast proposed in [13], by which a packet is sent to the MAC address of a neighbor (with RTS/CTS), but can be promiscuously copied by all systems within range. This effectively solves much of the performance issue, but not all due to mobility. We therefore also use the “send pending” method of [8]: a packet is sent in pseudo-broadcast mode if a neighbor’s MAC address was recently refreshed, else it is sent in broadcast mode (without RTS/CTS), but additional heuristics are used to detect if the transmission was successful (indication of neighbor activity, reception of duplicate).

IV. Analysis

A. Dynamical Equations for a Continuous Time Markov Chain

We now describe the analytical method we use. It is driven by the desire to model networks of any size, as is suitable for highway scenarios. The communication system under study could be seen as a dynamical system where local interactions between state variables drive the global behavior of the system to an eventually stationary global state. Each event in the communication system might results in state transitions. When the size of the communication system is large, the cause of possible state transition increases and the system follows approximately a continuous time Markov chain. This is only approximate, and we verify its accuracy against medium scale simulations. The starting point is the following well known result.

Theorem 1 (Forward Equation [3]): Consider a continuous time Markov Chain $\mathbf{Y}(t)$ defined over a finite state space $\mathcal{S}$. For any function $f()$ of the state $\mathbf{y}$:

$$\frac{\partial \mathbb{E}\left\{f(\mathbf{Y}(t))\right\}}{\partial t} = \mathbb{E}\left\{\sum_r h_r(\mathbf{Y}(t))\left(f(\mathbf{Y}(t)+\mathbf{\Delta}_r) - f(\mathbf{Y}(t))\right)\right\}$$

where the summation is over all possible state transitions r, $h_r(\mathbf{y})$ is the rate of transition r when the state vector is $\mathbf{y}$; $\mathbf{y}+\mathbf{\Delta}_r$ is a symbolic notation for the state after transition r has occurred.
The term $\sum_r h_r(\mathbf{y})(f(\mathbf{y}+\mathbf{\Delta}_r) - f(\mathbf{y}))$ is called the drift of function f. When applied to $f() =$ the indicator function of the set $\{\mathbf{y_0}\}$, the forward equation gives the well known equation for the time dependent probabilities, which is often called the "master equation" [9].

In our case the state of the process can be mapped to a subset of $\mathbb{Z}^d$ for some d, and by letting $f(\mathbf{y}) = \mathbf{y_i}$ for $i = 1$ to d one would get equations for the mean of $\mathbf{y}$. However, the right-handside is non linear in $\mathbf{y}$ and a direct use of the Forward Equation is difficult. For large scale systems, a deterministic approximation may work, depending on scaling properties of the system [3], [7]. We show in the next subsection that such a fluid limit exists for a model of our system. The ODE is then obtained from the Forward Equation by removing the expectation, listing all possible transitions and computing their contribution to the drift.

B. An ODE Model of Self-Limiting Epidemic Forwarding

We first derive the forward equation for a model of our system.

1) State Variables: Assume there are N node $i \in \{1, \ldots, N\}$ and there are at most M messages $k \in \{1, \ldots, M\}$ that can exist in this network. For each node i and each message k we have a state vector

$$\mathbf{S}_{ik}(t) = (X_i^k(t), N_i^k(t), M_i^k(t), A_i^k(t))$$

X_i^k is a Boolean equal to 1 if the message k have been received by node i; N_i^k contains the number of received copy of the message; M_i^k contains the number of sent copy of the message; A_i^kcontains the age of the message. We add to these state variables, the value D_i^k that is a Boolean equal to 1 if the message is dead. The death of a message is a function of the age of the packet, $D_i^k = f(A_i^k)$. In all rigour, we should use a function $D_i^k = \mathbb{1}_{\left\{A_i^k > A_{\max}\right\}}$. However, this poses problems for the solution of ODEs. Instead, we approximate the threshold function by a simple linear function that is 0 below $A_{\max} - 1$ and 1 above $A_{\max}$: $D_i^k = \min\left(\max(A_i^k - A_{\max} + 1, 0), 1\right)$.

2) State evolution equations: For the system under study only one event leads to state transitions: the emission of a message by a node. Consider first the case where the number of messages sent over the network are finite; we will relax this assumption in Section IV-C. Let's assume that emission of message k at node i occurs at rate $R_i^k(t)$ (that will be described later) and that this message is received by a node j in neighborhood of i at rate $R_{ij}^k(t) \leq R_i^k(t)$. These rates might be time varying function of the state vector $\mathbf{X}(t)$. In particular they will depend of the level of activity at MAC layer. Obviously $R_i^k(t) > 0$ only when the node i have received the message k ($X_i^k = 1$) and that this message have not died ($D_i^k = 0$). The forward equation is rewritten to derive the set of ODEs governing the dynamic of the system :

$$\frac{\partial \mathbb{E}\left\{X_i^k(t)\right\}}{\partial t} = \mathbb{E}\left\{\mathbb{1}_{\left\{X_i^k(t)=0\right\}} \sum_{j \in \mathcal{N}_i(t)} R_i^k(t)\right\}$$

$$\frac{\partial \mathbb{E}\left\{N_i^k(t)\right\}}{\partial t} = \mathbb{E}\left\{\sum_{j \in \mathcal{N}_i(t)} R_i^k(t)\right\}$$

$$\frac{\partial \mathbb{E}\left\{M_i^k(t)\right\}}{\partial t} = \mathbb{E}\left\{R_i^k(t)\mathbb{1}_{\left\{X_i^k(t)=1,\ D_i^k(t)=0\right\}}\right\}$$

$$\frac{\partial \mathbb{E}\left\{A_i^k(t)\right\}}{\partial t} = \mathbb{E}\left\{ \mathbb{1}_{\left\{X_i^k(t)=0\right\}} \left(\sum_{j\in\mathcal{N}_i(t)} R_{ji}^k(t)(A_j^k(t)+K_0) \right) + \mathbb{1}_{\left\{X_i^k(t)=1, D_i^k(t)=0\right\}} \left(K_2 \sum_{l\in\mathcal{M}} \sum_{j\in\mathcal{N}_i(t)} R_j^l(t) + K_1 \sum_{j\in\mathcal{N}_i(t)} R_j^k(t) + K_0 R_i^k(t) \right) \right\}$$

In these equations we assume that the neighborhood of a node i denoted as $\mathcal{N}_i(t)$ might be variable in time as it will be in the context of node mobility. Coefficient K_0, K_1 and K_2 are coefficients of the aging of packets as defined previously. The above stated equations are very general and valid for all type of MAC layers. Specific MAC layer details will affect the precise form of the rate terms $R_i^k(t)$ and $R_{ji}^k(t)$ as a function of the global state of the communication system.

