Performance Analysis of Epidemic with Immunity Routing for Disruption Tolerant Networks

Richard J. La[†], Padma Mundur[‡], Matt Seligman[⋆], and Ginnah Lee[†]

***Abstract*—We propose a new message routing mechanism based on the concept of *immunity* for Disruption Tolerant Networks (DTNs). The new mechanism allows more efficient utilization of limited buffer space at the nodes, by removing redundant copies of messages that have already been delivered. We first develop a new mathematical model for estimating the message delivery ratio (MDR) and average delivery delay (ADD) under a variant of epidemic routing with the immunity mechanism. We implement the proposed mechanism in ns2 simulator and demonstrate that our model provides accurate estimates of the MDR and ADD for varying traffic load and buffer size at the nodes. Secondly, we compare the performance of the epidemic routing with immunity against that of the original epidemic routing and spray-and-wait algorithms. We illustrate a trade-off between the number of copies forwarded to relay nodes and the stay times of messages at the nodes, both of which affect performance. Furthermore, while the spray-and-wait algorithm can achieve a higher MDR than our mechanism in some cases, we show that this comes at the price of a larger ADD.**

I. Introduction

Recently there have been growing interests in Disruption Tolerant Networks (DTNs), especially in military applications [9], [13], [14]. One of salient features of DTNs is that one-hop connectivity of the network between nodes is assumed to be sparse or intermittent. A consequence of this intermittent/sparse connectivity is that an end-to-end route between an information source and its intended destination is unlikely to be available when needed. For this reason traditional MANET routing protocols (e.g., AODV [17] or DSR [10]) that assume the availability of an end-to-end route are no longer suitable.

In addition to sparse connectivity, in general a pair of nodes in a network may never encounter each other. Therefore, even when infinite delay is allowed, some nodes may never be able to deliver messages directly to their destinations. Hence, in some cases nodes may not be able to count on a single (relay) node to deliver messages to intended destination(s), and multiple relay nodes may be required. For these reasons, some routing schemes (e.g., epidemic routing [22] and spray-and-wait routing [20]) allow multiple copies of messages in the network in order to increase the fraction of messages successfully delivered to their destinations, called message delivery ratio (MDR), and/or to reduce delivery delays. This is generally done at the expense of increased storage requirements at the nodes and higher resource needs necessary to forward multiple copies.

R. J. La and G. Lee are with the Department of Electrical & Computer Engineering at the University of Maryland, College Park. P. Mundur is with the UMIACS at the University of Maryland, College Park. M. Seligman is with the Laboratory for Telecommunications Sciences, College Park, MD.

This work was funded by the Laboratory for Telecommunications Sciences (LTS).

A. A short survey of related work

There are several existing routing algorithms for DTNs and studies on their analysis (e.g., [8], [14], [20], [21], [22], [23]). In this section, we limit our discussion to the studies most relevant to our study and provide a motivation for the study.

The achievable performance of a routing algorithm in DTNs depends on the time-varying network topology and the information available to the algorithm. On one hand, if the mobility of the nodes is deterministic and the contact times between the nodes are known in advance, a set of links can be scheduled for transmission at *different* times to offer end-to-end delivery of messages. This is one possible operational mode of original *delay* tolerant networks. On the other hand, if the mobility is stochastic, which is the scenario of interest to us, only time-varying one-hop local connectivity information may be available to the nodes for forwarding decisions.

We assume that nodes' mobility is stochastic. However, we focus on the scenario where, due to either inability to gather sufficient relevant statistical information pertaining to node mobility and network topology or a lack of computational power to process such information, the nodes are not capable of exploiting statistical properties of nodes' mobility. Instead, we are interested in investigating how the performance metrics, such as the MDR and average delivery delay (ADD), are shaped by how many copies of the messages are forwarded to other relay nodes (i.e., how many nodes receive a copy) and how long the nodes carry the copies before losing them to buffer overflow, which we call *stay time* of the messages at the nodes.

At one end of the spectrum, a simple approach to maximizing the number of nodes carrying a copy is to forward a copy to every node that comes in contact with another node with a copy. This is the basic idea behind epidemic routing [22], which mimics the way an infectious disease propagates throughout a population. Such an algorithm increases storage requirements at the nodes. Hence, when the buffer size is finite, it will lead to a higher message drop rate at the nodes, thereby reducing the stay times of the messages.

A variant of epidemic routing, called spray-and-wait [20], attempts to control the maximum number of copies in the network. The key idea behind the algorithm is that once a sufficient number of nodes carry a copy, the benefits from generating additional copies are marginal. Thus, by limiting the maximum number of copies, hence, curbing the message drop rate at the nodes, the spray-and-wait allows messages to remain in the buffer for a longer period, thereby increasing their chances of being delivered to their destinations before being dropped by buffer overflow. These observations suggest that there is a trade-off between the number of copies of messages produced and their stay times at the nodes.

There is another dimension to the problem of designing an efficient routing algorithm for DTNs. The copies of messages forwarded *after* the messages have already been delivered, while they consume resources (e.g., buffer space), do not improve the MDR or ADD. Thus, minimizing the proliferation of messages after their delivery will reduce resource consumption and, in doing so, increase the stay times of messages without affecting the number of messages forwarded *before* delivery. This is the basic observation exploited by our proposed *immunity* mechanism; the immunity mechanism provides a means for the nodes to propagate the information on the set of messages that have been delivered with the aim of curtailing wasteful proliferation of delivered messages.

We note that a similar immunity concept has been introduced earlier: Haas and Small [6] discuss the impact of deleting obsolete information in the context of an infostation model for sensor network applications. The identifier for a delivered or offloaded packet is called *antipacket*, and they propose several different methods (called IMMUNE, IMMUNE_TX and VACCINE), based on how antipackets are used. Another closely related study is the work by Zhang et al. [23] on epidemic routing and its variants, including one similar to epidemic routing with immunity studied in this paper.

Both of these studies examine delivery delay and buffer requirements based on either a Markov chain model [6] or ordinary differential equations (ODEs) for the fluid limits as the number of nodes increases [23]. When they analyze the buffer requirements, however, they assume that the buffer size is infinite and is not a performance bottleneck. Thus, they do not explicitly model the message drops at individual nodes caused by buffer overflow. As a result, their findings cannot be used to predict the performance when the buffer sizes are finite and are not large enough to avoid buffer overflow. In this paper we propose a new analytical model that takes into account finite buffers and provides a method to compute the MDR and ADD. This model is validated through extensive simulation using a realistic implementation of the proposed immunity routing protocol in ns2 simulator.

B. A summary of contributions

In this paper we propose and implement an *immunity* mechanism, which can be employed with existing routing schemes. The information on delivered messages is maintained at individual nodes in an immunity list with their identifiers and is exchanged on node encounters. The entries in the immunity lists, which we call immunity messages, are similar to the *antipackets* introduced by Haas and Small [6].

The objective of maintaining immunity lists is to minimize unnecessary exchange of the messages that have already been delivered and, at the same time, to remove their copies from the network to free up buffer space for new messages. The immunity message of a delivered message propagates throughout the network, with the destination as the source of the immunity information, in much the same way a data message spreads across the network.

We implement the proposed mechanism with the epidemic routing, which we refer to as ‘immunity algorithm’ throughout the paper, and compare its performance against the original epidemic routing and spray-and-wait routing. In addition, we develop a new mathematical model for estimating the MDR and ADD under the immunity algorithm with buffer constraints. As we will show, finite buffer sizes introduce several challenges in developing an appropriate mathematical model for performance evaluation and computing MDR and ADD.

We demonstrate that our mathematical model provides a good estimate of the MDR for different traffic load and buffer sizes. The estimated ADDs are accurate when the buffer sizes are large enough to yield reasonably large MDRs (higher than 80 percent). In addition, while the average numbers of copies forwarded to other relay nodes (per message) predicted by our model tend to be higher than the simulation numbers (differ by a constant of 3 to 4 for most scenarios), it accurately tracks the change in the average number with varying buffer size.

A comparison of epidemic and immunity routing algorithms reveals that indeed the immunity algorithm is effective at removing the superfluous copies after the delivery of messages and reducing the message drops due to buffer overflow. As a result, the immunity algorithm improves the MDR, albeit a slight increase in the ADD in some cases. In addition, while the spray-and-wait achieves a higher MDR over the immunity algorithm in some cases by limiting the number of copies forwarded to relay nodes, this higher MDR comes at the price of a significantly larger ADD.

The rest of the paper is organized as follows: We first describe the proposed immunity mechanism in Section II. The mathematical setup and an analytical model for the immunity algorithm are described in Sections III and IV, respectively. Sections V and VI explain how the analytical model can be used to estimate the MDR and ADD analytically. Simulation results are provided in Section VII to validate the accuracy of our model and to compare the performance of three routing schemes – epidemic, immunity and spray-and-wait.

II. Description of the immunity algorithm

In the basic epidemic routing algorithm, exchange of messages between nodes takes place as follows [22]: When two nodes meet, each node prepares a summary vector with a list of messages it is currently carrying. They exchange the summary vectors, and by comparing the two vectors, each node determines the messages it does not have. They then send a request for those messages to the other node so as to receive a copy of them.

In our proposed scheme, we modify this simple communication protocol in epidemic routing to accommodate exchange of immunity messages. The details of communication between two nodes on encounter are provided in Table I. In our proposed scheme, upon encounter each node sends (i) a message list (m-list), which is similar to the summary vector in the epidemic routing, and (ii) an immunity list (i-list). Both lists consist of message identifiers (IDs); the immunity list contains the IDs of those messages that have already been delivered to their destination. This immunity protocol is introduced in [15] with a preliminary simulation study.

The purpose of the i-list is to keep track of the list of already delivered messages in an attempt to reduce their proliferation after delivery and to free up buffer space at the nodes for future message exchanges. For instance, when a node, say i, encounters another node j with a message that is on its i-list, node i does not request the message even if a copy of the message does not reside in its buffer. Furthermore, upon receiving the i-list from node i, node j removes its copy of the message. Hence, this immunity information exchange prevents an unnecessary transmission of the message from node j to node i and also removes the unwanted copy at node j, freeing up scarce buffer space. Note that the epidemic routing would allow the unnecessary transmission of the message and, barring buffer overflow, the stay of the unneeded copy of the message at node j's buffer.

Using the two lists, the nodes compile and exchange the list of messages they want from the other node. In addition, they identify the messages to be removed from their buffers based on the i-list from the other node. After receiving the requested messages, they modify their m-list and i-list.

