Effective Dissemination of Presence Information in Highly Partitioned Mobile Ad Hoc Networks

Christoph Lindemann and Oliver P. Waldhorst
University of Leipzig
Department of Computer Science
Augustusplatz 10-11
04109 Leipzig, Germany
http://rvs.informatik.uni-leipzig.de/

Abstract— **Presence technology enables users of an instant messaging (IM) system to determine if their contacts are online and ready to communicate. In this paper, we propose an effective approach for the proactive dissemination of frequently changing presence information in highly partitioned mobile, wireless networks with IEEE 802.11 technology. Although communication techniques for intermittently connected networks have been extensively studied in the field of delay tolerant networking, the fact that presence information is highly delay sensitive requires a thorough revision of these techniques. To this end, we use discrete-event simulation based on a high-level stochastic model of the IM system to compare different approaches for disseminating presence information in terms of sustained consistency (i.e., fraction of time presence information is in a coherent state) and traffic requirements. Building upon the outcome of the simulation study, we propose the System for Presence information Exchange by Epidemic Dissemination (SPEED). Results of a detailed ns-2 simulation study show that SPEED outperforms an approach based on optimized flooding by up to 20% in terms of sustained consistency for low node density and saves up to 48% of control traffic for medium to high node density.**

Keywords- Novel applications in ad hoc networks, protocols for data dissemination, delay tolerant networking, modeling and performance evaluation

I. Introduction

Instant messaging has long been one of the Internet's most popular consumer applications, thanks to its combination of real-time communication, the ability to know in advance if people are available to chat, and the ease of use. A basic instant messaging (IM) system consists of three components: (1) an authentication component, which determines if clients trying to enter the system are permitted to do so, (2) a presence component, that identifies which contacts are available for messaging, and (3) a messaging component, which handles the communication process. Presence technology constitutes an integral part of an IM system, since it lets users determine if their contacts are online, signed onto the IM applications, and ready to communicate. Presence technology is used in a number of applications other than classical IM systems, e.g., in numerous computer-supported-cooperative-work applications. In fact, the 2003 release of Microsoft's Office constitutes one of the most prominent examples of a presence-enabled application.

The protocol design for disseminating presence information in the Internet has been matured and organizations such as the IETF and the Jabber software foundation have developed the protocol proposals SIMPLE [1] and XMPP [15]. Note that it is possible to request presence information on demand in a reactive fashion. However, all state-of-the-art IM systems display presence information proactively.

Due to the lack of fixed infrastructure, dynamic network topology, and intermittent connectivity the dissemination of presence information in mobile ad hoc networks (MANET) poses a challenging research problem. In particular in highly partitioned MANET, well-studied MANET routing protocols cannot be used for communicating presence information, since no route between contacts in different partitions exists. Approaches for enabling message-based communication in intermittently connected networks have been studied in the field of Delay Tolerant Networks (DTNs). While these approaches certainly enable the transmission of messages in the IM system, presence information is highly delay-sensitive. Thus, DTN techniques have to be extended to achieve message delays suited for this particular application.

In this paper, we extend DTN routing approaches for proactively disseminating presence information in highly partitioned MANET. On the first view, it seems questionable if the timely propagation of up-to-date presence information in such networks is useful, if message-based communication is prone to large delays due to employing a DTN message relaying protocol. However, we argue that up-to-date presence information is always valuable, since it enables a user to know in advance if a contact is ready to communicate using either the DTN protocol or *any other* communication technology, including more expensive or energy consuming radio technologies with larger transmission ranges or even a traditional telephone call.

We introduce sustained consistency as a novel performance measure that quantifies the fraction of time in which a node has a coherent view of the presence state of its contacts. Though this measure is designed to characterize the usefulness of an approach for disseminating presence information in an instant

This work was partially supported by the German Research Council (DFG) under Grant Li-645/13-1.

messaging system, it also implicitly measures the speed of the dissemination process, since faster dissemination means higher sustained consistency. As further performance measure of interest, we analyze the control traffic required for dissemination of presence information. To compare different design alternatives, we conduct discrete-event simulation based on a high-level stochastic model of the IM system and compute sustained consistency and control traffic. Building upon the results, we propose SPEED, the System for Presence information Exchange by Epidemic Dissemination in MANET. Similar to DTN routing approaches, SPEED transmits presence information epidemically using relay nodes. To optimize performance, SPEED estimates the current node density in the system by counting the direct neighbors of each node and adaptively adjusts system parameters according to these estimates.

To illustrate the effectiveness of SPEED, we conduct a detailed simulation study using the network simulator ns-2 [7] and compare the performance of SPEED to the results of the high-level simulation study and to an optimized approach based on periodical flooding of presence information. The simulation study reveals that for low node density SPEED achieves up to 20% higher sustained consistency than the flooding approach. For higher node density, SPEED saves up to 48% control traffic compared to the flooding approach while providing equal sustained consistency. Compared to the high-level simulation results the presented configuration of SPEED achieves high traffic efficiency at the expense of reduced sustained consistency. However, sensitivity studies show that SPEED can be easily configured to achieve higher sustained consistency at the expense of increased control traffic.

The remainder of this paper is organized as follows. Section II summarizes related work on dissemination of information in highly partitioned MANET. In Section III, we introduce sustained consistency as a measure for the coherence of state information displayed by an IM system. Different approaches for disseminating presence information are compared in terms of sustained consistency and control traffic in Section IV. Building upon this comparison, Section V presents the design of SPEED. In Section VI, a detailed comparative performance study of SPEED using the network simulator ns-2 is presented. Finally, concluding remarks are given.

II. Related Work

Although traditional IP routing assumes the existence of an end-to-end path between the source and the destination, emerging network technologies create many applications where the assumption of permanent end-to-end connectivity does not hold. Examples constitute mobile networks with intermittent connectivity due to interferences and node mobility, disaster recovery, as well as satellite and interplanetary networks. Although direct communication is generally impossible, it has been observed that such networks are well suited for messaging applications that can tolerate a certain delay, e.g., email. This observation founded the research area of Delay Tolerant Networks (DTNs). Fall proposed a general-purpose overlay architecture to enable message-based communication in DTNs [2]. Based on this architecture, Jain, Fall, and Patra developed a framework for design and evaluation of delay tolerant routing algorithms [6]. The framework was extended to multicast routing by Zhao, Ammar, and Zegura [20].