3) Rate Function: The rate of transmission of message k by node i (R_i^k) is resulting from three effects : the virtual rate $\texttt{vRate}_i^k$, the scheduling mechanism implemented in the node and the MAC layer used. The virtual rate is defined as $\texttt{vRate}_i^k(t) = R_0 a^{N_{jk}^k(t)} b^{M_j^k(t)}$. The scheduler effect could be described easily by the following relation

$$R_j^{k*}(t) = \min\left(\texttt{vRate}_j^k, \frac{\texttt{vRate}_j^k}{\sum_{l\in\mathcal{M}_j} \texttt{vRate}_j^l} \right) \; j \in \mathcal{M}_i$$

where $\mathcal{M}_i$ is the set of messages in node i that are still alive.

The sending rate (R_i^k)as well as the reception rate (R_{ij}^k) will also depend on the characteristics of the MAC layer. One could expect the transmission and reception rate to be related to the overall transmission rate in the neighborhood of the receiver :

$$R_i^k(t) = R_i^{k*}(t) f\left(\sum_{l\in\mathcal{M}_j} \sum_{j\in\mathcal{N}_i(t)} R_j^{l*} \right)$$

$$R_{ij}^k(t) = R_i^{k*} g\left(\sum_{l\in\mathcal{M}_j} \sum_{j\in\mathcal{N}_i(t)} R_{ij}^{k*} \right)$$

where the precise form of the decreasing function $f()$ and $g()$depends on the MAC layer used. As we are assuming in this work an IEEE 802.11 MAC layer, we could use the approximation the broadcast mode of IEEE 802.11 protocol as a p-persistent CSMA mechanism proposed in [5] to approximate f and g. The difference between the IEEE 802.11 protocol and the p-persistent resides in the way of selection of the back-off interval after a collision. Instead of binary exponential back-off as in IEEE 802.11, the p persistent approach samples its back-off interval size from a geometric distribution with parameter p. Probability p depends on the mean size of contention window. In [5] a formula for ρ, the capacity of an IEEE 802.11 network, as a function of p (persistence probability), the number of nodes M in the environment and message size is derived (formula 8 in [5]).This formula is validated with simulations and shown to be a good approximation of real behavior of IEEE 802.11. Moreover a procedure is presented in the same reference to derive the value of the p parameter as a function of source number and a formula is provided evaluating the probability of collision $\mathbb{P}\text{rob}\{\text{c}oll\}$ (in lemma 3) as a function of M and $p(M)$. Using these results the function f and g had been assigned as :

$$g(x) = \min\{1, \frac{\rho(p, M)}{x}\}\ ,\ f(x) = \frac{g(x)}{1 - \mathbb{P}\text{rob}\{\text{coll}\}}$$

4) ODE: In this section we take a fluid limit to replace the equations in Section IV-B.2 by ODEs. The scaling limit is obtained by replacing the original system by one where transitions occur N times faster, but make jumps N time smaller. This amounts to replacing the packets by bits of infinitely divisible fluid. In the limit of N going to ∞ the system becomes deterministic, with evolution equations given by the following theorem.

Theorem 2: Let Y^N be the process corresponding to the scaling parameter N just described, with values in $\mathbb{R}^d$ for some d, and where the state $Y^N(t)$ is the collection of all random variables appearing in the left handside of the equation in Section IV-B.2. Assume the process has some (non random) initial condition $\bar{Y}_0$. Let $\bar{Y}(t)$ be the deterministic, unique solution of the ODE below, ($\bar{Y}(t)$ is the collection of all variables appearing in the left handside) with initial condition $\bar{Y}_0$. Then for very $t \geq 0$

$$\lim_{N\to\infty} \sup_{0\leq s\leq t} \left\| Y^N(t) - \bar{Y}(t) \right\| = 0 \text{ almost surely.}$$

Note that the norm in the theorem can be any norm since we are in finite dimension.

Proof. We use the scaling result in [6], Theorem 6.4 page 456. We need first to imbed the original, non scaled system into $\mathbb{Z}^d$ for some integer d, which is easily done if all jumps (change of states) are integer. Now this is true for the X, N and M components of the state, but not necessarily for the A components because K_0 to K_2 are not necessarily integers. However, they are decimal, so we can always multiply everything by some constant (independent of N) so that all jumps are integer.

Second, we need to verify that

$$\sum_r |\Delta_r| \sup_{\mathbf{y}\in K} h_r(\mathbf{y}) < \infty$$

for any bounded area K, where $h_r(\mathbf{y})$ is the rate of transition r, Δ_r is the change of state that results from transition r, and the transitions are enumerated such that there is one transition per possible Δ_r. But in our case this follows from the fact that, on any bounded area K of the state space, there is only a finite number of such transitions.

Last, the mapping $F()$ defined by the right handside of the equation must be Lipschitz continuous, i.e. for every compact set K there must exist a bound m_k such that $\|F(\mathbf{y}) - F(\mathbf{y}')\| \leq m_K \|\mathbf{y} - \mathbf{y}'\|$ for any $\mathbf{y}, \mathbf{y}' \in K$. A sufficient condition for Lipschitz continuity on compact sets is that the function has a continuous differential (we say it is class C^1). There is a small technicality here because of the $\min$ and $\max$ functions involved in the definition of the R and D terms, which are not differentiable everywhere. We first show that the $\min$ and the $\max$ functions are Lipschitz continuous. This follows from the inequality, valid for any numbers a, b, a', b':

$$|\min(a,b) - \min(a',b')| \leq \max\left(|a-a'|, |b-b'|\right)$$

which shows that the function of two real variables $(a,b) \mapsto \min(a,b)$ is Lipschitz-continuous (with constant 1 with respect to the l^∞ norm in $\mathbb{R}^2$). The same is true for $\max$ as $\max(a,b) = -\min(-a,-b)$. Now it is sufficient to show Lipschitz continuity for each of the coordinates of $F()$ separately, i.e. for any of the lines of the equation below. Each coordinate of $F()$ is a class C^1 function of the state variables and of functions that can be expressed as the minimum or maximum class C^1 functions of the state variable. The desired property follows from the fact that the composition of Lipschitz continuous functions is Lipschitz continuous. □