Fig. 1 provides an example that illustrates how a message moves from its source to the destination through pairwise encounters of nodes over time. After node D, which is the intended destination, receives a copy of the message, it generates an immunity entry for the message in its i-list, and the immunity information is forwarded to other node at future encounters.

This immunity message spreads throughout the network in much the same way a message propagates in the epidemic routing. In the last event in Fig. 1, unnecessary forwarding of the message to node E is avoided by its immunity for the message. Moreover, node B also removes its copy of the message after it receives the immunity for the message from node E.

Implementing acknowledgments (ACKs) for exchange of messages is likely to speed up the process of distributing immunity. This is because the sender that delivers a message to its destination would be able to modify its i-list at the same time as the receiving destination node, allowing both the sender and the destination node to carry the initial immunity in their i-list and to spread the immunity in future encounters with other nodes. In this paper, however, we do not implement ACKs and, as a result, only the receiving destination node modifies its i-list (see Table I). Consequently, the immunity initially lies only with the destination until its next encounter.

III. Mathematical setup

We are interested in estimating the MDR and ADD experienced by the messages that are eventually delivered under the proposed immunity algorithm (IA) with a finite buffer size at the nodes.[1] To this end, we first develop a simple analytical model based on a Markov chain [3]. Rather than attempting to model and keep track of all the messages in the network, we focus on a *single* message and model the evolution of the number of copies of the message present in the network (i.e., the number of nodes carrying a copy of the message). We compare the numbers predicted by our model against ns2 simulation results in Section VII.

Let $\mathcal{N} := \{1, 2, \ldots, N\}$ be the set of mobile nodes in the network. These N nodes move on a domain $\mathbb{D}$. The location of node i at time $t \in \mathbb{R}_+ := [0, \infty)$ is denoted by $X_i(t)$. The mobility process of node $i \in \mathcal{N}$ is given by $\mathbb{X}_i := \{X_i(t);\ t \in \mathbb{R}_+\}$. Throughout the paper we assume that the mobility processes of the nodes, $\mathbb{X}_i, i \in \mathcal{N}$, are mutually independent and stationary.

For every pair of distinct nodes i and j in $\mathcal{N}$, we introduce a $\{0,1\}$-valued *reachability* process $\{\zeta_{ij}(t),\ t \in \mathbb{R}_+\}$ with the interpretation that $\zeta_{ij}(t) = 1$ if node i can communicate directly *to* node j at time $t \geq 0$ and $\zeta_{ij}(t) = 0$ otherwise. When $\zeta_{ij}(t) = 1$, we say that the communication link from node i to node j is 'up'. Otherwise, the communication link is 'down'. We assume that the communication links are bidirectional, i.e., $\zeta_{ij}(t) = \zeta_{ji}(t)$. The process $\{\zeta_{ij}(t),\ t \in \mathbb{R}_+\}$ is simply an alternating on-off process, with successive up and down time durations given by the random variables (rvs) $\{U_{ij}(k),\ k \in \mathbb{N}\}$ and $\{D_{ij}(k),\ k \in \mathbb{N}\}$, respectively, where $\mathbb{N} := \{1, 2, \ldots\}$. Note that the rvs $\{D_{ij}(k); k \in \mathbb{N}\}$ denote the intermeeting times between nodes i and j.

In order to make progress we introduce the following assumption on the intermeeting times between nodes:

Assumption 1: The intermeeting times $\{D_{ij}(k);\ k = 2, 3, \ldots\}$ between two nodes $i, j \in \mathcal{N}$ are given by a sequence of independent and identically distributed (i.i.d.) exponential rvs with mean $(\nu_\star)^{-1}$.

It has been suggested (e.g., [4], [12]) that the distribution of intermeeting times between a pair of nodes can be approximated by an exponential distribution (i) when nodes move according to common mobility model (e.g., the Random Waypoint (RWP) mobility model and Random Direction mobility model) on a bounded domain or (ii) when the intermeeting times can be represented as a (delayed) geometric sum of i.i.d. rvs. The same assumption was introduced by Zhang et al. in [23] as well.

For every distinct pair $i, j \in \mathcal{N}$, define $\mathbb{M}_{ij} := \{M_{ij}^k; k \in \mathbb{Z}_+\}$, where $M_{ij}^0 = 0$, M_{ij}^k $(k \geq 1)$ denotes the time at which the k-th meeting between nodes i and j takes place after time 0^+, and $\mathbb{Z}_+ := \{0, 1, 2, \ldots\}$. From the meeting times $M_{ij}^k, k \in \mathbb{Z}_+$, we can define another sequence of rvs $\mathbb{I}_{ij} = \{I_{ij}^k; k \in \mathbb{N}\}$, where $I_{ij}^k = M_{ij}^k - M_{ij}^{k-1}$. When contact times $U_{i,j}(k), k \in \mathbb{N}$, (i.e., the amount time during which the communication link between them is up) are much shorter than intermeeting times $D_{ij}(k),\ k \in \mathbb{N}$, from Assumption 1(i), we can approximate $I_{ij}^k, k \geq 2$, as i.i.d. exponential rvs with the parameter $\nu \simeq \nu_\star$.

Assumption 2: The rvs $I_{ij}^k, k \geq 2$, are i.i.d. exponential rvs

[1] The proposed model can be modified to study the performance of the original epidemic routing. However, due to space constraint we do not discuss the model for the epidemic routing in this paper.

node i	node j
Messages: m-list: $M_i = \{WXYZ\}$	m-list: $M_j = \{XLMN\}$
Immunity: i-list: $I_i = \{M\}$	i-list: $I_j = \{Y\}$
Send and receive m-list and i-list; merge i-lists: $I_i = \{MY\}$	Send and receive m-list, i-list; merge i-lists: $I_j = \{MY\}$
Find messages to request from node j: Step1: $R_i^\star = M_j - I_i = \{XLN\}$ (remove immunity messages from M_j); Step2: r-list $R_i = R_i^\star - M_i = \{LN\}$ (remove common messages)	Find messages to request from node i: Step1: $R_j^\star = M_i - I_j = \{WXZ\}$ (remove immunity messages from M_i); Step2: $R_j = R_j^\star - M_j = \{WZ\}$ (remove common messages)
Request and receive new messages: $\{LN\}$; Send ACKs (optional)	Request and receive new messages: $\{WZ\}$; Send ACKs (optional)
Note the dest. of message L is node i; Add final destination messages to i-list: i-list: $I_i = \{MYL\}$	Note the dest. of message Z is node j; Add final destination messages to i-list: i-list: $I_j = \{MYZ\}$
Add newly received message N to m-list Remove immunity message Y from m-list New m-list: $M_i = \{WXZN\}$	Add newly received message W to m-list Remove immunity message M from m-list New m-list: $M_j = \{WXLN\}$
New i-list: $I_i = \{MYL\}$	New i-list: $I_j = \{MYZ\}$

TABLE I
COMMUNICATION PROTOCOL ON NODE ENCOUNTERS

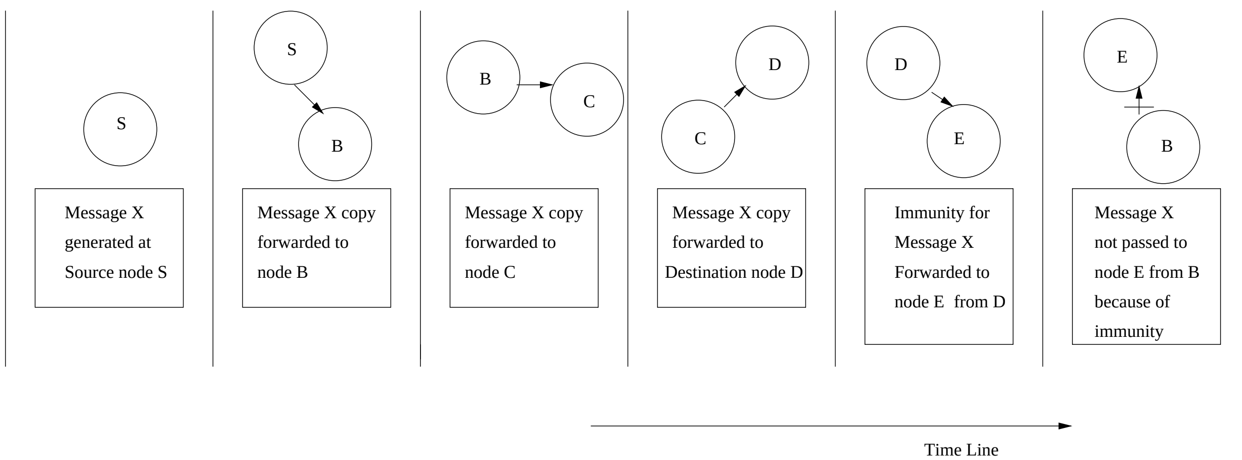


Fig. 1. Sequence of events for a single message through pair-wise node encounters.

with parameter ν. Furthermore, $\mathbb{I}_{ij}, i, j \in \mathcal{N}$, are mutually independent.

Assumption 2 implies that a node meets other nodes at rate $(N-1)\ \nu$, with the average amount of elapsed time between two consecutive meetings equal to $1/((N-1)\ \nu)$.

New messages arrive at node $i \in \mathcal{N}$ according to a Poisson process $\mathbb{B}_i$ with rate λ_i.[2] For our analysis, we assume that the new message arrival rates λ_i are the same, i.e., $\lambda_i = \lambda$ for all $i \in \mathcal{N}$ for some $\lambda > 0$, and the new message arrival processes $\mathbb{B}_i, i \in \mathcal{N}$, are mutually independent.

IV. MARKOV CHAIN-BASED MODEL

As mentioned earlier, we focus on a *single* message generated by some node and examine how the number of copies of the message evolves with time until either (i) a copy of the message is delivered to its destination or (ii) the message is purged from the network.[3] Without loss of generality, we assume that the message is created at time $t_0 = 0$.

Let $Y_C(t)$, $t \in \mathbb{R}_+$, denote the number of copies of the message (i.e., the number of nodes carrying a copy of the message) in the network at time t. In order to develop a tractable model we introduce following simplifying assumptions:

Assumption 3: (i) Exchange of messages between two nodes that meet takes place instantaneously, and all transmissions are successful. Further, either node is equally likely to request messages from the other node first. (ii) Suppose that two nodes i and j meet at time $t \in \mathbb{R}_+$, and a copy of message m requested by node j from node i causes a buffer overflow at node j. Then, every message present in node j's buffer just prior to the meeting is equally likely to be dropped. (iii) Message losses caused by buffer overflow at the nodes are independent. (iv) The buffer is full at every node *at steady state*.