It is easy to see that sparse MANET constitute a special case of DTNs. In such networks, messages can be delivered by exploiting node mobility. In general, a source node transmits a message to some relay nodes, which transmit the message to the destination on a direct encounter. Some approaches try to modify the trajectories of relay nodes [10] or use relay nodes with known trajectories [9], [19] to increase message delivery probability and decrease message delay. However, in most application scenarios nodes move autonomously and their trajectories are unknown or random. For such scenarios, Grossglauser and Tse showed that using a single-copy routing algorithm (i.e., an algorithm that uses a single relay node per message) the average throughput per source-destination pair can be kept constant for an increasing density of nodes [4]. De Moraes, Sadjadpour, and Garcia-Luna-Aceves showed that this property still holds for multicopy routing algorithms (i.e., algorithms that use several relay nodes per message), while with such algorithms the message delay is linear or quadratic in the number of nodes [13]. Groenevelt, Nain, and Koole provided closed-form expressions for the expected message delay for two different multicopy routing algorithms [3]. Besides these theoretical results, several routing algorithms have been proposed. Vahdat and Becker developed a multicopy algorithm denoted as epidemic routing, that floods the network with messages until the final destination is reached [17]. The overhead of epidemic routing was reduced by Harras, Almeroth, and Belding-Royer, who proposed to control flooding by time-to-life information and added a mechanism called passive cure to stop flooding after successful message delivery [5]. Widmer and Le Boudec reduced overhead by network coding of messages [18]. Nain, Petigara, and Balakrishnan developed the Mobile Relay Protocol (MRP), that can be used on top of an existing MANET Routing protocol and employs DTN routing when direct routing fails [14]. With MRP a message is delivered from a relay node to the destination as soon as a multihop path of length d to the destination exists. Although in general it may hold $d > 1$, only $d = 1$ is considered in the evaluation presented in [14].

Recall that an instant messaging system combines message delivery capabilities with presence technology. Obviously, for implementing message delivery in a highly partitioned MANET any routing algorithms for mobile DTNs can be used. Unfortunately, presence information is not delay tolerant, but highly delay sensitive. That is, DTN routing algorithms that provide a message delivery delay in the order of minutes cannot be employed for disseminating presence information. In this paper, we investigate more aggressive multicopy algorithms that trade message delay for network traffic. These algorithms deliver messages to destinations over multiple hops similar to [14]. Furthermore, opposed to all DTN routing protocols, they copy messages to relay nodes over multiple hops, too.

To discover the existence of multi hop routes, our system proposal, SPEED, uses a hybrid approach combining pushing and polling control messages. Hybrid pushing / polling has been proposed by Mohan, Almeroth, and Belding-Royer in the context of service discovery in dense MANET [12]. Opposed to [12] we consider sparse MANET and complement the pushing / polling scheme by caching of presence information. In addition to the hybrid push/poll approach, SPEED communicates presence information by an approach called directed flooding, that aggregates redundant presence information and transmits them in the direction of a polling node. Related approaches have been proposed in the context of wireless sensor networks, e.g., the directed diffusion approach by Intanagonwiwat, Govindan, and Estrin [8]. However, since sensor networks constitute dense and mostly static networks, such approaches cannot be employed directly in the considered scenarios.

III. COMPUTATION OF SUSTAINED CONSISTENCY

In this section, we introduce sustained consistency as performance measure for evaluating the effectiveness of different communication paradigms for proactively disseminating the presence state of a mobile node. Consider N mobile nodes which have subscribed to an IM system. We denote mobile nodes by the variables n, m, k, and l, with $1 \leq n,m,k,l \leq N$. We derive a discrete-event stochastic system as a model for describing the dynamics of an IM system. The model comprises two stochastic processes: the mobility model and the system model. The location of each mobile node within a geometric area $\mathbb{A} = a_1 \times a_2$ is determined by the mobility model. The mobility model is basically a continuous-time, continuous-state stochastic process $(X_n(t),Y_n(t))$, $0 \leq X_n(t) \leq a_1$, $0 \leq Y_n(t) \leq a_2$. That is, at an instance of time t the mobility model associates a location $(X_n(t),Y_n(t))$ with each mobile node n. The mobility model interacts with the system model by determining whether two nodes can communicate at a given instance of time t. Assuming a homogenous transmission range R, nodes n and m can communicate directly (i.e., in one hop) at time t, if and only if holds $\left\| \left(X_n(t), Y_n(t) \right), \left(X_m(t), Y_m(t) \right) \right\| \leq R$ for the Euclidian distance $\|\cdot\|$. Note that this constitutes a best-case assumption, since a transmission between two nodes may fail due to various reasons, including the impact of the medium access protocol, interference, and wireless errors.

To incorporate multihop communication in our model, we define a distance matrix $D(t)$ for an instance of time t. $D(t)$ is a random matrix with dimensions $N \times N$ and $D_{n,m}(t) \in \{0,1,\ldots,N\text{-}1,\infty\}$ determined by mobility model. Here, $D_{n,m}(t)$ constitutes the length l of the shortest sequence of nodes connecting nodes n and m, $n \neq m$. We set $D_{n,m}(t) := \infty$ if no such sequence between n and m exists and $D_{n,n}(t) := 0$. Note that $D(t)$ describes a continuous-time discrete-state stochastic process that is controlled by the mobility model.

For the system model, we assume that at an instance of time t a node n is in a presence state $S_n(t)$ taking values in a set $\mathbb{S}$, e.g., $\mathbb{S}$ = {*online*, *busy*, *do-not-disturb*, *away*}. Besides its own presence state, a mobile node n stores information about the presence states $R_{n,k}(t)$ of all other nodes k, $k \neq n$. Presence states $R_{n,k}(t)$ take values in $\mathbb{S} \cup$ {*undefined*}, where *undefined* denotes that the presence state of node k is unknown at node n. One can think of $R_{n,k}(t)$ as a cache for storing remote information that can be used by the communication approach. Note that at an instance of time t it may hold $R_{n,k}(t) \neq S_k(t)$. Putting it together, at any instant of time t, a mobile node n can be completely described by a set of random variables:

$$\left(S_n(t), R_{n,1}(t), R_{n,2}(t), \ldots, R_{n,n-1}(t), R_{n,n+1}(t), \ldots, R_{n,N}(t) \right) \quad (1)$$

State changes in the system model are triggered by one of the following three types of events:

Presence state change: A mobile node changes its presence state after some dwell time has elapsed. A state change event that occurs for node k at an instance of time t changes the state of the system by changing $S_k(t)$. Additionally, it may trigger one or more communication events that are described below.

Elapsed Timer: The presence communication approach of an IM system may perform periodic communication tasks triggered by timers. Depending on the communication tasks, several timers with different parameters may be active concurrently. In general, the occurrence of a elapsed timer does not change the state of the system directly, but triggers one or more communication events.