The ODE used in the theorem is the following:

$$\frac{\partial \bar{X}_i^k(t)}{\partial t} = (1 - \bar{X}_i^k(t)) \sum_{j\in\mathcal{N}^i(t)} \bar{R}_{ji}^k(t)(1-\bar{D}_j^k(t))\bar{X}_j^k(t)$$

$$\frac{\partial \bar{N}_i^k(t)}{\partial t} = \sum_{j\in\mathcal{N}^i(t)} \bar{R}_{ji}^k(t)(1-\bar{D}_j^k(t))\bar{X}_j^k(t)$$

$$\frac{\partial \bar{M}_i^k(t)}{\partial t} = \bar{R}_i^k(t)(1-\bar{D}_i^k(t))\bar{X}_i^k(t)$$

$$\begin{aligned}\frac{\partial \bar{A}_i^k(t)}{\partial t} = {} & (1 - \bar{X}_i^k(t)) \\ & \left(\sum_{j\in\mathcal{N}^i(t)} \bar{R}_{ji}^k(t)\bar{X}_j^k(t)(1-\bar{D}_j^k(t))(\bar{A}_j^k(t)+K_0)\right) + \\ & \bar{X}_i^k(t)(1-\bar{D}_i^k(t)) \left(K_1 \sum_{j\in\mathcal{N}^i(t)} \bar{R}_{ji}^k(t)\bar{X}_j^k(t)(1-\bar{D}_j^k(t)) + \right. \\ & \left. K_2 \sum_{l\in\mathcal{M}} \sum_{j\in\mathcal{N}^i(t)} \bar{R}_{ji}^l(t)\bar{X}_j^l(t)(1-\bar{D}_j^l(t)) + K_0\bar{R}_i^k(t)\right)\end{aligned}$$

where the $\bar{D}$ and $\bar{R}$ variables depend on the deterministic state variables $\bar{X}, \bar{N}, \bar{M}, \bar{A}$ in the same way as as the random variables D, R depend on the X, N, M and A variables.

This dynamical system has two stable equilibrium points. One is when $\bar{R}_j^k(t) = 0$ and the second is when all messages in all nodes die $\bar{D}_j^k = 1$. Two mechanisms drive the system to equilibrium: the message killing mechanism that drives all nodes to the dead state and the inhibition mechanism that drives the transmission rate to 0.

C. Transient and steady state analysis

With the ODE we presented, the transient behavior of the Self-Limiting Epidemic Forwarding could be predicted for any set of parameters and any scenario defined by an adjacency matrix $A(t)$ (that could vary in time because of mobility) and an initial state of the system

$$\mathbf{S}_i^k(0) = (X_i^k(0), M_i^k(0), N_i^k(0), A_i^k(0))$$

for $i = 1, \ldots, N$ and $j = 1, \ldots, M$. (The source of message k is set to node i by setting $(X_i^k(0) = 1, M_i^k(0) = 0, N_i^k(0) = 0, A_i^k(0) = 0)$) However, this analysis could not be used when sources generate a constant flow of messages, as the size of the state space explodes, when we represent every message as an entity. For such realistic situations, the steady state of the system is the main interest. Fortunately we could extend the transient equations to the steady state through the following heuristic.

We now extend the model and assume that a source generate packets with a rate λ. Because of the killing mechanism the lifetime of any clone of a packet is finite. We represent any source of rate λ as a collection of h virtual, parallel mini-sources, each with rate $\frac{\lambda}{h}$. Let's define the virtual system such that, when one of the h mini-sources emits a packet, all previous packets of this mini-source are instantly discarded from all nodes in the network. This is clearly an ideal system; however, if the original system is stable and h is large, the virtual system should be close to the real one, since packets then have a finite lifetime. Let's choose the number h such that the lifetime of a packet in the system is smaller than $\frac{h}{\lambda}$, such that when a new message is generated by a mini-source, all previous messages and clones generated by this source are dead. We need therefore to keep only one state vector (containing 4 variables) per mini-source. Moreover one could observe that all mini-sources situated in the same nodes will converge to the same steady state as they are not distinguishable. This means that for a network of N states and M sources, the steady state analysis needs to solve $4 \times N \times M$ state equations.

Using such a value of h, we now assume that renewals happens with a rate $\frac{\lambda}{h}$ at each mini-source packet. A renewal of a message generated in source k results in a state transition that sets in all nodes the states relative to this mini-source to zero, $(X_i^k = 0, M_i^k = 0, N_i^k = 0, A_i^k = 0)$. Only the source k itself will stay with $X_k^k = 1$. By augmenting the transient equation with the above transition, we can derive a set of ODEs in exactly the same way as before. The steady state can be derived by setting the right-hand side of the system of ODE to 0 leading to the following fixed point equation :

$$
\begin{aligned}
\frac{\lambda}{h}\bar{X}_i^k(\infty) &= (1-\bar{X}_i^k(\infty)) \sum_{j\in\mathcal{N}^i(t)} \bar{R}_{ji}^k(\infty)(1-\bar{D}_j^k(\infty))\bar{X}_j^k(\infty) \\
\frac{\lambda}{h}\bar{N}_i^k(\infty) &= \sum_{j\in\mathcal{N}^i(t)} \bar{R}_{ji}^k(\infty)(1-\bar{D}_j^k(\infty))\bar{X}_j^k(\infty) \\
\frac{\lambda}{h}\bar{M}_i^k(\infty) &= \bar{R}_i^k(\infty)(1-\bar{D}_i^k(\infty))\bar{X}_i^k(\infty)
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
\frac{\lambda}{h}\bar{A}_i^k(\infty) = &(1-\bar{X}_i^k(\infty)) \sum_{j\in\mathcal{N}^i(t)} \bar{R}_j^k(\infty)(\bar{A}_j^k(\infty)+K_0)+ \\
&\bar{X}_i^k(\infty)\left(K_1 h \sum_{l\in\mathcal{M}, j\in\mathcal{N}^i(t)} \bar{R}_j^l(\infty)\bar{X}_j^l(\infty)(1-\bar{D}_j^l(\infty))+\right. \\
&\left.+K_0 \sum_{j\in\mathcal{N}^i(t)} \bar{R}_j^k(\infty)\bar{X}_j^k(\infty)(1-\bar{D}_j^k(\infty))\right)(1-\bar{D}_i^k(\infty))
\end{aligned} \tag{3}
$$

where the ∞ relates to the steady state. The parameter h is chosen such that $\frac{h}{\lambda}$ is larger than the packet lifetime. However, if a too large value of h is chosen it will not change the attained steady state. We can therefore keep increasing h until the obtained steady state converges to a stable point.