Assumption 3(ii) is introduced for technical convenience so that we do not have to keep track of the position of each message in the buffer of every node. This causes some discrepancy between the analytical model and the ns2 simulation results (Section VII) when the buffer size is very small in relation

[2]The new messages here refer to the messages generated by node i, not including those received from other nodes.

[3]The number of copies after delivery can be modeled using a similar Markov chain with a larger state space. However, the computation becomes significantly more involved.

to the traffic load. Assumption 3(iii) does not hold in general, especially when the buffer sizes are small. Assumption 3(iv) is a reasonable assumption when the buffer size is a performance bottleneck, which is the scenario of interest to us.

Recall from Assumptions 1 through 3 that the intermeeting times between nodes are given by i.i.d. exponential rvs and new messages are generated by the nodes according to mutually independent Poisson processes. Thus, we can model $\mathbf{Y}_C := \{Y_C(t); t \in \mathbb{R}_+\}$ using a continuous-time Markov chain (MC) [3]. The state space of the MC is given by $\mathcal{S} := \{0, 1, \ldots, N-1, D\}$, where (i) $Y_C(t) = k, k = 1, 2, \ldots, N-1$, means that the message has not been delivered and there are k nodes carrying a copy of the message at time t, and (ii) $Y_C(t) = D$ indicates that a copy of the message has been delivered to its destination. Once the message is eliminated from the network, i.e., the MC $Y_C(t)$ reaches state 0, the MC stays there forever. Similarly, once a copy of the message is delivered to the destination, the MC remains at state D for good.

A. *Generator of continuous-time Markov chain*

Let us define q, $0 \leq q \leq 1$, to be the probability that a message present in the buffer of a node remains in the buffer after the node encounters another node and exchanges messages. Note from Assumption 3 that q is also the average fraction of messages in the buffer at the time of meeting which are not lost to buffer overflow during ensuing exchange of messages. Define $q_c := 1 - q$.

Denote the buffer size at the nodes by B. Assuming that each message is destined for a single destination, the off-diagonal elements of the generator of the continuous-time MC $\mathbf{Y}_C$ [3], denoted by $G = [g_{k,\ell};\ k, \ell \in \mathcal{S}]$, are given by

$$g_{k,\ell} = \begin{cases} k\ (N-k-1)\ \nu\ q \\ \quad \text{if} \ \ k = 1, 2, \ldots, N-2 \ \ \text{and} \ \ \ell = k+1 \\ k\ \nu\ \frac{1+q}{2} \\ \quad \text{if} \ \ k = 1, 2, \ldots, N-1 \ \ \text{and} \ \ \ell = D \\ \frac{k\ (k-1)}{2}\ \nu\ (q_c)^2 \\ \quad \text{if} \ \ k = 2, 3, \ldots, N-1 \ \ \text{and} \ \ \ell = k-2 \\ \frac{\lambda}{(B\ q)/2} + \frac{(N-1)\ \nu\ q_c}{2} \\ \quad \text{if} \ \ k = 1 \ \ \text{and} \ \ \ell = 0 \\ \frac{k\ \lambda}{(B\ q)/2} + \frac{k\ (N-k)\ \nu\ q_c}{2} + \ k\ (k-1)\ \nu\ q\ q_c \\ \quad \text{if} \ \ k = 2, 3, \ldots, N-1 \ \ \text{and} \ \ \ell = k-1 \\ 0 \quad \text{otherwise.} \end{cases} \tag{1}$$

The diagonal elements are given by $g_{k,k} = -\sum_{\ell \neq k} g_{k,\ell}$ for all $k \in \mathcal{S}$.

Let us explain the transition rates in (1): Recall from Assumption 2 that meetings between a pair of nodes occur at rate ν.

- $g_{k,k+1}$ – When there are k $(1 \leq k \leq N-2)$ nodes carrying a copy of the message, from the assumed mutual independence of $\mathbb{I}_{ij}$ (Assumption 2) meetings between the nodes with a copy of the message and other nodes without a copy, excluding the destination, take place at the rate of $k(N-k-1)\nu$. Since a node with a copy of the message, when it encounters a node without a copy, will successfully deliver a copy to the other node and not lose its own copy with probability q, the transition rate $g_{k,k+1}$ is given by $k(N-k-1)\nu\ q$.
- $g_{k,D}$ – Analogous to the previous case, when k nodes have a copy of the message, say m, they will meet the destination of message m at rate $k \cdot \nu$. When a node i carrying a copy of message m meets the destination, it will successfully deliver message m if (i) it delivers the messages requested by the destination, including message m, first or (ii) it first receives new messages it requested from the destination without dropping its copy of messages m. Since we assume that either node will request messages first from the other node with equal probability of 1/2, the probability that node i will successfully deliver message m to the destination upon encounter is given by $(1+q)/2$.
- $g_{k,k-2}$ – When there are k nodes with a copy, these nodes meet with each other at rate $(k\ (k-1)\ \nu)/2$, where $(k\ (k-1))/2$ is the number of different pairs of the nodes that can meet amongst the k nodes. When two of these nodes meet, the copy at each node will be lost with probability q_c, independently of each other (Assumption 3(iii)). Thus, the rate at which two copies are lost to buffer overflow equals $(k\ (k-1)\ \nu\ (q_c)^2)/2$.
- $g_{k,k-1}$ – There are two separate cases to consider:
(i) $k = 1$: If only a single node has a copy of the message, the message could be lost in two different ways. First, the copy may be lost due to buffer overflow caused by generation of new messages at the carrier of the copy between meetings with other nodes. Since we do not know the exact position of the message in the buffer, we assume that it is in the middle of the $B \cdot q$ messages that survived the last meeting with another node (hence, $B \cdot q/2$) and approximate the rate of the event as $\lambda/((B \cdot q)/2)$.
Second, when the node with the only copy meets another node, it may take on some of messages being carried by the other node which are absent in its buffer and, in the process, drop the only copy in order to free up enough buffer space for requested messages before it had an opportunity to deliver the message to the other node. This will happen with probability $q_c/2$ because the probability that the carrier will request messages from the other node first is 1/2 and, given that it does, the only copy of the message will be dropped from the buffer with probability q_c. This yields the rate $((N-1)\ \nu\ q_c)/2$ for the second case.
(ii) $k > 1$: When there are $k > 1$ copies in the network, the number of copies can decrease by one in three different ways. The first two are the same as in the case of $k = 1$. The third case arises when two nodes carrying a copy of the message meet (at rate $(k\ (k-1)\ \nu)/2$ as explained earlier) and one of the two copies is dropped, which happens with probability $2\ q \cdot q_c$ (i.e., one copy is lost due to buffer overflow while the other copy survives the exchange of message(s)).

B. Embedded discrete-time MC

Let $\{t_n; n = 1, 2, \ldots\}$ denote the sequence of times at which the continuous-time MC $\mathbf{Y}_C$, starting at state 1 at time $t = 0$, makes a transition to another state. Then, we can define a discrete-time MC $\mathbf{Y}_D := \{Y_D(n); n \in \mathbb{Z}_+\}$ with initial state $Y_D(0) = 1$ and $Y_D(n) = Y_C(t_n^+)$, embedded in the continuous-time MC with the same state space $\mathcal{S}$.

The one-step transition probabilities of the discrete-time MC $\mathbf{Y}_D$ can be found from the transition rates of the continuous-time MC in (1): The entries of the one-step transition matrix $\mathbf{P} = [P_{k,\ell}; k, \ell \in \mathcal{S}]$ of $\mathbf{Y}_D$ are equal to

$$P_{k,\ell} = \begin{cases} \frac{g_{k,\ell}}{\sum_{\ell^* \neq k} g_{k,\ell^*}} & \text{if } \ g_{k,\ell} > 0 \\ 1 & \text{if } \ k = \ell = 0 \ \text{ or } \ k = \ell = D \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The discrete-time MC $\mathbf{Y}_D$ is shown in Fig. 2.

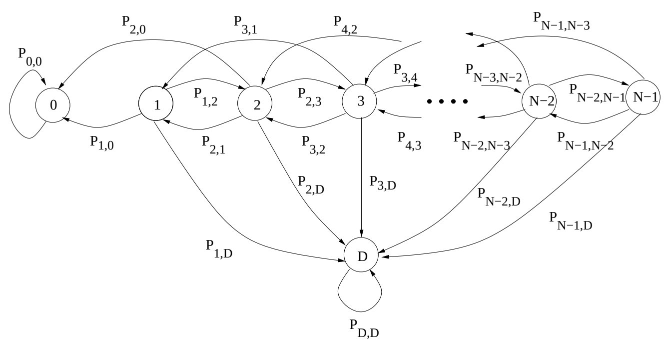


Fig. 2. Discrete-time Markov chain $\mathbf{Y}_D$.

V. Estimation of message delivery ratio and average delivery delay

In this section we describe, given (B, N, Λ, ν, q) where $\Lambda = N \cdot \lambda$ is the aggregate new message generation rate at all nodes, how we can estimate both the MDR and the ADD under the immunity algorithm, using the continuous-time and discrete-time MCs described in the previous section. We denote the MDR by p_{MDR} and the ADD by D_{avg}.

A. Estimation of message delivery ratio

First, note that states 0 and D of the discrete-time MC $\mathbf{Y}_D$ are the only two absorbing states, and the other states are transient. This tells us that, starting at state 1, the MC will reach one of these two absorbing states at some finite $n \in \mathbb{Z}_+$ with probability one. Hence, it is easy see that the probability that a copy of the message is successfully delivered to its destination is the probability that the MC $\mathbf{Y}_D$, starting at $Y_D(0) = 1$, eventually reaches state D (as opposed to being absorbed to state 0).

Let $f_k, k \in \mathcal{S}$, denote the probability that the MC $\mathbf{Y}_D$ will reach state D, starting at state $k \in \mathcal{S}$. It is obvious from the definition that $f_0 = 0$ and $f_D = 1$, and $p_{MDR} = f_1$. For each state $k \in \mathcal{S}$, by conditioning on the first transition out of state k, we obtain

$$f_k = \sum_{\ell \in \mathcal{S}} P_{k,\ell} \ f_\ell. \quad (3)$$

Eq. (3) yields the following set of linear equations.

$$\begin{aligned} f_1 &= P_{1,2} \ f_2 + P_{1,0} \ f_0 + P_{1,D} \ f_D = P_{1,2} \ f_2 + P_{1,D} \\ f_k &= P_{k,k+1} \ f_{k+1} + P_{k,k-1} \ f_{k-1} + P_{k,k-2} \ f_{k-2} \\ &\quad + P_{k,D} \ f_D \qquad (4) \\ &= P_{k,k+1} \ f_{k+1} + P_{k,k-1} \ f_{k-1} + P_{k,k-2} \ f_{k-2} + P_{k,D}, \\ &\qquad k = 2, 3, \ldots, N-2 \\ f_{N-1} &= P_{N-1,N-2} \ f_{N-2} + P_{N-1,N-3} \ f_{N-3} + P_{N-1,D} \ f_D \\ &= P_{N-1,N-2} \ f_{N-2} + P_{N-1,N-3} \ f_{N-3} + P_{N-1,D} \end{aligned}$$

Note that there are $N-1$ unknowns $\{f_1, \ldots, f_{N-1}\}$ and $N-1$ linearly independent equations. Hence, we can solve for the unknowns as follows.