Communication: Each communication operation used by the IM system is decomposed into one or more communication events, regardless if it is one-to-one or one-to-many communication. A communication event constitutes the information exchange between two nodes. A communication event that occurs between nodes n and k at time t changes the state of the system by changing either $R_{n,k}(t)$ or $R_{k,n}(t)$. Whether a communication event between nodes n and k occurs is determined by both the presence communication approach of the IM system and the distance matrix $D_{n,k}(t)$.

Presence state change events constitute stochastic events. For all mobile nodes, the dwell times in a presence state are drawn according to the same probability distribution with finite mean and variance. Elapsed Timer events occur exactly after a certain period of time has elapsed. Thus, the time until the occurrence of a timeout event is chosen according to a deterministic distribution. Since communication takes place on the timescale of milliseconds while presence state changes take place on the timescale of minutes, we consider communication events as immediate events occurring in zero-time.

We denote the instances of time at which presence state changes of $S_k(t)$ take place by $T_k(1)$, $T_k(2)$,…. That is, $T_k(i)$ constitutes the instant of time of the i-th state change of the presence state of mobile node k. A presence state change of node k occurring at time $T_k(i)$ may be perceived at a node n due to communication events. We denote the instance of time at which the i-th presence state change of node k is perceived at node n as $\hat{T}_{n,k}(i)$. With an optimal communication approach, we have $\hat{T}_{n,k}(i) = T_k(i)$. That is, node n has a coherent view of the presence state of node k. However, in a highly partitioned MANET typically holds $\hat{T}_{n,k}(i) > T_k(i)$ or even $\hat{T}_{n,k}(i) > T_k(i+1)$ if the i-th presence state change of k is never perceived by node n. For convenience in the notation of Equations (2) to (5), we set $\hat{T}_{n,k}(i) = T_k(i+1)$ in this case.

In an IM system, the current state of node n is displayed only at a subset of all other nodes denoted as the contacts of node n. Hence, each mobile node n has assigned a number c_n of contacts drawn from some probability distribution. Given c_n, the set of contacts $\mathbb{C}_n \subseteq \{1,2,\ldots,n\text{-}1,n\text{+}1,\ldots,N\}$ is also determined according to some probability distribution with $|\mathbb{C}_n| = c_n$. For ease of exposition, we assume that for each node the set of contacts does not change within the observation period. Obviously, the goal of a communication approach for disseminating presence information in MANET constitutes the maintenance of a coherent view of the presence state of a node n at all its contacts $k \in \mathbb{C}_n$. To evaluate the effectiveness of a particular communication approach, we define the sustained consistency for a node n with respect to contact k, denoted by $C(n,k)$, as the fraction of time that its presence information is coherent at contact k:

$$C(n,k) = \sum_{i=1}^{\infty} \frac{T_n(i+1) - \hat{T}_{k,n}(i)}{T_n(i+1) - T_n(i)} \qquad (2)$$

The sustained consistency of node n, denoted by $C(n)$, is determined as the sustained consistency with respect to an individual contact k given in (2) averaged over all contacts. Thus, we have:

$$C(n) = \sum_{k\in\mathbb{C}_n} C(n,k) \Big/ |\mathbb{C}_n| \qquad (3)$$

Note that sustained consistency of node n is only determined by the remote state information $R_{k,n}(i)$ and $\hat{T}_{k,n}(i)$ stored at contacts $k \in \mathbb{C}_n$. Since it is more important to maximize sustained consistency for nodes with many contacts, we define in (4) the overall sustained consistency, denoted by C_{System}, as the sum of the sustained consistency for each node, $C(n)$, weighted by the number of contacts of node n. That is:

$$C_{System} = \sum_{n=1}^{N} |\mathbb{C}_n| C(n) \Big/ \sum_{n=1}^{N} |\mathbb{C}_n| \qquad (4)$$

In the infinite time horizon, a mobile node n changes its presence state infinitively often. In the remainder of this paper, we consider a finite observation interval (0, T]. Let the counting process $K_n(t)$ denote the number of presence state changes of mobile node n during the time interval (0, t]. In the observation interval (0, T], mobile node n makes $K_n(T)$ presence state changes. We compute sustained consistency in the observation interval (0, T] for a node n with respect to contact k, denoted by $C(n,k,T)$, by summing up over i, $1 \le i \le K_n(T) - 1$ in (2). Analogous to (3) and (4), we can compute overall sustained consistency in time T denoted as $C_{System}(T)$ by averaging $C(n,k,T)$ over all contacts and nodes, respectively. That yields:

$$C_{System}(T) = \sum_{n=1}^{N} \frac{\sum_{k\in\mathbb{C}_n} \sum_{i=1}^{K_n(T)-1} T_n(i+1) - \hat{T}_{k,n}(i)}{\sum_{n=1}^{N} |\mathbb{C}_n| \left(T_n(K_n(T)) - T_n(1)\right)} \qquad (5)$$

IV. Approaches For Communicating Presence Information

The goal of this section is to analyze different approaches for disseminating presence information in highly partitioned MANET. Recall that presence information is highly delay sensitive and using DTN routing algorithms for presence information dissemination will induce a significant message delay, leading to unacceptable performance in terms of sustained consistency. Thus, we analyze multicopy approaches that are more aggressive than DTN routing algorithms. That is, presence information is not only copied from / to relay nodes on a direct encounter, but also when the nodes are connected via multihop routes. The approaches considered in this section differ in the selection of the set of relay nodes for the i-th presence state of node n, denoted as $\mathbb{R}_n(i)$, as well as in the way presences information is copied between the source node, relay nodes, and the destination node, respectively. For the description of the algorithms, we denote the presence state of node n at time t as $s_n(t)$ and the remote information stored about the state of node n at node k as $r_{k,n}(t)$. For ease of exposition, we define $r_{n,n}(t) := s_n(t)$, $\bar{\mathbb{C}}_n(t) := \{m \mid r_{m,n}(t) = s_n(t)\}$, and $\mathbb{H}_n(t) := \{k | D_{n,k}(t) < \infty\}$.