D. Scaling behavior

We analyzed so far the dynamic of a network of node containing a finite number of nodes with a known topology (described by the Adjacency matrix $A(t)$). The size of the state space of the dynamical system to analyze was $4 \times N \times M$. Nevertheless we also want to analyze situation when we have millions of nodes that are also potential sources, *i.e.* M and N are very large. Here the proposed steady state analysis is not enough and we need to add some assumptions. Let's assume then that the nodes are sitting over an infinite regular grid with regular connectivity pattern (for example each node is connected with all nodes at a distance less than K hops in the grid) and each node contains a source generating packets with a rate λ (with h mini-sources as before). Because of the regular and symmetric structure of the grid the steady state behavior for any node and any source is identical. The steady state vector of source messages of source k at node i ($\mathbf{S}_i^k(\infty)$) depends only on the distance of the node i to the source node k, *i.e.* $\mathbf{S}_i^k(\infty) = \mathbf{F}(|i-k|)$ where the function $\mathbf{f}$ is a decreasing function converging to 0 that have a value significantly larger than 0 only in a finite number of points. Let's call this number of points L. This shows that the steady state behavior of an infinite grid of nodes and sources can be analyzed through the equation (2) by replacing any term of $\mathbf{S}_i^k(\infty)$ by its counterpart in the function $\mathbf{F}$. This leads to a fixed point system with only $4 \times L$ equations.

E. Parameter Choice

The steady state obtained through the above equations depends on the parameters a, b, K_0 to, K_2. As we explained in the introduction we are not analyzing here the incidence of the inhibition parameters a and b, and we are mainly interested on the effect of K_0 to K_2 and λ.

The transient behavior is expected to highly depends on the value of auto-inhibition and inter-inhibition factors a and b but their impact on the steady state should be lower but it should be mainly governed by the parameters K_1 to K_2 that defines the killing process.

The parameter K_2 has a strong homeostatic property as they control the global number of packets that exists in the network. Let $n(t)$ be the global number of packets in the network. With global aging, the dynamic equation governing this number is $\frac{\partial n(t)}{\partial t} = M_s\lambda - K_2 n(t)$ and at equilibrium we have $n_\infty = \frac{M_s\lambda}{K_2}$ where M_s is the spread factor. We could therefore expect that the overall number of packets in the network to be strongly related to K_2 and λ where a and b affect the spread factor M_s. But this spread factor could be mainly derived using the transient analysis as it is defined as the number of copy sent over the network for any injected packet. We will illustrate this point in the next section.

V. Performance Analysis Results

In this section we describe the main results we find by applying the analytical model as well as detailed simulations.

A. Setting

Our simulations are carried out through JIST-SWANS [2], an open source simulator for ad hoc networks. The MAC layer is a very accurate implementation of 802.11b in DCF mode with the basic rate of 1 Mbps since we transmit in broadcast (pseudo broadcast). As for the radio, we use the capture effect to approach the real WIFI cards, which all implement it [16]. We consider fading channels with free space path-loss.

In our simulations, we consider the model in [8] since it is the only one, to the best of our knowledge, that was designed for wide range of settings (e.g. large scale, sparse, dense, mobility). Further, it supports the aforementioned spread control methods explicitly which makes dealing with them easier.

Beside a static scenario that we use to validate the ODE model, we applied the full protocol of [8] to the vehicular network. We use an extension of JIST-SWANS called STRAW [4], which simulates the vehicular traffic and provides a mobility model based on the operation of real vehicular traffic. We simulate vehicles on a n urban road with tow lanes in each direction and a speed limit of $80Km/h$. We show only results for the two extreme cases: Very sparse and very dense. Corresponding densities are 2.5 $vehicles/Km$ and 400 $vehicles/Km$ respectively. Intermediate results are omitted since they have the same interpretation and they can be deducted by analogy. The transmission range is around 300 m [1], which is typical for outdoor line of site (LOS) communication of vehicular networks.

Each node has a greedy application that injects fresh packets into the epidemic buffer whenever it is allowed (see sect.3.3.3) and thus it achieves the maximum rate allowed by the scheduler according to the network conditions.

B. Spread control is needed

We argued in the introduction that spread control is needed to ensure that the transmission rate of any user is satisfactory. Formally speaking, let λ be the user application rate (generating new information to forward), FF the forwarding factor (the number of time a node forward a message), N the number of nodes receiving messages sent by user application (the spread) and R_0 the available transmission rate over the channel, which includes self packets from node owner application and foreign data coming from other nodes. In a hypothetical symmetric network where self packets are transmitted only once and foreign packets forwarded FF times we have:

$$\lambda + FF * \lambda * N = R_0 \Leftrightarrow \lambda = \frac{R_0}{1 + FF * N}$$

So the rate-spread trade-off is obvious.

One could ask if an aging mechanism is really needed, and if we cannot use just the control of the forwarding factor, in the form of the simple inhibition mechanism described in Section III-C.1 to limit the spread (without any age transferred with the packet in form of a TTL or any other mean). To analyse this, we simulated a linear grid of node each with a single hop connectivity and we used different value of inhibition values a and b to see if for a specific value we would have a limited spread. We do not observe such a limited spread for any value of a and b (unless the trivial case where $a = b = 0$ where the spread is limited to its source). The reason is that without information about the past history of a packet, copies of a message at different nodes are indistinguishable, and therefore the packet will have the same chance to be propagated at any node of the network, as the neighborhood is not different. We show in figure 1 the propagation over time of a single message sent at time $t = 0$ for a node in the middle of a linear grid. A point in this graph means that the message has been received by the node sitting at this x coordinates. One can see that the spread is not limited and that only the speed of the spread is controlled by the inhibition parameters. We need therefore to transmit along with the packet, some information about the past history of packet to make packets that should not be forwarded distinguishable, and eventually to kill packets. In forthcoming sections we evaluate the performance of the different schemes presented in Section III.