Given a matrix $\mathbf{A}$, we denote the submatrix of $\mathbf{A}$ containing rows r1 through r2 and columns c1 through c2 by $\mathbf{A}_{r1:r2,c1:c2}$. When the submatrix contains a single row or a column, we simply write $\mathbf{A}_{r1,c1:c2}$ or $\mathbf{A}_{r1:r2,c1}$. We can rewrite the relationship in (4) in the following simpler matrix form:

$$\mathbf{f} = \mathbf{P}_{1:N-1,1:N-1} \ \mathbf{f} + \mathbf{P}_{1:N-1,D} \ ,$$

where $\mathbf{f} = (f_1, f_2, \ldots, f_{N-1})^T$, and $\mathbf{P}$ is the one-step transition matrix of the discrete-time MC in (2). Hence, we obtain

$$\mathbf{f} = (\mathbf{I}_{N-1,N-1} - \mathbf{P}_{1:N-1,1:N-1})^{-1} \mathbf{P}_{1:N-1,D} \ , \quad (5)$$

where $\mathbf{I}_{N-1,N-1}$ is an $(N-1) \times (N-1)$ identity matrix.

B. Estimation of average delivery delay

We define the end-to-end delivery delay of a message that is successfully delivered (i.e., a copy of the message reaches its destination) to be the amount of time it takes after the generation of the message for the destination to receive a copy of the message. Then, the ADD D_{avg} experienced by successfully delivered messages is equal to the expected amount of time it takes for the continuous-time MC $\mathbf{Y}_C$, starting at state 1, to reach state D, conditional on the event that it reaches D. This expected delay can be computed using the MCs in a similar way we computed the MDR.

First, since we are dealing only with the messages that are successfully delivered, the MC must reach the absorbing state D (instead of state 0). Hence, we need to modify the transition probabilities of the discrete-time MC as follows:
When the MC $Y_D(n)$ is at state 1, it jumps either to state 2 with probability $(N-2) \ q/((N-2) \ q + 1)$ or to state D with probability $1/((N-2) \ q + 1)$. Note that these are the conditional probabilities $P_{1,2}/(1 - P_{1,0})$ and $P_{1,D}/(1 - P_{1,0})$, respectively. Similarly, when the MC is at state 2, it is not allowed to jump to state 0 and we need to modify the transition probabilities out of state 2 accordingly.

Let us define an $N \times N$ matrix $\mathbf{P}^\star = [P^\star_{k,\ell}; k, \ell \in \mathcal{S}^\star]$, where $\mathcal{S}^\star := \mathcal{S} \setminus \{0\}$, and

$$P^\star_{k,\ell} = \begin{cases} P_{k,\ell}/(1 - P_{k,0}) & \text{if } \ k = 1, 2 \ \text{ and } \ \ell \in \mathcal{S}^\star \\ P_{k,\ell} & \text{otherwise.} \end{cases} \quad (6)$$

Define $d(k) = -(g_{k,k})^{-1}$, $k = 1, 2, \ldots, N-1$, to be the expected amount of time the continuous-time MC spends at state k after it enters the state till the next jump out of the state. Suppose that $ED(k), k = 1, 2, \ldots, N-1$, denotes the expected delivery delay till a copy of the message reaches the destination, starting with k copies of the message in the network, minus $d(k)$. It is clear that $ED(D) = 0$ by definition. Then, by conditioning on the first transition out of the state under consideration, we obtain the following relation: For every $k \in \mathcal{S}^\star$,

$$ED(k) = \sum_{\ell \in \mathcal{S}^\star} P^\star_{k,\ell} \ ED(\ell) + d(k). \quad (7)$$

Define $\mathbf{ED} = (ED(1), \ldots, ED(N-1))^T$ and $\mathbf{d} = (d(1), \ldots, d(N-1))^T$. Then, we can rewrite the relation in (7) in the following matrix form:

$$\mathbf{ED} = \mathbf{P}^\star_{1:N-1,1:N-1} \ \mathbf{ED} + \mathbf{d}$$

or, equivalently,

$$\mathbf{ED} = (\mathbf{I}_{N-1,N-1} - \mathbf{P}^\star_{1:N-1,1:N-1})^{-1} \mathbf{d} \ . \quad (8)$$

The ADD experienced by successfully delivered messages is then given by $D_{avg} = ED(1) + d(1)$.

VI. Performance estimation under epidemic routing with immunity

Given the parameters (B, N, Λ), if the meeting rate ν between a pair of distinct nodes and the probability q are known, the MDR p_{MDR} and the ADD D_{avg} can be computed using (5) and (8), respectively. However, much of difficulty in estimating these performance measures under the proposed immunity algorithm lies in calculation of q. In this section we explain how we can estimate ν and q in order to analytically compute p_{MDR} and D_{avg} for the IA.

A. Estimation of probability q

In this subsection we first assume that the meeting rate ν is known and describe how we estimate q. Approximation of the meeting rate ν is detailed in the following subsection. First, we show that, given a fixed value of q, there are two constraints (dependent on other fixed system parameters) which must be satisfied by the average message arrival rate at a node. Let χ_n be the message arrival rate at a node, including messages generated by the node and those received from other nodes. We denote χ_n that satisfies the first (resp. second) constraint by $\chi^1_n(q)$ (resp. $\chi^2_n(q)$). We then find $q^\star$ that satisfies both constraints, i.e., $\chi^1_n(q^\star) = \chi^2_n(q^\star)$, and use $q^\star$ to estimate p_{MDR} and D_{avg}.

(i) CONSTRAINT 1: Suppose that α is the probability that a node that receives a copy of a message, conditional on the event that the message is successfully delivered to the destination, will be immunized before it loses its copy to buffer overflow and μ_n is the buffer overflow rate at a node (i.e., time average of the number of messages lost to buffer overflow per unit time). Then, we have the following relation:

$$\mu_n = \chi_n \ (1 - p_{MDR} \cdot \alpha) \ , \quad (9)$$

where the right-hand side is the message arrival rate times the probability that a message will be dropped due to buffer overflow (i.e., one minus the probability that the message will be removed successfully through immunization before buffer overflow).

The average number of messages dropped by a node due to buffer overflow *per meeting* with other nodes, denoted by Σ, can be computed as follows: As mentioned in subsection IV-A, there are two types of events that cause buffer flow. First, when the buffer of a node is full, any message generated by the node causes buffer overflow. Secondly, when a node encounters another node and receives new messages currently absent in its buffer, it causes buffer overflow if there is not enough buffer space for the new messages.

Recall that by Assumption 3(iv) we assume that a buffer is always full at steady state. Since the overall buffer overflow rate of a node is μ_n and the rate at which messages are lost to the first type of buffer overflow is $\lambda = \Lambda/N$, the buffer overflow rate due to the second type equals $\mu_n - \lambda$. Obviously, the rate $\mu_n - \lambda$ is equal to the average number of messages dropped by a node per meeting, namely Σ, times the rate at which the node meets other nodes, $(N-1)\ \nu$. Therefore, we have

$$\Sigma \ (N-1) \ \nu = \mu_n - \lambda \ \text{ or } \ \Sigma = \frac{\mu_n - \lambda}{(N-1) \ \nu} \ . \quad (10)$$

From (10), the fraction of messages in a buffer lost during an exchange of messages following a meeting, q_c, is given by

$$q_c = \frac{\Sigma}{B} = \frac{\mu_n - \lambda}{(N-1) \ \nu \ B}. \quad (11)$$

Substituting (9) in (11) for μ_n and solving for p_{MDR} yields

$$p_{MDR} = \frac{1}{\alpha} - \frac{\lambda}{\chi_n \ \alpha} - \frac{q_c \ (N-1) \ \nu \ B}{\chi_n \ \alpha}. \quad (12)$$

As explained in the previous section, for a given value of q, we can compute p_{MDR} from (5). Thus, we are interested in finding χ_n and α that satisfy (12). To this end, we first rewrite α as a function of p_{MDR} and χ_n and then solve for χ_n.

Suppose that a message m is delivered to its destination at time t_0. The immunization delay of a node i for message m is defined to be the delay experienced until node i receives the immunity for message m after t_0. The expected immunization delay, denoted by ξ^{-1}, can be computed using a simple continuous-time MC, based on the meeting rates between nodes. This is explained in Appendix A.

Assume that node i has a copy of message m at time t_0. The *residual life* of message m at node i refers to the additional stay time after t_0 the message would spend at node i till it is removed by buffer overflow *if there were no immunity*. We assume that we can model both the immunization delay of node i and the residual life of message m at node i as independent exponential rvs with parameter ξ and B/μ_n, respectively.[4] Then, α is equal to the probability that node i will be immunized before message m is lost to buffer overflow. This is given by

$$\alpha = \frac{\xi}{\xi + \mu_n/B} = \frac{\xi}{\xi + \chi_n \ (1 - p_{MDR} \cdot \alpha)/B}, \quad (13)$$

[4]In general these rvs are not exponential. However, we make this assumption to simplify the computation of α.

where the second equality follows from (9). We can solve (13) for α and obtain

$$\alpha = \frac{\xi \ B + \chi_n - \sqrt{(\xi \ B + \chi_n)^2 - 4 \ \chi_n \ p_{MDR} \ \xi \ B}}{2 \ \chi_n \ p_{MDR}}. \quad (14)$$

Note that α depends only on χ_n and p_{MDR} (for fixed B and ξ). Therefore, given q (hence, $p_{MDR}(q)$), we can find a unique value of χ_n that satisfies (12) and (14). We denote this value by $\chi_n^1(q)$.