We consider the following algorithms:

Approach 1: With this approach, presence information is copied from a source node n to a contact $k \in \mathbb{C}_n$ when a multihop route is available. That is, no relays are used, i.e., $\mathbb{R}_n(i) = \varnothing$, and it holds:

$$\hat{T}_{k,n}(i) = \min\{t \mid T_n(i) \le t < T_n(i+1) \wedge D_{n,k}(t) < \infty\}$$

Approach 2: With this approach, the source node n copies its current presence state to a contact $k \in \mathbb{C}_n$ when a multihop route is available. Additionally, two nodes $k,l \in \mathbb{C}_n$ exchange most recent information about the shared contact n when a multihop route is available. That is, $\mathbb{R}_n(i) = \mathbb{C}_n$ and it holds:

$$\begin{aligned}\hat{T}_{k,n}(i) = \min\{t \mid T_n(i) \le t < T_n(i+1) \\ \wedge \exists m \in \bar{\mathbb{C}}_n(t) : D_{m,k}(t) < \infty\}\end{aligned}$$

Approach 3: Building upon Approach 2, a source node n additionally copies its presence state immediately after the i-th presence state change to all nodes $m \in \mathbb{H}_n(T_n(i))$ to increase the number of relay nodes. Copying is performed using a flooding operation. The additional relay nodes $m \in \mathbb{H}_n(T_n(i))$ transmit the presence information to contacts $k \in \mathbb{C}_n$ of the source node n when a multihop route is available. That is $\mathbb{R}_n(i) = \mathbb{C}_n \cup \mathbb{H}_n(T_n(i))$ and it holds:

$$\begin{aligned}\hat{T}_{k,n}(i) = \min\{t \mid T_n(i) \le t < T_n(i+1) \wedge \\ \exists m \in \bar{\mathbb{C}}_n(t) \cup \mathbb{H}_n\left(T_n(i)\right) : D_{m,k}(t) < \infty\}\end{aligned}$$

Approach 4: With this algorithm, two nodes synchronize all available contact information when a multihop route is available. With this algorithm, each node acts as relay for the presence information of each other node. That is $\mathbb{R}_n(i) = \{1,\ldots,n\text{-}1,n\text{+}1,\ldots,N\}$ and it holds:

$$\begin{aligned}\hat{T}_{k,n}(i) = \min\{t \mid T_n(i) \le t < T_n(i+1) \\ \wedge \left(\exists m \in \{1,\ldots,n\} : r_{m,n}(t) = s_n(t) \wedge D_{m,k}(t) < \infty\right)\}\end{aligned}$$

To compare the approaches, we compute the sustained consistency and the control traffic using a simulation model of the discrete event stochastic system described in Section III. To compute the control traffic, we assume that the traffic for the

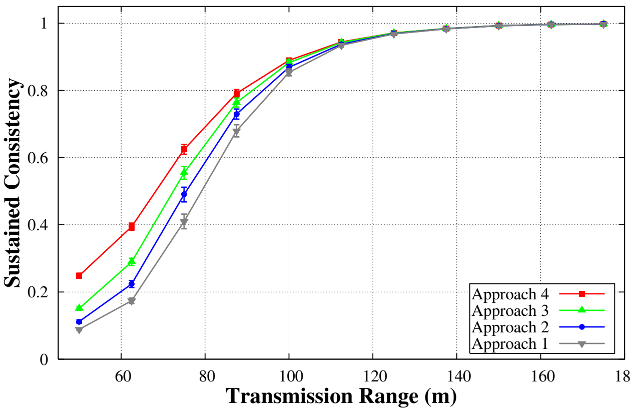

Figure 1. Sustained consistency achieved by the different approaches

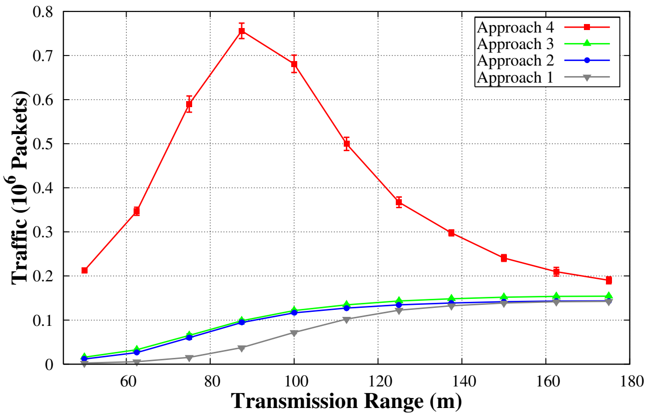

Figure 2. Control traffic generated by the different approaches

initial flooding the new presence state after the i-th state change of node n at time $T_n(i)$ to all reachable nodes is given by the number of reachable nodes, i.e., $|H(T_n(i))|$. Additionally, we assume that the traffic used for a successful exchange of state information in subsequent operations is equal to the number of hops on the shortest path between the communicating nodes. That is, if node n sets $r_{n,k}(t)$ to $r_{m,k}(t)$ as result of a communication operation with node m at time t, we count a traffic of $D_{n,m}(t)$ packets. Note that the idealized transmission semantics implicitly lead to upper bounds for sustained consistency, since all transmissions are successful and no information is lost. For the same reason, the transmission semantics lead to a lower bound for control traffic, since re-transmissions are not required. Furthermore, no traffic for the discovery of multihop routes is considered.

To the best of our knowledge, no workload model for IM systems has been published in the academic literature. Thus, we have to rely on intuitive assumptions for both the distributions of the number and the popularity of contacts as well as for the dwell time in a presence state. We present a parameterized workload model. Due to space limitations, we fix a single set of parameters for the remainder of this paper. However, we have analyzed other parameter settings and made similar observations, although with slight differences in absolute values.

To ensure that performance results for different connectivity levels are not biased by different node populations in the IM system, we adjust connectivity by changing the transmission range rather than the number of nodes in our experiments. Thus, we fix a number of mobile nodes $N = 100$. For the contact distribution, we assign a popularity of $p(n) = c/n$ to each node n with some constant c. That is, the popularity is given by a Zipf distribution. Node n inserts some other node k into $\mathbb{C}_n$ with probability $p(k)$. Note that contact relations in most IM systems are bi-directional. That is, n is on the contact list of k and vice versa if either n has chosen k or k has chosen n as a contact. The probability for a contact relation between n and k is $1 - (1-p(n))(1-p(k))$. Since it seems to be reasonable to have below 10 contacts on average in a community of 100 people, we set the constant c to 0.87. This yields a mean cardinality of 8.7 for the set $\mathbb{C}_n$ with a variance of 9.9. The dwell time of mobile node n in presence state i, i.e., $T_n(i+1) - T_n(i)$ is drawn from a lognormal distribution. Since it seems reasonable to have presence state dwell times between 5 and 10 minutes, we use the shape parameter $\sigma = 0.6$, and scale parameter $\mu = 6$. With this parameter setting, the distribution has a mean of 483s, i.e., 8.05 min, and a standard deviation of 14.4s. Note that for the computation of the sustained consistency, it is not required to define the exact presence state, but only the instances of time on which presence state changes occur. The nodes move on a plane of size 1000m × 1000m according the random waypoint (RWP) mobility model with a speed v from (0,2] m/s and a pause time from (0,120] s. To avoid initial transient behavior in simulation study, we employ perfect simulation by choosing initial node positions and node speed from the steady state spatial distribution of the RWP mobility model [11].