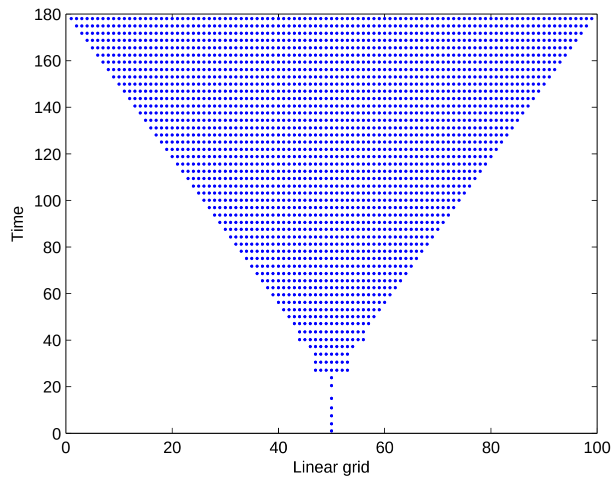


Fig. 1. Propagation of a message in a linear grid with single hop connectivity where nodes use the inhibition mechanism with $a = b = 0.15$.

C. Validation of the Fixed Point Approach

Before going further we have first to validate the ODE model and the fixed point approach described in section IV. Let us first give some idea about the complexity of running the ODE model. As explained previously in section IV-D the scaling steady state is obtained through finding the fixed point of a non linear system of dimension $4 \times L$. Four our analysis we assumed a grid of $N \times N$ nodes leading to $L = N^2$. We have analyzed scenarios with $N = 100$ leading to a system of $L = 10000$ equations. Using Matlab software, finding a stable fixed point for such a system needed around 3 minutes on a single 3 Ghz processor. Alternatively doing one run of simulation using JIST/SWANS for a 800 nodes scenario can take up to 20 hours and should be repeated to get a consistent estimate through averaging. This motivates the use of the theoretical model in place of simulation for large scale systems.

Finding the fixed point state give results as presented in Fig. 2. For all the setting analyzed here a value of h, *i.e.* number of mini-sources equal to 30 was enough to make the system converging to a stable steady state behavior.

However as the theoretical model is based on some simplification we could expect to have some discrepancy between the model and simulation results. To evaluate these difference we have compared performance metric obtained using the ODE model as well as using JIST/SWAN java simulator. We are presenting here one result of this validation tests that is typical.

We have analyzed a rectangular grid of nodes each containing an information source sending at a rate λ. We assume a connectivity of $k = 1, 2, 6$ hops for each node in the grid, *i.e.* if the transmission range is assumed to be constant this means an increase in the density of nodes as k^2. The simulation scenario consists of a grid of 24×24 nodes (because of simulation complexity we have restrained ourself to this network size, as it contains 576 nodes), with parameter values $a = b = 0.15$, $K_0 = 25$ and $K_2 = 1$. The results are compared in Fig. 3. The comparison show a good adequation between the results. We can see that the theoretical model has a tendency to overestimate the spread. This is because of the optimistic MAC layer model we are using (see section IV-B.3). As the number of competing sources increases (for a neighborhood size of 6) the MAC layer model behaves better and the obtained results are tighter for the function R_{ij}^k, but the approximation on collision probability become worse and that explains the discrepancy observed on the spread factor value M_s. There exists also a difference between the value of saturation rate obtained through simulation and what predicted by the theoretical model. This come from the approximative approach that is used to obtain the saturation rate in the theoretical approach.

All in all even if the theoretical model does not give a precise quantitative prediction it is able to capture the qualitative properties of the large scale network. The results presented in this section are typical of what could be expected through the theoretical model. We therefore use the results obtained through the fixed point model to evaluate the large scale properties of Self Limiting Epidemic Forwarding schemes.

We can also validate the homeostasis property described in section IV-E. This validation is done in Fig. 4. The figure shows clearly the relation between spread factor (M_s) and λ as $M_s \propto \frac{1}{\lambda}$. The neighborhood size have a effect on the global number of packet in the network n_∞ as seen on the curve. The Fig.4 shows also despite change in density of node the spread remains constant. If we assume that the communication range of a node is constant, increasing the connectivity pattern from 1 hop to 6 hops means an increase of density. We could therefore expect to have a a geo-coverage decrease when k goes from 1 to 6.