(ii) CONSTRAINT 2: Let $C^\star$ be the average number of copies generated of messages, including the original copy of the messages. Similarly, $C_D^\star$ and $C_{UD}^\star$ denote the average number of copies generated of successfully delivered messages and that of undelivered messages, respectively. Then, we have

$$C^\star = p_{MDR} \ C_D^\star + (1 - p_{MDR}) \ C_{UD}^\star \ . \quad (15)$$

We can write $C_D^\star$ as the sum $C_D^b + C_D^a$, where C_D^b (resp. C_D^a) is the average number of copies of successfully delivered messages generated before (resp. after) they are delivered to the destinations.

The aggregate message arrival rate at *all* nodes must satisfy $N \cdot \chi_n = \Lambda \cdot C^\star$. This, with (15), gives us

$$\begin{aligned} \chi_n &= \frac{\Lambda}{N} \left(p_{MDR} \ C_D^\star + (1 - p_{MDR}) \ C_{UD}^\star \right) \\ &= \frac{\Lambda}{N} \left(p_{MDR} \ (C_D^b + C_D^a) + (1 - p_{MDR}) \ C_{UD}^\star \right). \quad (16) \end{aligned}$$

Hence, given q, in order to calculate χ_n we need to compute C_D^b, C_D^a, and $C_{UD}^\star$.

1. Computation of C_D^b – We can compute C_D^b by following the same steps used to compute D_{avg} in subsection V-B: Let $C_k^*, k = 1, 2, \ldots, N-1$, denote the expected number of copies produced of a successfully delivered message *until* it is delivered to the destination, starting with k copies in the network, and $C_D^* = 0$. Then, for every $k = 1, 2, \ldots, N-1$, by conditioning on the first transition out of the state, we obtain the relation

$$C_k^* = \sum_{j=1}^{k-1} P_{k,j}^\star \ C_j^* + \sum_{j=k+1}^{N-1} P_{k,j}^\star \ (C_j^* + 1). \quad (17)$$

This relation states that the number of copies generated increases by one with each transition from state k ($k = 1, 2, \ldots, N-2$) to $k+1$.[5] In a matrix form, (17) becomes

$$\mathbf{C}^* = P_{1:N-1,1:N-1}^\star \ \mathbf{C}^* + \mathbf{p}^\dagger \ ,$$

where $\mathbf{C}^* = (C_1^*, \ldots, C_{N-1}^*)^T$, and $\mathbf{p}^\dagger = (P_{1,2}^\star, P_{2,3}^\star, \ldots, P_{N-2,N-1}^\star, 0)^T$. Therefore, $C_D^b = C_1^*$ can be computed from

$$\mathbf{C}^* = (I_{N-1,N-1} - P_{1:N-1,1:N-1}^\star)^{-1} \ \mathbf{p}^\dagger.$$

2. Computation of $C_{UD}^\star$ – In order to compute $C_{UD}^\star$, we first calculate the expected number of copies produced of a message (not necessarily successfully delivered) until either a copy of the message is delivered or the message is purged from the network without being delivered, which we denote by $C^{\star\star}$. It is clear that

$$C^{\star\star} = p_{MDR} \ C_D^b + (1 - p_{MDR}) \ C_{UD}^\star \ . \quad (18)$$

Let $C_k^\ddagger, k = 1, 2, \ldots, N-1$, denote the expected number of copies generated of a message till either a copy reaches the destination or all copies disappear from the network, starting with k copies in the network. Clearly, $C^{\star\star} = C_1^\ddagger$. Again, by conditioning on the first transition out of state k, we have a relation similar to (17): For every $k = 1, 2, \ldots, N-1$,

$$C_k^\ddagger = \sum_{j=1}^{k-1} P_{k,j} \ C_j^\ddagger + \sum_{j=k+1}^{N-1} P_{k,j} \ (C_j^\ddagger + 1) \quad (19)$$

Eq. (19) is equivalent to $\mathbf{C}^\ddagger = P_{1:N-1,1:N-1} \ \mathbf{C}^\ddagger + \mathbf{p}^\ddagger$, where $\mathbf{C}^\ddagger = (C_1^\ddagger, \ldots, C_{N-1}^\ddagger)^T$, and $\mathbf{p}^\ddagger = (P_{1,2}, P_{2,3}, \ldots, P_{N-2,N-1}, 0)^T$. Hence,

$$\mathbf{C}^\ddagger = (I_{N-1,N-1} - P_{1:N-1,1:N-1})^{-1} \ \mathbf{p}^\ddagger.$$

In order to compute $C_{UD}^\star$, we use the relationship (18). Note that C_D^b can be computed as explained above, and p_{MDR} can be obtained from (5) given q. Thus, we get

$$C_{UD}^\star = \frac{C^{\star\star} - p_{MDR} \ C_D^b}{1 - p_{MDR}} \ .$$

3. Computation of C_D^a – Suppose that a copy of message m is delivered to its destination at time t_0 and that node j does not have a copy of message m at time t_0. Without loss of generality, denote the set of nodes with a copy of the message at time t_0 by $\{1, 2, \ldots, K\} =: \mathcal{K}$. To simplify our analysis we assume that the immunization delays experienced by the nodes for message m can be modeled as i.i.d. exponential rvs with parameter ξ, which are independent of $\mathbb{I}_{ij}, i, j \in \mathcal{N}$.[6] Let $\mathcal{A}_K :=$ {node j does not receive a copy of message m from the K nodes in $\mathcal{K}$}, assuming that the copies at the nodes in $\mathcal{K}$ can be dropped only by immunity (but no buffer overflow).

Let $\Theta_i, i \in \mathcal{N}$, be the time at which node i receives the immunity for message m and $M_{ij}^\star, i \in \mathcal{K}$, the first time nodes i and j meet after t_0. Define $\mathcal{A}^i := \{\min(\Theta_i, \Theta_j) < M_{ij}^\star\}$. Note that $\mathcal{A}^i$ is the event that node j does not request message m from node i when they meet because either node i has dropped its copy after receiving immunity ($\Theta_i < M_{ij}^\star$) or node j has been immunized ($\Theta_j < M_{ij}^\star$) before $M_{ij}^\star$. Thus,

$$\mathbf{Pr}\left[\mathcal{A}_K\right] = \mathbf{Pr}\left[\bigcap_{i=1}^{K} \ \mathcal{A}^i\right] = \mathbf{Pr}\left[\bigcap_{i=1}^{K} \ \left\{\min(\Theta_i, \Theta_j) < M_{ij}^\star\right\}\right].$$

[5]When a node with a copy meets another node without a copy, it is possible for a new copy to be generated in our model even when the MC stays at the same state. We discount these copies in calculation of $C^\star, C_D^\star$, and $C_{UD}^\star$ as they are small compared to the total number of copies generated.

[6]This assumed independence between the immunization delays and $\mathbb{I}_{ij}, i, j \in \mathcal{N}$, does not hold in practice as the immunization delays are closely tied to the meeting times between nodes.

By conditioning on the immunization delay Θ_j of node j,

$$\begin{aligned}
&\mathbf{Pr}\left[\mathcal{A}_K\right] \\
&= \int_{\mathbb{R}_+} \mathbf{Pr}\left[\bigcap_{i=1}^{K} \{\min(\Theta_i, \Theta_j) < M^\star_{ij}\} \;\Big|\; \Theta_j = t \right] \\
&\qquad \times \xi \ \exp(-\xi\ t)\ dt \\
&= \int_{\mathbb{R}_+} \left(\prod_{i=1}^{K} \mathbf{Pr}\left[\min(\Theta_i, \Theta_j) < M^\star_{ij} \;\Big|\; \Theta_j = t \right] \right) \\
&\qquad \times \xi \ \exp(-\xi\ t)\ dt \\
&= \int_{\mathbb{R}_+} \left(\prod_{i=1}^{K} \mathbf{Pr}\left[\min(\Theta_i, t) < M^\star_{ij} \right] \right) \xi \ \exp(-\xi\ t)\ dt\ ,
\end{aligned} \tag{20}$$

where the second equality follows from the conditional independence of $\mathcal{A}^i, i \in \mathcal{K}$, given Θ_j. Since $\Theta_i, i \in \mathcal{K}$, are i.i.d. by assumption, $\mathbf{Pr}\left[\min(\Theta_i, t) < M^\star_{ij}\right], i \in \mathcal{K}$, are the same and we only need to compute $\mathbf{Pr}\left[\min(\Theta_1, t) < M^\star_{1j}\right]$ in order to determine $\mathbf{Pr}\left[\mathcal{A}_K\right]$ using (20).

For every $t \in (0, \infty)$, let $\mathcal{E}(t) = \{M^\star_{1j} > t\}$ and $\mathcal{E}^c(t) = \{M^\star_{1j} \le t\}$. Then, by the law of total probability,

$$\begin{aligned}
&\mathbf{Pr}\left[\min(\Theta_1, t) < M^\star_{1j}\right] \\
&= \mathbf{Pr}\left[\min(\Theta_1, t) < M^\star_{1j} \;\Big|\; \mathcal{E}(t)\right] \cdot \mathbf{Pr}\left[\mathcal{E}(t)\right] \\
&\quad + \mathbf{Pr}\left[\min(\Theta_1, t) < M^\star_{1j} \;\Big|\; \mathcal{E}^c(t)\right] \cdot \mathbf{Pr}\left[\mathcal{E}^c(t)\right].
\end{aligned}$$

Clearly, $\mathbf{Pr}\left[\min(\Theta_1, t) < M^\star_{1j} \;\Big|\; \mathcal{E}(t)\right] = 1$ and $\mathbf{Pr}\left[\mathcal{E}(t)\right] = \exp(-\nu t)$. Note that

$$\begin{aligned}
&\mathbf{Pr}\left[\min(\Theta_1, t) < M^\star_{1j} \;\Big|\; \mathcal{E}^c(t)\right] \cdot \mathbf{Pr}\left[\mathcal{E}^c(t)\right] \\
&= \mathbf{Pr}\left[\left\{\min(\Theta_1, t) < M^\star_{1j}\right\} \bigcap \mathcal{E}^c(t)\right].
\end{aligned}$$

By conditioning on $M^\star_{1j}$,

$$\begin{aligned}
&\mathbf{Pr}\left[\left\{\min(\Theta_1, t) < M^\star_{1j}\right\} \bigcap \mathcal{E}^c(t)\right] \\
&= \int_0^t \mathbf{Pr}\left[\min(\Theta_1, t) < M^\star_{1j} \;\Big|\; M^\star_{1j} = \tau\right]\ \nu\ \exp(-\nu\tau)\ d\tau \\
&= \int_0^t \mathbf{Pr}\left[\Theta_1 < \tau\right]\ \nu\ \exp(-\nu\tau)\ d\tau \\
&= \int_0^t \left(1 - \exp(-\xi\tau)\right)\ \nu\ \exp(-\nu\tau)\ d\tau \\
&= 1 - \exp(-\nu t) - \frac{\nu}{\xi + \nu}\left(1 - \exp(-(\xi + \nu)t)\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbf{Pr}\left[\min(\Theta_1, t) < M^\star_{1j}\right] \\
&= \exp(-\nu t) + 1 - \exp(-\nu t) - \frac{\nu}{\xi + \nu}\left(1 - \exp(-(\xi + \nu)t)\right) \\
&= 1 - \frac{\nu}{\xi + \nu}\left(1 - \exp(-(\xi + \nu)t)\right).
\end{aligned} \tag{21}$$

The probability $\mathbf{Pr}\left[\mathcal{A}_K\right]$ can now be obtained by substituting (21) in (20) and carrying out the integration.