For all performance curves in the remainder of this section, we conducted finite time horizon simulations of length T = 7200s. For all performance measures, average values are calculated using 20 independent replicates. All figures include 99% confidence intervals. Since these confidence intervals are very tight, for some values their borders may not be visible in the figures.

Figure 1 shows that using relay nodes increases sustained consistency by up to 0.22 compared to direct communication. Here, Approach 4 clearly outperforms Approaches 2 and 3, which increase performance by up to 0.06 and 0.11, respectively. Looking at the traffic generated by the approaches in Figure 2, we find that Approach 2 and Approach 3 generate almost identical traffic. Nevertheless, since Approach 3 uses additional relay nodes, it disseminates presence state information faster than Approach 2. Approach 1 generates lower control traffic than Approach 2 and 3. This holds in particular for medium-connected networks, where presence information is copied to / from relay nodes over long routes with Approaches 2 and 3. Unrestricted copying as employed in Approach 4 generates most traffic, in particular in networks with medium transmission ranges. In such networks, synchronization is performed using long multi-hop routes, and, thus, requires many relay operations for each packet. In networks with small transmission ranges, long routes do not exist due to a lack of connectivity, while in networks with large transmission ranges they do not exist since the network can be traversed within a few hops. This causes the bell shape of the curve for Approach 4 in Figure 2.

Note that Figure 4 implies that with Approach 4 each node in average has to send up to 1.04 packets per second, while with Approach 3 the load is only up to 0.13 packets per second and node. Trading of the additional network load for the small gain in sustained consistency indicated by Figure 3, we will focus on Approach 3 in the remainder of this paper.

V. A SYSTEM FOR DISSEMINATING PRESENCE INFORMATION

In this section, we propose SPEED, the System for Presence information Exchange by Epidemic Dissemination in MANET that is motivated by Approach 3. With SPEED, each node floods (or pushes) its new presence state on a presence state change to all nodes reachable via multiple hops. To trigger transmission of presence information from relay nodes, every node polls its contact's presence states periodically by scoped flooding with a maximum number of hops. To bound the overhead of polling, an age is assigned to each information about the presence state of a remote node. A node polls the state of a contact only if the age has exceeded a certain threshold. A contact itself or any relay node having more up-to-date presence information responds to a poll message. As further difference to Section IV responses are not transmitted using unicast-communication, but transmitted by an approach denoted as directed flooding. Directed flooding constitutes a compromise between traffic efficiency and choosing additional relay nodes. In contrast to a conventional flooding operation, with directed flooding a message is only retransmitted by the nodes m that have received it from a node n with $D_{m,k}(t) < D_{n,k}(t)$ for the inquiring node k. All intermediate nodes overhear responses and extract new presence information. Information are cached so that each intermediate node becomes an additional relay node. In the remainder of this section we describe all mechanisms in more detail.

In addition to a list of the current state of its contacts denoted as *contact list*, each node n has a data structure $cache_n$ that stores information about the presence state of nodes $k \notin \mathbb{C}_n$. Note that $cache_n$ has at most N - $|\mathbb{C}_n|$ entries. We do not consider cache replacement, since required cache size is small. We denote presence information for node k stored at node n by $r_{n,k}$, regardless if it is stored in the contact list or in the cache. Note that cached entries do not affect sustained consistency. To determine how up-to-date a state information is, we assign an age denoted by $a_{n,k}$ to all $r_{n,k}$. For convenience $a_{n,n}$ is set to 0. Note that $a_{n,k}$ constitutes a time difference. Thus, $a_{n,k}$ can be easily computed using the local clock of a node and there is no need for synchronizing clocks.

SPEED uses messages of types PUSH, POLL, and RESPONSE. Besides payload, each message contains the additional fields *Type*, *Hops*, *TTL*, *Origin* and *Inquirer* that are described in Table 1. Upon receiving a message the *Hops* field is incremented by one and the *TTL* field is decremented by one. To approximate the distance matrix D defined in Section III, each node maintains an approximate distance to all nearby nodes in a data structure d. That is, the receiving node n stores the distance to the origin node m of the message, i.e., $d_{n,m}$:= *Message.Hops*. Since distances change due to node mobility, we assign each distance with a timeout of 10s. Experiments have shown that such timeout is sufficient, since each node that has to forward a RESPONSE message has also received the corresponding POLL message. If node n has no up-to-date distance information for node m, $d_{n,m}$ is set to ∞. Note that each node n only stores a single column of the matrices given by $r_{n,k}$, $a_{n,k}$ and $d_{n,k}$.

TABLE 1: DESCRIPTION OF MESSAGE FIELDS

Field	Description
Type	Message type (PUSH, POLL, RESPONSE)
Hops	Distance from receiving node to origin node in hops
TTL	Only if TTL > 0, message is send and forwarded
Origin	Node which generated that message
Inquirer	Used for responses to identify the inquiring node

When a node k changes its presence state at time $T_k(i)$, $1 \leq i \leq K_k(T)$, it pushes the new state to all nodes within a distance of at most TTL_{Push} using scoped flooding based on the field *Message.TTL*. The PUSH message comprises of a tuple (k,s), where $s = S_k(T_k(i))$ constitutes the new presence state of node k. Each node n that receives the message sets $r_{n,k} := s$ and $a_{n,k} := 0$.

To synchronize its contacts via relay nodes, each node polls the state of its contacts periodically in time intervals T_{Poll}. To reduce the number of messages, a single POLL message can contain queries for the states of multiple contacts. A node assures that the information about the presence state of its contacts is more up-to-date than some *information up-to-date timeout* TO_{iu}. A contact is only polled, if its state information in the contact list is older than TO_{iu}. A POLL message contains entries that comprise of a tuple (k,a), where a is the age information of contact k in the contact list. The age information enables nodes that receive a POLL message to decide if they have more up-to-date state information. Like a PUSH message, a POLL message is flooded with scope TTL_{Poll}. Note that an alternative design approach is to implement expanding ring search, that is, to successively flood the POLL message with increasing scope ttl, $1 \leq ttl \leq TTL_{Poll}$, until responses for all contacts in the POLL message are received. However, we found that this approach does not improve performance since the fraction of POLL messages that trigger responses for all contacts is low for small TTL_{Poll}. An algorithmic description of requesting contact's presence states is given by Algorithm 1.