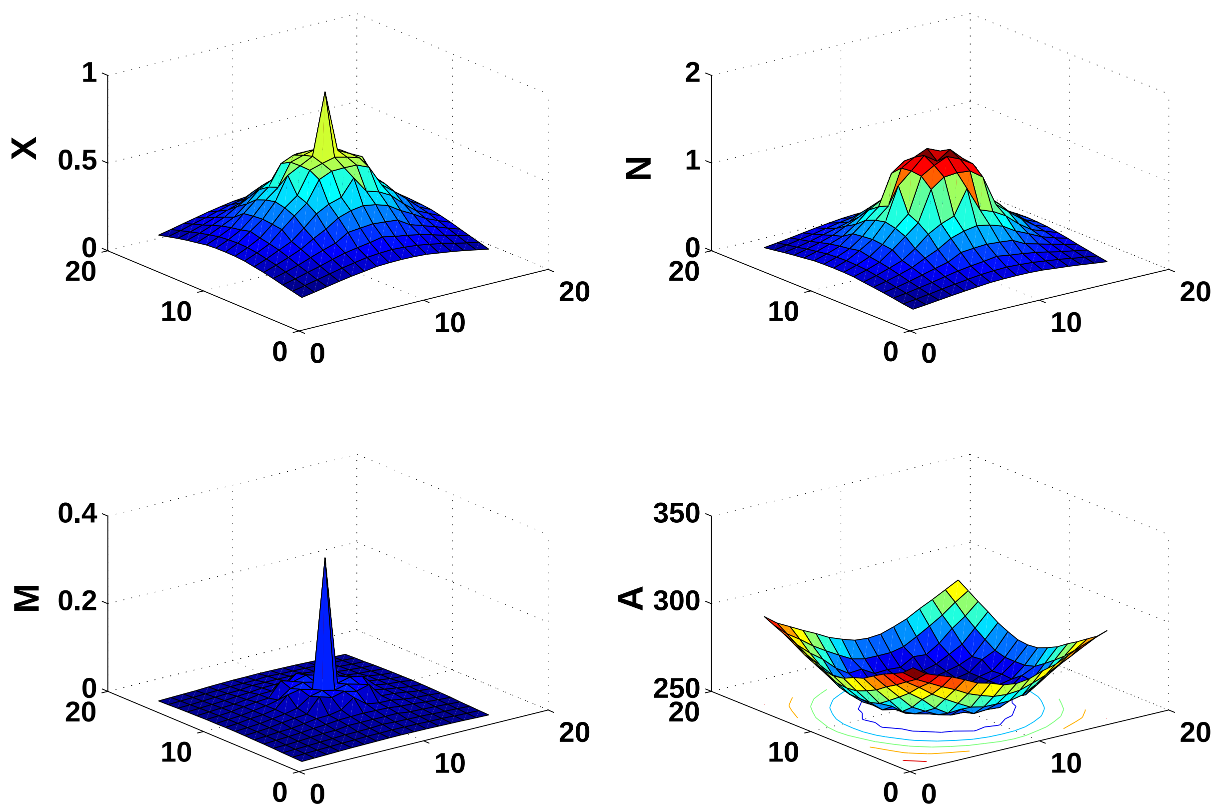


Fig. 2. State values obtained through fixed point approach for a scenario with $K_0 = 30$ and $K_2 = 1$ and $a = b = 0.15$. The surface shown contains clockwise from upper-left, the probability of receiving message send by source in the middle, $\bar{X}_j(\infty)$; the mean number of copy of message received, $\bar{N}_j(\infty)$; the mean number of copy of message sent, $\bar{M}_j(\infty)$; the mean age,$\bar{A}_j(\infty)$. For all figures the x and y axis are node coordinates in the grid

D. Controlling Spread Factor is More Efficient Than Classical TTL

In this section we will first answer the question of sufficiency of Classical TTL to limit the spread of epidemic forwarding and its ability to guarantee at the same time a reasonable transmission rate. For this purpose we have first run two simulation campaigns with two scenario in a vehicular network (see section V-A). In the first scenario we have a sparse highway with a density of 2.5 cars/km; in the second scenario we are in a traffic jam with a high density of cars, 400 cars/km. We have tested over these two scenarios two variants of a forwarding scheme using an inhibition mechanism as described in section III-C.1. In the first variant we show results for the Classical TTL variant called "stored age" as ìt is the one that performs better among all Classical TTL variants (see Section V-B), and in the second variant we are use hybrid aging. We compare the two variants over different values of their parameters. The results are shown in Fig. 5. The comparison of the two variants for the sparse scenario shows that :

- For the classical TTL a geo-coverage of about 2.2 km is reached at a cost of around 300 to 250 sent packets per message (based on TTL decrement). This leads to a buffer size of around 5000 packets and a transmission rate of around 0.7 packets/sec, which increases with higher TTL decrement at each hop.
- The hybrid aging method reaches a transmission rate of 2 pkts/sec and a geo-coverage slightly lower than the classical TTL (1.8 Km) but with a much lower cost: 120 packets sent per message and a buffer size equal to between 100 and 80. The value of the parameter K_2 has an important effect on the queue size as expected (a larger value of K_2 results in faster packet death). The spread factor M_s is not too sensitive to K_2. This is also expected as M_s depend mainly on the parameter a and b.

In the dense scenario with traffic jam, things are even clearer.

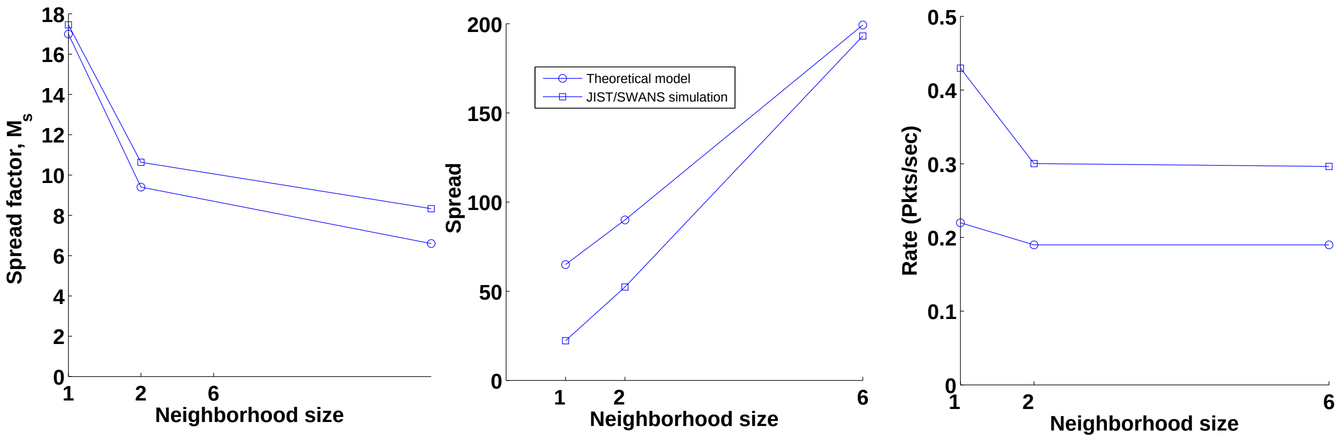


Fig. 3. Comparison of the theoretical fixed Point model with results obtained by JIST/SWANS simulation.

- Classical TTL achieves a very low transmission rate of one packet every 1000 sec, but still consumes 800 transmission for transmitting one packet ! In this situation the geo-coverage falls to 1.1 km.
- The hybrid aging method succeeds in sustaining a larger rate (between 0.1 to 0.3 packets/secs) at the cost of 30 transmissions per message and a buffer occupation that goes from 20 to 10 packets.

This medium scale simulation shows that using a stored age approach is not enough to limit the spread in particular in dense situation, but in comparison the aging approach is able to manage a large spectrum of situation going from sparse to dense scenarios. One could expect that the use of classical TTL will give result that are worse than the stored age as this latter scheme is an improvement over classical TTL.

We also make a simulation of the selective aging scheme. The results are presented in Fig. 6. It can be seen an improvement compared to the classical TTL (or stored age alone) however a larger improvement is achieved through the use of global aging mechanism. This could be explained by the fact that in high traffic the time period between the reception of copies of a particular message can be very large. We also find, as expected from this result, (results not shown) that the difference between global aging and hybrid aging is marginal.

To validate these finding in the large scale we do the same analysis on an infinite two dimensional grid. To simulate the change of neighbor density we do our analysis with two neighborhood size $k = 1$ and $k = 6$. The results are presented in Fig. 7.

The obtained results are qualitatively similar to what we got from JIST/SWANS simulation. It appears that on the infinite grid, the aging mechanism show more robust behavior to density change than the classical TTL.