We approximate the probability that a node without a copy of message m at the time of its delivery will not receive a copy by $\mathbf{Pr}\left[\mathcal{A}_K\right]$ with $K = C^b_D$ and

$$C^a_D \simeq (N - 1 - C^b_D) \cdot \left(1 - \mathbf{Pr}\left[\mathcal{A}_{C^b_D}\right]\right)\ .$$

In this approximation we ignore two factors whose contributions to C^a_D tend to cancel each other out. First, other nodes that receive a copy after t_0 can also forward a copy to node j. At the same time, although on the average C^b_D copies are generated before delivery, some of these copies may have been lost to buffer overflow before the delivery takes place. The first tends to increase to the probability that node j will receive a copy, while the latter decreases the probability.

Once we obtain $C^\star_D = C^b_D + C^a_D$ and $C^\star_{UD}$, we can compute χ_n using (16) as a function of q (through $p_{MDR}(q)$), which we denote by $\chi^2_n(q)$. Since the correct value of q must satisfy both (12) and (16), we can numerically find $q^\star \in [0, 1]$ that satisfies $\chi^1_n(q^\star) = \chi^2_n(q^\star)$.

B. Estimation of average intermeeting times and meeting rate ν between two nodes

In the previous subsection we assumed that the meeting rate between two nodes, namely ν, was known. In practice, this quantity may be estimated by individual nodes, for example, by maintaining a record of meetings with other nodes. In this subsection, we explain how it can be approximated for our analysis when the one-hop connectivity between nodes is determined by the distances between them [16]: Recall from Section III that, for every pair of distinct nodes i and j, $\{\zeta_{ij}(t),\ t \in \mathbb{R}_+\}$ denotes the reachability process between the nodes and $\{U_{ij}(k); k \in \mathbb{N}\}$ and $\{V_{ij}(k); k \in \mathbb{N}\}$ are the sequence of contact times and intermeeting times, respectively, between them. When nodes' mobility is stationary, from elementary renewal theory [19],

$$\mathbf{Pr}\left[\zeta_{ij}(0) = 1\right] = \frac{\mathbf{E}\left[U_{ij}(2)\right]}{\mathbf{E}\left[U_{ij}(2)\right] + \mathbf{E}\left[V_{ij}(2)\right]} \tag{22}$$

Suppose that the spatial distribution $\mathcal{G}$ of the nodes is known. For example, the spatial distribution of the nodes under the RWP mobility model has been studied extensively in the literature (e.g., [2], [7]). If two nodes can communicate directly if and only if their distance is not larger than their transmission range γ, the probability $\mathbf{Pr}\left[\zeta_{ij}(t) = 1\right]$ is equal to

$$\mathbf{Pr}\left[\zeta_{ij}(0) = 1\right] = \int_{\mathbb{D}} \Big(\int_{\mathbb{D} \cap D_\gamma(\mathbf{x})} d\mathcal{G}(\mathbf{y}) \Big)\ d\mathcal{G}(\mathbf{x})\ , \tag{23}$$

where $\mathbb{D}$ is the mobility domain, and $D_\gamma(\mathbf{x})$ is the disk centered at $\mathbf{x}$ with radius γ.

It is clear from (22) and (23) that the meeting rate ν, which is given by $\nu = (\mathbf{E}\left[U_{ij}(2)\right] + \mathbf{E}\left[V_{ij}(2)\right])^{-1}$, can be computed if we can find either $\mathbf{E}\left[U_{ij}(2)\right]$ or $\mathbf{E}\left[V_{ij}(2)\right]$. For instance, Han et al. [5] investigated the distribution and expected value $\mathbf{E}\left[U_{ij}(2)\right]$ of contact times under the RWP mobility model and illustrated how they can be calculated. The expected value $\mathbf{E}\left[V_{ij}(2)\right]$ is then obtained from (22) as

$$\mathbf{E}\left[V_{ij}(2)\right] = \mathbf{E}\left[U_{ij}(2)\right] \times \frac{1 + \mathbf{Pr}\left[\zeta_{ij}(0) = 1\right]}{\mathbf{Pr}\left[\zeta_{ij}(0) = 1\right]}.$$

VII. Simulation

We implemented the original epidemic algorithm (EA) along with our proposed variant – immunity algorithm (IA) – using ns2 simulator [1]. For subsection VII-B we also implemented a simple version of spray-and-wait (S&W) where the source and a known maximum number of distinct relay nodes carry a copy of the message to its destination. The goal of the first part (subsection VII-A) is to examine the accuracy of the analytical model described in Section IV. In the second part, we compare the performance of the three algorithms and examine several trade-offs.

There are 50 nodes in the network ($N = 50$). The nodes move in a rectangular region of size 1500 m $\times$ 300 m, following the RWP mobility model implemented using the newer version of ns2 mobility model. Speeds of the nodes are given by i.i.d. rvs uniformly distributed over the interval [2 20] m/s. The same mobility file is used for different protocols for fair comparison. The average intermeeting times between a pair of nodes, $(\nu_\star)^{-1}$, under these parameters is approximately 700 seconds. Messages arrive according to a Poisson process with rate Λ (messages per second). For each arrival, we select its source and destination randomly from all possible pairs of the nodes with equal probability.

The simulation is run for 5000 seconds. The first 2000 seconds are a warm-up period, and we collect the measurements using the messages generated only between 2000 and 4000 seconds. These messages are followed through till the end of simulation. However, the messages that are not delivered till 5000 seconds are considered undelivered. We checked the average speed of the nodes over different sliding windows during the period of [2000 4000] seconds in order to ensure that the network is close to the steady-state. Other simulation parameters are given in Table II. We selected these values to produce meaningful performance in terms of the MDR and the ADD, which can be compared for different schemes.

Parameters	Range
Number of nodes (N)	50
Coverage area	1500 m $\times$ 300 m
Radio range	50 m
Node speed	Uniformly distr. over [2 20] m/s
Message length	1 Kbytes
Simulation time	5000 seconds
Traffic load (Λ)	(0.0625, 0.125, 0.1875)
Buffer capacity (B)	2-10 messages

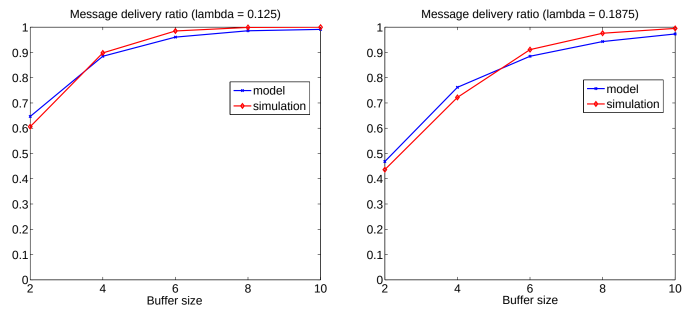
TABLE II
SIMULATION PARAMETERS.

We consider the MDR and the ADD to measure the accuracy of the analytical model and to analyze the performance of the proposed IA. The MDR from the simulation is given by the fraction of the messages that are generated during [2000 4000] seconds and are successfully delivered to the destinations before 5000 seconds. The ADD is computed as the total delay of all messages that are generated over the same interval and are delivered to their respective destinations divided by the number of delivered messages. Redundant deliveries for the same message are discarded in computing the MDR and the ADD. In addition to these two metrics, for comparison of the three algorithms we also analyze the number of transmissions and drops as well as the numbers of existing copies per message after delivery.

A. Comparison of analytical model and simulation results for immunity routing

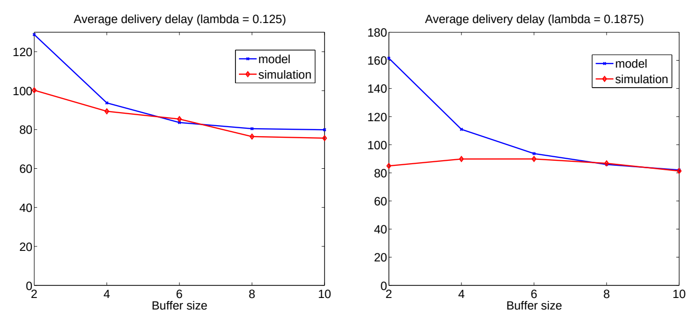
In this subsection we compare the ns2 simulation results against the numbers predicted by our analytical model outlined in Section IV for two different traffic loads – $\Lambda = 0.125$ and $\Lambda = 0.1875$.

1. Message delivery ratio: Fig. 3 plots the MDRs from both the ns2 simulation and our analytical model as the buffer size at the nodes (B) varies from 2 to 10. It is clear from the plots that the MDRs predicted by the model are very close to the simulation numbers for all values of the buffer size for both traffic loads.



(a) (b)

Fig. 3. Message delivery ratio. (a) $\Lambda = 0.125$, (b) $\Lambda = 0.1875$.

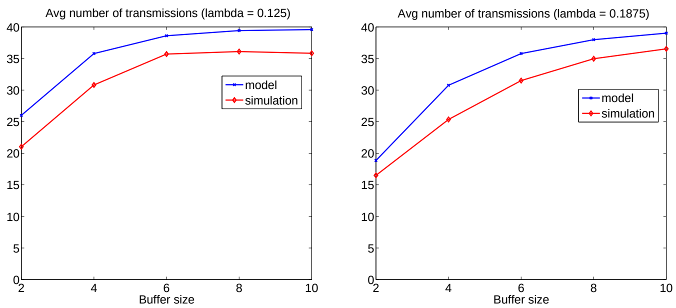


(a) (b)

Fig. 4. Average delivery delay. (a) $\Lambda = 0.125$, (b) $\Lambda = 0.1875$.