If a node receives a POLL message and has more up-to-date state information either in its contact list or in $cache_n$ for one or more contacts in the POLL message, it generates a RESPONSE message. The payload of a RESPONSE message comprise of triples (k,a,s) of the contact k, the age of the information a and the contacts presence state s. If the local state information for a contact is sufficiently up-to-date, i.e., the age of that information is lower than the timeout TO_{iu}, the contact is removed from the POLL message, since it is not necessary that other nodes send responses for this contact. That applies in particular when the contact itself responds. A non-empty POLL message is forwarded to implement flooding, if $TTL > 0$. The message is not directly forwarded, but inserted into a forward queue together with a random value denoted as Δ. An algorithmic description of the actions upon receiving a polling message is given by Algorithm 2.

```
Algorithm 1: Node $n$ polls its contacts
Poll := ∅
For all nodes $k \in \mathbb{C}_n$
    If $a_{n,k} > TO_{iu}$
            Add $(k, a_{n,k})$ to Poll
If Poll ≠ ∅
    Flood Poll with $TTL_{Poll}$

Algorithm 2: Node $n$ receives a POLL message
Response := ∅
For each entry $(k, a) \in$ Poll
    If $k = n$
            Add $(k, 0, s_n)$ to Response
            Remove entry $(k, a)$ from Poll
    If $k \in \mathbb{C}_n$ or $k \in Cache_n$
            If $a_{n,k} < a$
                    Add entry $(k, a_{n,k}, r_{n,k})$ to Response
            If $a_{n,k} < TO_{iu}$
                    Remove entry $(k, a)$ from Poll
If Response ≠ ∅
    Response.TTL := $d_{n,Poll.Inquirer}$
    Δ := uniform random (0, 1]
    Response → ForwardQueue with Δ
If Poll ≠ ∅ and Poll.TTL > 0
    Δ:= uniform random (0, 1]
    Poll → ForwardQueue with Δ

Algorithm 3: Node $n$ receives a RESPONSE message
If Response.Inquirer = n
    For each entry $(k, a, s) \in$ Response
            If $a < a_{n,k}$
                    Set $r_{n,k} := s$ and $a_{n,k} := a$
Else
    Redundant = ∅
    For each entry $(k, a, s) \in$ Response
            If $k = n$
                    remove entry $(k, a, s)$ from Response
            If $k \in \mathbb{C}_n$ or $k \in Cache_n$
                    If $a \le a_{n,k}$
                            Add k to Redundant
                    If $a < a_{n,k}$
                            Set $r_{n,k} := s$ and $a_{n,k} := a$
                    Else
                            remove entry $(k, a, s)$ from Response
            Else
                    Add entry $(k, a, s)$ to $Cache_n$
    q := Response.InquiringNode
    r := Response.Origin
    If $d_{n,q} < d_{r,q}$ and Response.TTL > 0
            If ForwardQueue contains a response to this poll
                    Add Response to queued response
                    whereas older entries are replaced
            else
                    Δ := uniform random (0, 1]
                    Response → ForwardQueue with Δ
    Else
            For each response message in ForwardQueue
                    Remove all $(k,a,s)$ with $k \in Redundant$
```

Recall that RESPONSE messages are returned to the inquiring node using directed flooding. To implement directed flooding, a node r broadcasts its RESPONSE message, where *TTL* is set to the distance to the inquiring node. A node n relays the message if it is closer to inquiring node q than responder r, i.e., $d_{n,q} < d_{r,q}$. Note that node n can determine $d_{r,q}$ by summing up the number of hops and remaining TTL of the RESPONSE message. A node n within the transmission range of a responding or relaying node overhears the message and extracts new information about nodes k contained in the RESPONSE message. For each entry (k,a,s) in the RESPONSE message, node n searches for k in its contact list and in $cache_n$, respectively. If k is not found or the presence information s in the RESPONSE message is more up-to-date than the local information, i.e., $a < a_{n,k}$, the local state information $r_{n,k}$ is updated. Otherwise, the entry (k,a,s) is eliminated from RESPONSE message, since the node n itself has already responded with at least as up-to-date information with high probability. Thus, we avoid forwarding stale or redundant information. Furthermore, if it holds $d_{n,q} > d_{r,q}$, SPEED eliminates all entries, which are not more up-to-date than information in the received RESPONSE message from RESPONSE messages pending in the forward queue, since a node closer to the inquirer has already responded. An algorithmic description of the actions upon receiving a RESPONSE message is given by Algorithm 3.

As a disadvantage of flooding, redundant transmissions of PUSH and POLL messages may occur. That is, a node receives the same message from multiple sources. We found that more than 80% of all POLL messages are redundant with the approach described so far. To eliminate redundant transmissions we use a counter based flooding approach [16]. Such approach probabilistically reduces the number of forwarding nodes while keeping the number of covered nodes almost constant. Before forwarding a message, SPEED adds a random delay $\Delta \in (0,1]$, which is chosen according to a uniform distribution. The message is stored in the forward queue until the delay Δ has elapsed. If before the end of the delay the node n receives an identical message from *Tr* or more neighbors for a threshold *Tr*, it discards the message, because with high probability all neighbors of node n have already received that message. We found that an optimal value of the threshold *Tr* depends on the message type. Thus, we use a threshold Tr_{Push} for PUSH messages and Tr_{Poll} for POLL messages, respectively.

VI. A DETAILED PERFORMANCE STUDY OF SPEED

A. Simulation Methodology

Opposed to the high-level stochastic simulation studies performed for comparing different approaches for presence information dissemination in Section IV, we provide an evaluation of SPEED in a realistic setup in this section. For evaluation purpose, we use the network simulator ns-2 [7] instead of the high-level discrete event simulator. ns-2 considers communication delays as well as MAC and physical layer effects causing interference and packet collisions. We implemented SPEED as described in Section V as an ns-2 application. The application uses the UDP/IP protocol stack as well as the IEEE 802.11 MAC and physical layer running in ad-hoc mode. The payload for each entry in a PUSH, POLL or