VI. State of the Art

Epidemic forwarding has been an active area of research during recent years. Most of the authors in this area have identified limiting the spread of diffusion as a major challenge. Different solutions to deal with this problem have been proposed. In [20] a basic epidemic forwarding scheme is presented that use a fixed and predetermined value of TTL to limit the spread of diffusion. The proposed scheme use also content negotiation prior to information diffusion to ensure that redundant information is not sent.

A large class of papers have dealt with the spread of messages through inhibitory mechanisms to reduce the spread factor. As explained in section III-B inhibitory mechanism are not enough to limit the spread and death mechanisms are needed. Nevertheless, reducing the spread through the use of IP TTL is implicit in most epidemic forwarding papers and could be

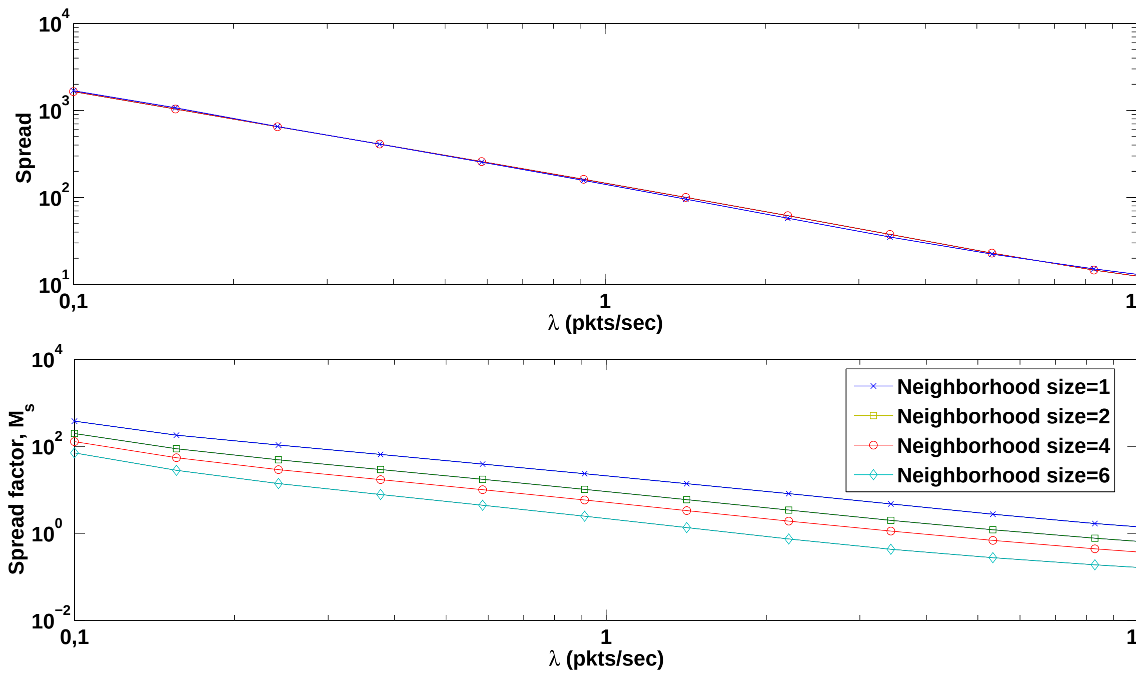


Fig. 4. Performance metrics (spread and Spread factor) obtained by fixed point method for an infinite grid with different connectivity pattern ($k = 1, \ldots, 6$) as function of λ the injection rate of packets . In all these plots a fixed value of $K_0 = 25$ and $K_2 = 1$ is used.

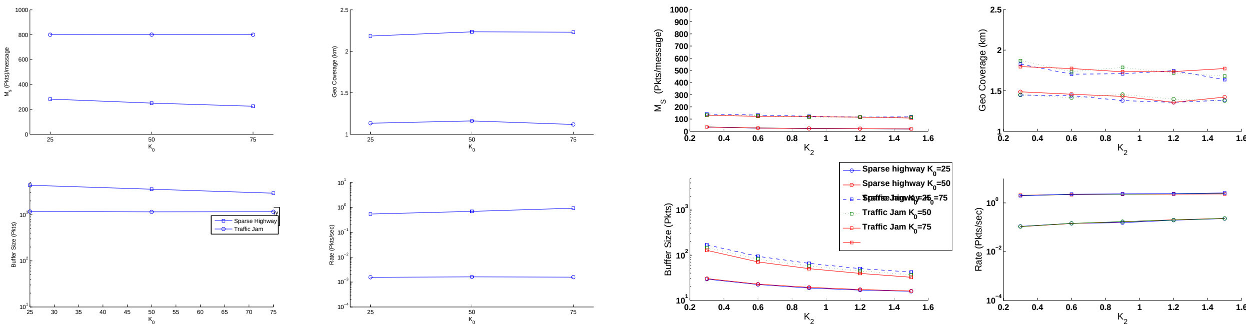
used in these papers. In [19] the forwarding factor is reduced through defining localized dominating sets that will ensure that messages are broadcast only when the list of neighbors that might need the message is not empty. However this scheme assumes that each node has knowledge of the position of its neighboring nodes. In [11] a probabilistic forwarding scheme named Gossip-based is presented. This scheme forwards a received packet with a fixed probability. Adaptive versions of Gossip-based forwarding are also proposed that adapt the forwarding probability to neighborhood size, in order to prevent the spread from dying. In [21] a broadcast mechanism is presented that uses two neighborhood coverage conditions to decide if a node should forward a message. The neighborhood coverage conditions are based on k-hop connectivity information that are hard to gather in a mobile environment. In [17] an inhibition mechanism is proposed that prevents a packet from being forwarded if a fixed number of copy of this same packet have been received.

In [15] an epidemiological model is applied to the study of a simple information diffusion mechanism. The approach followed in this paper bears similarity with ours. However, we present in this paper a more methodological derivation of the ODEs that could be easily extended to other scenarios, whereas the approach followed in [15] is a direct fitting of an epidemiological model to a simple single source problem.

As self limiting epidemic forwarding supports also mobility, it is related to the Delay Tolerant Network literature. However, two points are frequently cited in the DTN literature that reduce the interest for epidemic mechanism in DTNs. First, buffer overflows and inefficient use of the transmission media that is inherent to simplistic epidemic forwarding have discouraged its application. Epidemic forwarding has only been proposed in [12] in the context of sparse network where TTL based spread control is still efficient. In this paper a transport layer overlay architecture for sparse mobile networks is proposed that use a gossip-based inhibition mechanism. The goal of this paper is similar to our aims however we develop here a larger perspective on spread control that could be used in the context of DTNs.