2. Average delivery delay: The ADDs of successfully delivered messages are plotted in Fig. 4. These plots suggest that, when the buffer size is large enough (in relation to the traffic load) so that the MDR is reasonably high (above 0.75), the ADD predicted by our analytical model is close to the simulation results. However, when the buffer size is too small for the given traffic load, buffer overflow at the nodes is frequent and messages are either delivered quickly before being purged from the network or all copies are lost to buffer overflow before any of them reaches the destination. As our analytical model does not explicitly keep track of the position of individual messages in the buffers for analytical tractability (Assumption 3), this causes our model to overestimate the ADD as shown in the plots, providing conservative estimates.

3. Average number of transmissions per message: We also plot the average numbers of transmissions per message (for



(a) (b)

Fig. 5. Average number of transmissions per message. (a) $\Lambda = 0.125$, (b) $\Lambda = 0.1875$.

both delivered and undelivered messages) in Fig. 5. Recall that, from Assumption 3(i), that the number of transmissions is equal to the number of copies forwarded to other (relay) nodes. Although our model tends to slightly overestimate the average number of transmissions compared to the ns2 simulation, it does a good job of tracking the trend in the average number of transmissions per message as the buffer size changes for both traffic loads.

B. Performance comparison – epidemic, immunity and spray and wait

In this subsection we present a comparative analysis of the performance of three algorithms – EA, IA and S&W. Under the S&W algorithm [20], the maximum number of copies of a message that can be generated, which we denote by $M_{\max}$, is a control parameter. For the simulation we set $M_{\max} = 4$ and 8. These numbers are chosen to illustrate the relative delay performance of the S&W while the MDRs are comparable to that of the IA.

1. Effects of storage capacity: We fix the traffic load at $\Lambda = 0.125$. We measure the MDR, ADD, average number of drops per message and average number of transmissions per message while varying the buffer size from 2 to 10 messages. In addition, we compute the average number of copies per message captured at different points of time after delivery of the messages.

Figs. 6 and 7 show the effects of varying buffer capacity at the nodes on the MDR and the ADD, respectively. We can make the following observations from the figures:

(i) The MDR of the IA is close to those of S&W(4) and S&W(8) when the buffer size is large enough to maintain the MDR above 0.8 (i.e., $B \geq 4$). The ADD of the IA is, however, considerably smaller than those of the S&W. In particular, when $B = 4$ the ADD of the S&W(4) is nearly twice that of the IA, and the ADD of S&W(8) is approximately 60 percent higher than that of the IA for $B \geq 8$.

(ii) When the buffer size is small, the EA attains the smallest ADD. However, the achieved MDR is considerably lower than that of the IA or the S&W. In order to achieve the same level of MDR, the EA requires almost twice the buffer capacity; the MDR of the S&W and IA at $B = 4$ is close to that of the EA with $B = 8$. In fact, even when $B = 10$, the EA fails

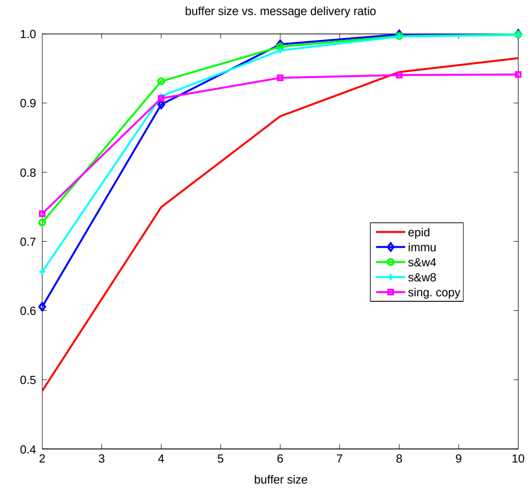


Fig. 6. Message delivery ratio ($\Lambda = 0.125$).

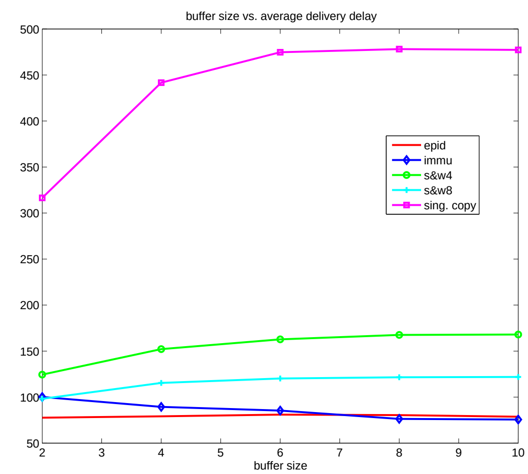
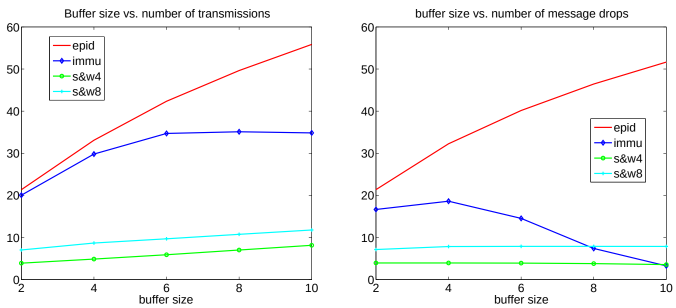


Fig. 7. Average delivery delay ($\Lambda = 0.125$).

to achieve the MDR close to one realized by the IA and the S&W. In addition, when $B \geq 8$, the IA outperforms the EA in delay performance as well.

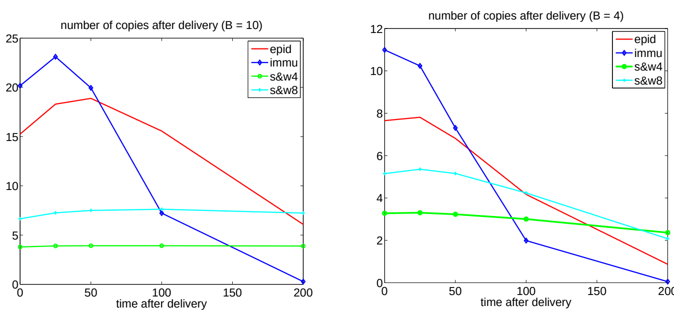
(iii) The single-copy algorithm does not benefit from increasing buffer size as much as the other algorithms do, and even when the buffer size is $B = 10$, it fails to reach an MDR near one. Moreover, while it achieves the highest MDR when $B = 2$, its ADD is much larger than that of other algorithms.

Fig. 8 plots the average number of transmissions (for forwarding a copy) per message and the average number of copies that are dropped by buffer overflow per message as the buffer size varies. This figure suggests that the network behavior under the EA and the IA is quite different; while both numbers climb steadily under the EA as the buffer size increases, the number of transmissions saturates under the IA and the number of drops in fact decreases after $B = 4$. The difference between the number of transmissions and drops reveals that, when the buffer size is sufficiently large, most of the messages are removed by immunity messages under the IA (as opposed to buffer overflow). This difference between



(a) (b)

Fig. 8. Average number of drops and transmissions per message. (a) transmissions, (b) drops.



(a) (b)

Fig. 9. Average number of copies after delivery. (a) $B = 10$, (b) $B = 4$.

the two algorithms is caused by indiscriminate exchanges of messages between a pair of nodes by the EA on every encounter. This in turn leads to inefficient use of limited buffer space at the nodes and lower MDRs.

Buffer size	2	4	6	8	10
EA	37.9	50.4	60.6	70.2	78.7
IA	38.8	52.0	63.0	74.1	81.2
S&W(4)	198.6	393.0	582.8	768.8	947.5
S&W(8)	111.0	201.2	300.1	400.0	499.3

TABLE III
AVERAGE STAY TIME OF MESSAGES AT A NODE.

More efficient utilization of buffer space by the IA can also be seen from Fig. 9. The figure plots the average number of copies in the network of delivered messages captured at different points in time after the delivery.[7] At the time of delivery, the number of copies of delivered messages is higher under the IA than the EA as shown in the figure. However, the number decreases much more rapidly under the IA once the immunity message starts to propagate throughout the network.

Figs. 8 and 9 confirm that indeed the immunity mechanism can quickly remove the redundant copies of delivered messages in the network, freeing up buffer space at the nodes. As a result, nodes retain the copies of undelivered messages for longer periods and, because of larger stay times, more nodes carry a copy of undelivered messages until they are delivered, increasing the chance of delivery. This is corroborated by Table III that lists the average stay times of messages. Given that many messages are removed by immunity messages before experiencing overflow, it is clear from the table that the average stay time of the messages lost to buffer overflow under the IA would be much larger than that under the EA.

Buffer Size	Total Drops	Before Delivery	After Delivery
2	3.9	1.6	2.3
4	3.9	0.5	3.4
6	3.9	0.2	3.7
8	3.8	0.05	3.75
10	3.56	0.02	3.54

TABLE IV
AVERAGE DROPS PER MESSAGE ($M_{\max} = 4$).

We can make the following observations about the S&W algorithm. First, note that the number of transmissions can exceed $M_{\max}$ due to redundant deliveries to the destinations. Secondly, the average stay time under S&W(4) is about twice the average stay time under S&W(8). This is consistent with the expectation that the message arrival rates at the nodes, including those received from other nodes, would be roughly proportional to $M_{\max}$. Thirdly, although the numbers of drops per message are close to $M_{\max}$, it is clear from Table IV that, when $B \geq 4$, most of the drops occur after the messages have already been delivered. This can be inferred from Table III and Fig. 7 as well; when $B \geq 4$, the average stay time of the messages is much larger than the ADD. Thus, the fraction of copies dropped before the delivery should be small. Figs. 6 and 7 and Table III also illustrate the aforementioned trade-off between the number of copies generated and the average stay times at the nodes and its effects on the MDR and ADD.

2. Effects of traffic load: Fig. 10 shows the effect of varying traffic load on the MDR and ADD with $B = 4$. As one would expect, there is a significant decline in the MDR as the traffic load increases under all three algorithms. Moreover, the MDR falls off slightly faster under the EA and IA compared to the S&W. However, as shown in Fig. 10(b), this comes at the price of a larger ADD under the S&W. In addition, one should keep in mind that this simulation is run with a very small buffer size of $B = 4$. When the buffer size increases, the MDR of the IA becomes comparable to that of the S&W (Fig. 6) while the difference in the ADD performance becomes more pronounced (Fig. 7).