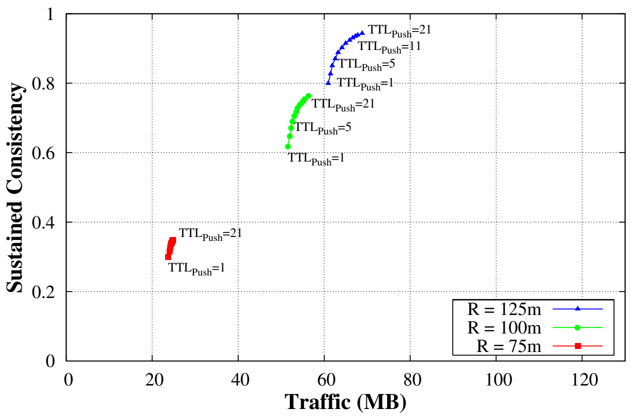

Figure 3. Sensitivity to scope of PUSH messages

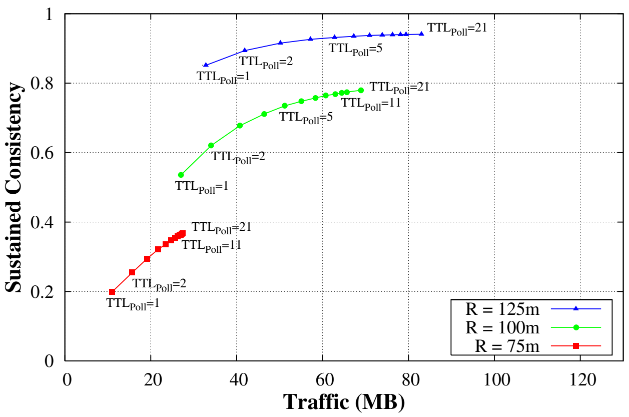

Figure 4. Sensitivity to scope of POLL messages

RESPONSE message has a size of 20 Bytes. Consistent with Section IV, we analyze scenarios with $N = 100$ mobile nodes in a plane of 1000 m × 1000 m and use different transmission ranges to account for different node densities. Contacts, state dwell times and node mobility are chosen according to the probabilistic models presented in Section IV. In fact, we use the same input data for identical scenarios an the same methodology for computing mean values and confidence intervals to assure that the results of this section are comparable to Section IV.

B. *Impact of System Parameters*

We conducted extensive simulations for exploring the entire design space of possible settings of the SPEED system parameters. Goal of the simulation was to gain insight into the characteristics of the system. We omit most of the results due to space limitations and show only the results for TTL_{Push} and TTL_{Pull}, respectively, in Figures 3 and 4. Default values for all parameters are given in Table 2.

Figure 3 plots sustained consistency vs. control traffic for $TTL_{Push} \in [1,21]$. Note that in this type of plots, a higher slope indicates that changing the parameter TTL_{Push} is more beneficial for the performance since it increases sustained consistency with a negligible increase of control traffic. We find that sustained consistency is highly sensitive to TTL_{Push}. However, the counter-based flooding approach for Push messages causes the traffic to increase gracefully with TTL_{Push}. That is, sustained consistency can be increased by 16% to 28% by increasing TTL_{Push} from 1 to 11. At the same time, traffic increases by at most 12%. Figure 4 plots sustained consistency vs. control traffic for different scopes of POLL messages $TTL_{Poll} \in [1,21]$. Recall that polling state information is a more frequent operation than pushing state information. Thus, traffic is more sensitive to TTL_{Poll} despite of the counter-based approach for flooding POLL messages. Nevertheless, increasing TTL_{Poll} from 1 to 6 yields a significant increase of sustained consistency for low and medium transmission range. Further increasing TTL_{Poll} does not significantly increase sustained consistency, but increases traffic by up to 18%.

TABLE 2: DEFAULT SETTING OF SYSTEM PARAMETERS

Parameter	Value
T_{Poll}	60 s
TTL_{Push}	10
TTL_{Poll}	*6*

Parameter	Value
Tr_{Push}	3
Tr_{Poll}	1
TO_{iu}	100 s

The results shown in Figures 3 and 4 indicate that with respect to control traffic the choice of TTL_{Poll} is much more crucial than the choice of TTL_{Push}. We found a similar sensitivity for the poll interval T_{Poll} and the information up-to-date timeout TO_{iu}. In particular, the choice of the three parameters TTL_{Poll}, T_{Poll}, and TO_{iu} is highly dependent on the transmission range R. Thus, inspired by [16] we use an adaptive approach for selecting T_{Poll}, TTL_{Poll}, and TO_{iu}. That is, each node keeps track of the number of nodes currently located in the transmission range. That can be easily accomplished by a node n by counting the nodes m with $d_{n,m} = 1$. Depending on the current number of nodes, T_{Poll}, TTL_{Poll}, and TO_{iu} are set according to Table 3. All other parameters are selected according to Table 2.

C. *Performance Comparison*

To show the efficiency of SPEED, we compare sustained consistency and control traffic to the results computed for Approach 3 and an optimized periodic flooding approach with period $T_{Push\_flooding}$, respectively. To provide a fair comparison, the flooding approach uses a counter-based flooding optimization with threshold $Tr_{Push} = 3$ and adaptive selection of $T_{Push\_flooding}$ with settings given by Table 3. For computing the control traffic generated by Approach 3, we assume that each message has a size of 92 bytes.

Figure 3 shows that SPEED performs equal or better compared to the flooding approach in terms of sustained

TABLE 3: ADAPTIVE PARAMETERS FOR SPEED AND THE FLOODING BASED APPOACH

# Neighbors	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	> 14
T_{Poll} **(s)**	20	30	40	50	55	55	65	65	75	75	75	75	80	100	105	110
TTL_{Poll}	7	6	5	5	5	5	4	4	4	3	3	3	2	2	1	1
TO_{iu}	30	40	50	60	75	90	100	150	150	150	200	200	200	250	250	250
$T_{Push\_flooding}$ **(s)**	10	10	10	20	20	25	30	50	50	75	75	100	100	150	150	200

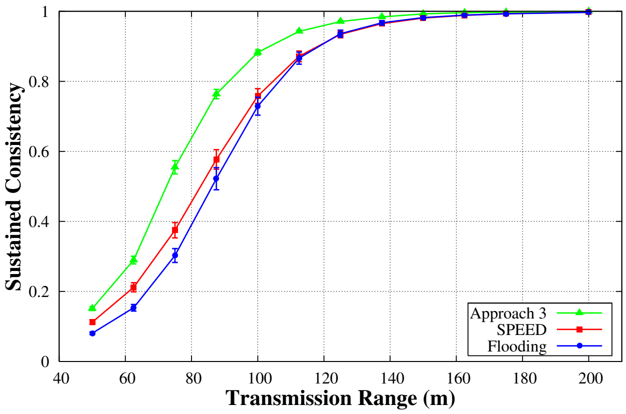

Figure 5. Sustained system consistency for different design alternatives

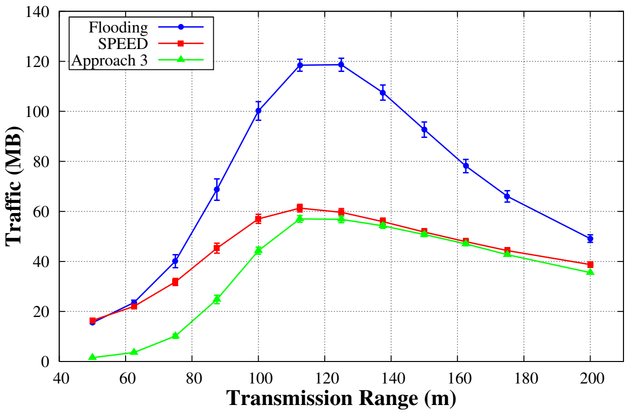

Figure 6. Required control traffic for different design alternatives

consistency. In particular in scenarios with transmission ranges $R \leq 115$ m, SPEED is superior to the flooding approach with a performance gain of up to 20%. For transmission ranges $R >$ 115 m, the flooding approach performs equal to SPEED with respect to sustained consistency, since increasing connectivity favors flooding. However, Figure 4 shows that in such scenarios SPEED saves up to 48% of the control traffic generated by the flooding approach. Note that similar to Approach 4 in Section IV the flooding approach generates most traffic in scenarios with medium transmission ranges. These scenarios exhibit large network partitions; however, most nodes are connected by routes with many hops. Thus, flooded messages must be retransmitted several times to reach every node in a network partition. In contrast, the combination of message relaying, polling, and directed flooding used by SPEED ensures that presence information is only transmitted between nearby nodes. Note that with SPEED, the average bandwidth used per node for presence state maintenance is about 87 Bytes per second and can be easily handled with current wireless communication hardware.