VII. Conclusion

We have analyzed the performance of various spread limitation mechanisms for epidemic forwarding. We found that solely relying on the classical TTL mechanism is not adequate; in contrast, an aging mechanisms that decrements the TTL field of stored packets by small amounts based on packet receptions is needed. Among the variants of aging, global aging performs well.



(a) (b)

Fig. 5. Comparison of the performance of stored TTL schemes with hybrid aging. The results are obtained over a sparse scenario over a highway with a density of 2.5 cars/km and over a traffic Jam in a highway with a density of 400 cars/km using the JIST/SWANS simulator for different value of aging parameters.

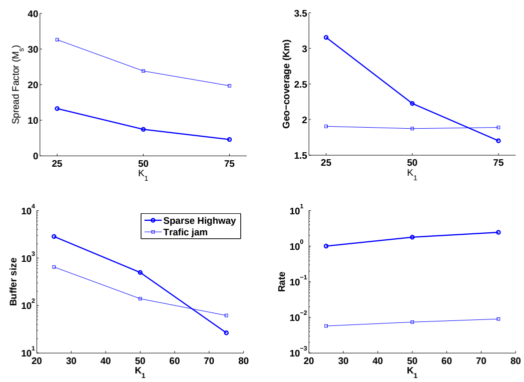


Fig. 6. Performance of selective aging obtained by JIST/SWANS simulation.

REFERENCES

[1] Cisco aironet 350 series client adapters.

[2] Java in simulation time / scalable wireless ad hoc network simulator , jist/swans, http://jist.ece.cornell.edu/.

[3] *Reference suppressed for double blind reviewing*.

[4] Street random waypoint / vehicular mobility model for network simulations , straw, http://www.aqualab.cs.northwestern.edu/projects/straw/.

[5] F. Cali, M. Conti, and E. Gregori. Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Trans. Netw.*, 8(6):785–799, 2000.

[6] V. Capasso and D. Bakstein. *An Introduction to Continuous-Time Markov Processes*. Birkhäuser, 2005.

[7] R. Durrett. *Ten Lectures on Particle Systems*, volume 1608. Springer Verlag, Lecture Notes on Mathematics, 1995.

[8] A. El Fawal, J.-Y. Le Boudec, and K. Salamatian. Self-Limiting Epidemic Forwarding. Technical Report LCA-REPORT-2006-126, EPFL, 2006.

[9] C. W. Gardiner. *Handbook of Stochastic Methods*. Number 56. Springer-Verlag, 1985.

[10] A. Garyfalos and K. Almeroth. Coupons: Wide scale information distribution for wireless ad hoc networks. In *IEEE Global Telecommunications Conference (Globecom) Global Internet and Next Generation Networks Symposium Dallas, Texas, USA*, pages 1655–1659, December 2004.

[11] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. *IEEE/ACM Transactions on Networking*, 14(3):479–491, 2006.

[12] K. Harras, K. Almeroth, and E. Belding-Royer. Delay tolerant mobile networks (dtmns): Controlled flooding in sparse mobile networks. In *IFIP Networking Conference, Waterloo, Ontario, CANADA*, May 2005.

[13] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard. The importance of being opportunistic: Practical network coding for wireless environments. In *Allerton conference*, 2005.

[14] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi. Broadcast capacity in multihop wireless networks. In *Mobicom '06*, September 2006.

[15] A. Khelil, C. Becker, J. Tian, and K. Rothermel. An epidemic model for information diffusion in manets, 2002.

[16] A. Kochut, A. Vasan, A. U. Shankar, and A. Agrawala. Sniffing out the correct physical layer capture model in 802.11b, berlin, germany. In *IEEE International Conference on Network Protocols (ICNP 04)*, pages 252– 261, October 2004.

[17] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Mobicom, Seattle, Washington, United States, August 15 - 19, 1999*, pages 151–162.

[18] J. Scott, P. Hui, J. Crowcroft, and C. Diot. Haggle: A networking architecture designed around mobile users. In *IFIP WONS 2006, January 18-20, Les Menuires, France*.

[19] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, Jan. 2002.

[20] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks, 2000.

[21] J. Wu and F. Dai. Broadcasting in ad hoc networks based on self-pruning.

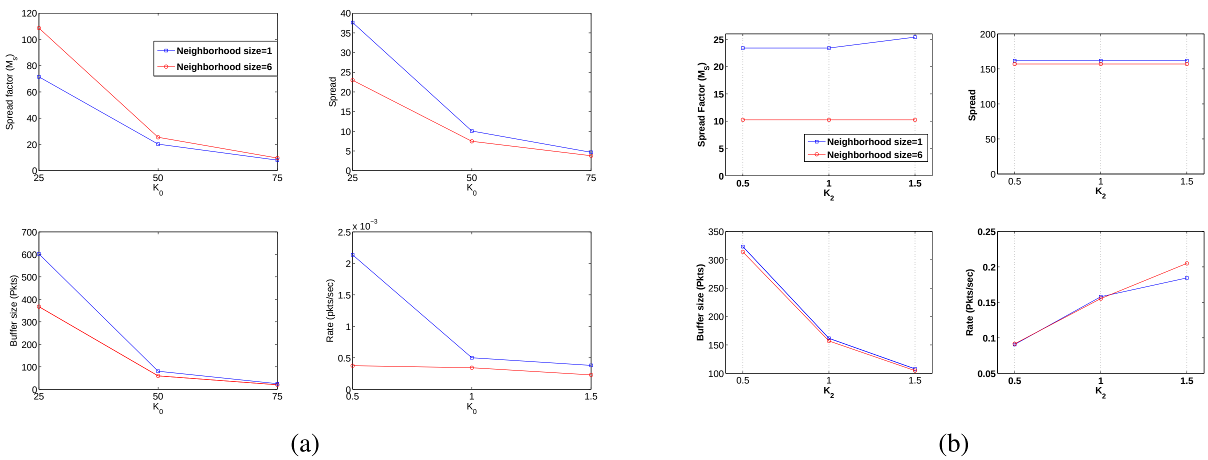


Fig. 7. Comparison of the performance of stored TTL schemes with global aging method. The results are obtained using the fixed point approach for two scenarios. The first with a neighborhood size of 1 and the second with a neighborhood size of 6.