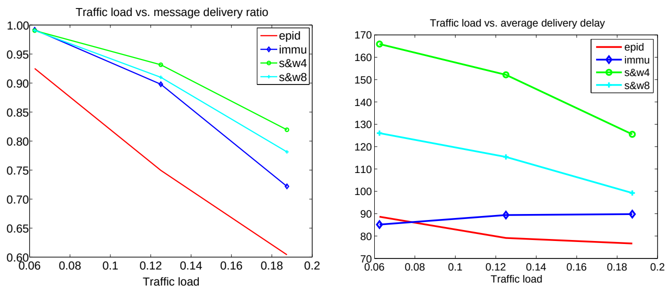
VIII. Conclusion

We proposed and implemented a new routing scheme for DTNs based on the idea of immunity, which we call immunity algorithm. We first developed a new mathematical model

[7] Each point in Fig. 9 at time $\Delta \in \{0, 25, 50, 100, 200\}$ seconds is obtained as a sample average computed as follows: Let $\mathcal{M}$ be the set of messages that are generated between 2000 and 4000 seconds and are delivered successfully before 4800 seconds. For each $m \in \mathcal{M}$, we denote the time at which message m is delivered by T_m and the number of nodes carrying a copy of message m at time $T_m + \Delta$ by $n_m(\Delta)$. For each $\Delta \in \{0, 25, 50, 100, 200\}$, the sample average plotted in Fig. 9 is given by

$$n^{avg}(\Delta) := \frac{\sum_{m \in \mathcal{M}} n_m(\Delta)}{|\mathcal{M}|},$$

where $|\mathcal{M}|$ is the cardinality of $\mathcal{M}$.



(a) (b)

Fig. 10. (a) Message delivery ratio for varying load, (b) average delivery delay for varying load ($B = 4$).

that can be used to estimate the MDR and ADD under the immunity algorithm. We showed that our model can accurately predict the MDR for varying traffic load and buffer size. When the buffer size is reasonably large relative to the traffic load, it can also predict the ADD accurately.

Secondly, we compared the performance of three different routing schemes – epidemic, immunity and spray-and-wait algorithms. We demonstrated that the immunity algorithm significantly outperforms the epidemic routing in most scenarios by utilizing limited buffer space more efficiently. Furthermore, we illustrated that the spray-and-wait algorithm can sometimes provide higher MDRs, especially when the buffer size is small relative to the traffic load, by limiting the maximum number of nodes that can carry a copy of messages. However, this gain in MDR often comes at the expense of larger ADDs. Thus, this represents a trade-off between the MDR and ADD and suggests that when evaluating the performance of a routing algorithm, both performance metrics should be considered.

Appendix

Suppose that a message is delivered at time t_0, at which time the destination receives immunity. Without loss of generality assume that the destination is node 1 and $t_0 = 0$. Let $N_I(t), t \geq 0$, be the number of nodes with immunity at time t, with $N_I(0) = 1$. Then, by Assumption 2, $\{N_I(t);\ t \geq 0\}$ is a continuous-time MC with state space $\tilde{\mathcal{S}} = \{1, 2, \ldots, N\}$ and the generator $\tilde{G} = [\tilde{g}_{ij}; i, j \in \tilde{\mathcal{S}}]$, where

$$\tilde{g}_{i,j} = \begin{cases} i\ (N-i)\ \nu \\ \qquad \text{if } i = 1, 2, \ldots, N-1 \text{ and } j = i+1 \\ -i\ (N-i)\ \nu \\ \qquad \text{if } i = 1, 2, \ldots, N-1 \text{ and } j = i \\ 0 \qquad \text{otherwise.} \end{cases}$$

This tells us that when $N_I(t) = k$, the number of nodes with immunity $N_I(t)$ jumps to state $k+1$ at rate $k(N-k)\nu$, which is the rate at which one of the k nodes with immunity meets one of the remaining $N-k$ nodes without immunity. Let $T_k, k = 2, 3, \ldots, N$, denote the time at which $N_I(t)$ jumps from $k-1$ to k, and $T_1 = 0$. Then, we have

$$\begin{aligned} \mathbf{E}\left[T_k\right] &= \mathbf{E}\left[T_{k-1}\right] + \frac{1}{(k-1)(N-k+1)\nu} \\ &= \sum_{l=2}^{k} \frac{1}{(\ell-1)(N-\ell+1)\nu}\ , \qquad k = 2, 3, \ldots, N. \end{aligned}$$

We denote by Γ_i, $i \in \mathcal{N}$, the time at which node i receives immunity with $\Gamma_1 = 0$. It is plain that $\{T_k, k = 2, \ldots, N\}$ are the order statistics of $\{\Gamma_i, i = 2, \ldots, N\}$ [19]. Therefore,

$$\mathbf{E}\left[\sum_{i=2}^{N}\ \Gamma_i\right] = \mathbf{E}\left[\sum_{k=2}^{N}\ T_k\right] = \sum_{k=2}^{N}\left(\sum_{l=1}^{k-1}\frac{1}{l\ (N-\ell)\ \nu}\right),$$

and the average immunization delay is given by

$$\xi^{-1} = \frac{\mathbf{E}\left[\sum_{i=2}^{N}\ \Gamma_i\right]}{N-1}.$$

References

[1] http://www.isi.edu/nsnam/ns/

[2] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 2(3), pp. 257-269, 2003.

[3] G. Grimmett, and D. Stirzaker, *Probability and Random Processes*, third ed., Oxford University Press, 2001.

[4] R. Groenevelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," *Performance Evaluation*, vol. 62(1-4), pp. 210-228, Oct. 2005.

[5] Y. Han, R. J. La, A. M. Makowski, and S. Lee, "Distribution of path durations in mobile ad-hoc networks – Palm's Theorem to the rescue," *Elsevier Computer Networks Journal (COMNET)*, vol. 50(12), pp. 1887-1900, Aug. 2006.

[6] Z. Haas and T. Small, "A new networking model for biological application of ad hoc sensor networks," *IEEE/ACM Transactions on Networking*, vol. 14(1), pp. 27-40, Feb. 2006.

[7] E. Hyytiä, P. Lassila, and J. Virtamo, "Spatial node distribution of the random waypoint mobility model with applications," *IEEE Trans. on Mobile Computing*, vol. 5(6), pp. 680-694, Jun. 2006.

[8] Y.-K. Ip, W.-C. Lau and O.-C. Yue, "Performance modeling of epidemic routing with heterogeneous node types," *Proc. of IEEE ICC*, 2008.

[9] S. Jain, K. Fall and R. Patra, "Routing in a delay tolerant networks," *Proc. of ACM SIGCOMM*, 2004.

[10] D. B. Johnson, "Routing in ad hoc networks of mobile hosts," *Proc. of the Workshop on Mobile Computing Systems and Applications*, 1994.

[11] V. Kalashnikov, *Geometric Sums: Bounds for Rare Events with Applications*, Kluwer Academic Publishers, 1997.

[12] R. J. La, "Distributional convergence of inter-meeting times in generalized hybrid random walk model," submitted to IEEE Trans. on Mobile Computing, in Dec. 2007, revised n Jan. 2009.

[13] Z. Li, L. Sun and E. C. Ifeachor, "Adaptive multi-copy routing for intermittently connected mobile ad hoc networks," *Proc. of IEEE Globecom*, 2006.

[14] A. Lindgren, A. Doria and O. Schelen, "Probabilistic routing for intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7(3), pp. 19-20, July 2003.

[15] P. Mundur, M. Seligman, and G. Lee, "Epidemic routing with immunity in Delay Tolerant Networks," *Proc. of the Military Communications Conference*, 2008.

[16] M. D. Penrose, *Random Geometric Graphs*, Oxford Studies in Probability, Oxford University Press, 2003.

[17] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," *Proc. of the Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[18] A. Rényi, "A characterization of Poisson processes," *Magyar Tud. Akad. Mat. Kutato Int. Kozl.*, vol. 1, pp. 519-527, 1956 (English translation in *Selected Papers by Alfréd Rényi*, ed. by Pál Turán, vol. 1, pp. 622-628, 1976).

[19] S. M. Ross *Stochastic Processes*, 2nd ed., Wiley, 1996.

[20] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," *Proc. of ACM SIGCOMM Workshop on Delay Tolerant Networking*, 2005.

[21] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility," *Proc. of the Fifth Annual International Conference on Pervasive Computing and Communications Workshops*, 2007.

[22] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke University Technical Report CS-200006, April 2000.

[23] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, “Performance modeling of epidemic routing,” *Computer Networks: The International Journal of Computer and Telecommunication Networking*, vol. 51(10), pp. 2867-2891, Jul. 2007.

PLACE PHOTO HERE

Richard J. La received his B.S.E.E. from the University of Maryland, College Park in 1994 and M.S. and Ph.D. degrees in Electrical Engineering from the University of California, Berkeley in 1997 and 2000, respectively. From 2000 to 2001 he was with the Mathematics of Communication Networks group at Motorola Inc,. Since 2001 he has been on the faculty of the Department of Electrical & Computer Engineering and the Institute for Systems Research at the University of Maryland, where he is currently an Associate Professor.

PLACE PHOTO HERE

Padma Mundur received her Ph.D. degree in Information Technology from George Mason University, Fairfax, VA in 2000. Her Masters degree is in Systems Engineering from the University of Virginia, Charlottesville, VA. She was a faculty member in the Computer Science Department at the University of Maryland, Baltimore County from 2000 to 2006. She joined the Institute for Advanced Computer Studies at University of Maryland, College Park (UMIACS) in 2006 as a Research Faculty member. She is on the editorial board of IEEE Communications Surveys and Tutorials Journal; she has been on several conference program committees and served as a reviewer for NSF panels and many IEEE/ACM journals.

PLACE PHOTO HERE

Matt Seligman is a networking researcher at the Laboratory for Telecommunications Sciences (LTS) and is a Ph.D. candidate in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. He received his Masters degree in Electrical Engineering from Stevens Institute of Technology and his Bachelors degree in Computer Engineering and Electrical Engineering from Pennsylvania State University. He previously held positions at Coree Networks and Lucent Technologies in New Jersey. His research interests include DTNs, network simulation, and embedded systems. He is a member of ACM and IEEE.

PLACE PHOTO HERE

Ginnah Lee is currently doing her Ph.D in the Electrical and Computer Engineering Department at University of Maryland, College Park, MD. And, she has been working as a graduate research assistant at UMIACS under Dr. Mundur since Jan 2008. Ginnah got her B.S. in Computer Science and Engineering from Seoul National University, Seoul, Korea in Feb 2006. Ginnah’s research interests are mainly in network field including network modeling, protocol design, distributed systems and Delay Tolerant Networks. She is also interested in Human Computer Interaction and parallel computing.