Comparing both the sustained consistency and the control traffic generated by SPEED to the results for Approach 3 shows that the configuration of SPEED presented in Tables 2 and 3 performs close to the optimum with respect to control traffic. However, there is much space for improvement with respect to sustained consistency. Nevertheless, the sensitivity studies indicated a significant trade-off between traffic and sustained consistency. Thus, depending on the requirements of the application, the parameters given by Tables 2 and 3 can be easily modified to achieve a higher sustained consistency at the expense of increased control traffic.

VII. Conclusions

In this paper, we provided an in-depth analysis of approaches for disseminating frequently changing presence information in highly partitioned mobile ad-hoc networks. Though routing techniques for delay tolerant networks are applicable for this problem, the fact that presence information is highly delay-sensitive requires a thorough revision of these techniques.

We proposed sustained consistency as a novel performance measure for the coherence of the presence information experienced within the system. As further performance measures of interest we consider the control traffic. We compared several approaches for presence information dissemination in terms of sustained consistency and control traffic. All bounds were derived by discrete-event simulation based on a high-level stochastic model of an instant messaging system. We showed that an approach that uses all contacts and all nodes reachable at the time of a presence state change as relay nodes constitutes the method of choice.

Building upon this observation, we presented SPEED, the System for Presence information Exchange by Epidemic Dissemination in a MANET. SPEED comprises an optimized mix of controlled flooding, caching, and disseminating information via relay nodes. We found that the choice of some system parameters is crucial and proposed adaptive selection of the poll interval, the scope for flooding poll messages, and the information timeout based on the number of nodes located in the transmission range. A comparison to an optimized approach based on periodical flooding of presence information reveals that using relay nodes increases sustained consistency for small transmission ranges, while it reduces control traffic for large transmission ranges. In fact, for transmission ranges up to 115 m SPEED achieves up to 20% higher sustained consistency than the flooding approach while for larger transmission ranges, is saves up to 48% control traffic. A comparison between SPEED and the results of the high-level simulation study showed that the presented configuration of SPEED achieves high traffic efficiency at the expense of reduced sustained consistency. However, sensitivity studies show that SPEED can be easily configured to achieve higher sustained consistency at the expense of increased control traffic.

Acknowledgement

The authors would like to thank Christian Lambert for his contributions to the simulation-based optimization of the SPEED system.

References

[1] M. Day, J. Rosenberg, and H. Sugano, A Model for Presence and Instant Messaging, IETF RFC 2778, 2000.

[2] K. Fall, A Delay-Tolerant Network Architecture for Challenged Internets, *Proc. ACM SIGCOMM*, Karlsruhe, Germany, 2003.

[3] R. Groenevelt, P. Nain, and G. Koole, The Message Delay in Mobile Ad Hoc Networks, *Performance Evaluation* **62**, 210-228, 2005.

[4] M. Grossglauser and D. Tse, Mobility Increases the Capacity of Ad-hoc Wireless Networks, *IEEE/ACM Trans. on Networking* **10**, 477-486, 2002.

[5] K. Harras, K. Almeroth, and E. Belding-Royer, Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding Schemes in Sparse Mobile Networks, *Proc. IFIP Networking Conf.*, Waterloo, Canada, 1180-1192, 2005.

[6] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. *Proc. ACM SIGCOMM*, 2004.

[7] K. Fall and K. Varadhan (editors), *The ns-2 manual*, Technical Report, The VINT Project, UC Berkeley, LBL, and Xerox PARC, 2006.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin, Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, *Proc. ACM MobiCom*, Boston, MA, 56-67, 2000.

[9] J. LeBrun, C. Chuah, and D. Ghosal, Knowledge Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks, *Proc. IEEE VTC-Spring*, 2005.

[10] Q. Li and D. Rus, Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks, *Proc. ACM MobiCom*, Boston, MA, 44-55, 2000.

[11] J.-Y. Le Boudec and M. Vojnovic, Perfect Simulation and Stationarity of a Class of Mobility Models, *Proc. IEEE INFOCOM*, Miami, FL, 2005.

[12] U. Mohan, K. Almeroth, E. Belding-Royer, Scalable Service Discovery in Mobile Ad Hoc Networks, *Proc. IFIP Networking Conf.*, Athens, Greece, 137-149, 2004.

[13] R. de Moraes, H. Sadjadpour, and J. Garcia-Luna-Aceves, Throughput-Delay Analysis of Mobile Ad-hoc Networks with a Multi-copy Relaying Strategy, *Proc. IEEE SECON*, Santa Clara, California, 4-7, 2004.

[14] D. Nain, N. Petigara, H. Balakrishnan, Integrated Routing and Storage for Messaging Applications in Mobile Ad Hoc Networks, *Mobile Networks and Applications* **9**, 595-604, 2004.

[15] P. Saint-Andre (Ed.), *XMPP – Core Protocol*, IETF RFC 3920, 2004.

[16] Y. Tseng, S. Ni, and E. Shih, Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network, *IEEE Transactions on Computers*, **52**, 545-557, 2003.

[17] A. Vahdat and D. Becker, *Epidemic Routing for Partially-Connected Ad hoc Networks*, Technical Report CS2000-06, University of California San Diego, 2000.

[18] J. Widmer and J.-Y. Le Boudec, Network Coding for Efficient Communication in Extreme Networks, *Proc. Workshop on Delay Tolerant Networking and Related Networks (WDTN-05)*, Philadelphia, PA, 2005.

[19] W. Zhao, M. Ammar, E. Zegura, A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks, *Proc. 5th ACM MobiHoc*, Tokyo Japan, 2004.

[20] W. Zhao, M. Ammar, and E Zegura, Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms, *Proc. Workshop on Delay Tolerant Networking and Related Networks (WDTN-05)*, Philadelphia, PA, 2005.