ROUTING IN MOBILE DTNS: PERFORMANCE MODELING, NETWORK CODING BENEFIT, AND MOBILITY TRACE MODELING

A Dissertation Presented

by

XIAOLAN ZHANG

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2007

Department of Computer Science

ROUTING IN MOBILE DTNS: PERFORMANCE MODELING, NETWORK CODING BENEFIT, AND MOBILITY TRACE MODELING

A Dissertation Presented

by

XIAOLAN ZHANG

Approved as to style and content by:

Jim Kurose, Co-chair

Don Towsley, Co-chair

Weibo Gong, Member

Brian Neil Levine, Member

Andrew Barto, Department Chair
Department of Computer Science

To Chen Yu, my parents, and my sister.
In memory of Tao Zhang.

ACKNOWLEDGMENTS

It's my great fortune to have pursued my Ph.D. studies under the guidance of my advisors, Professor Jim Kurose and Professor Don Towsley. They introduced me to the subject area of this thesis, and guided me in every step with their profound knowledge, insights and wisdom. Their high standards for research and genuine interests for intellectual challenges have stimulated me and will remain a source of inspiration to me. I express my deep gratitude to them.

I would like to thank Professor Brian Levine for his advice, encouragement and support. I also thank Professor Weibo Gong for his candid advice and suggestions on my thesis work. My thanks also go to other faculty members that I worked with, Professor Prashant Shenoy, Professor Jack Wileden and Professor Beverley Woolf, for helpful discussion and advice during my graduate studies.

My special thanks go to Giovanni Neglia for being such a wonderful people to collaborate with. His commitment to doing good work, and his ways of approaching problems are a great example to me. It has also been a wonderful experience to collaborate with Michael Bradshaw on the AMPS project. I thank him for many wonderful suggestions and helpful discussions. During my graduate studies, I have had many interesting and inspiring discussions with current and former students and postdocs in the network research group, including Bing Wang, Chun Zhang, Wei Wei, Honggang Zhang, Yu Gu, Weifeng Chen, Junning Liu, Ping Ji, Zihui Ge, Yang Guo, Sudarshan Vasudevan and Bruno Ribeiro and others.

For my analysis studies of the DieselNet mobility traces, I have benefited greatly from discussions with John Burgess, who explained to me the setup details of the

testbed. I also thanks Adam Sherson, Yuri Pyuro, Jeff Jason and many others for providing information and data access which made the study possible.

I thank Laurie Connors and Betty Hardy for their great support and help in various aspects of my graduate studies. I also thank Tyler Trafford for his unfailing help with setting up machines and testbeds and solving all kinds of software and hardware problems. I am also grateful to staff members with Computer Science Computing Facility (CSCF) in the department for rescuing my data from the crashed disk, and their timely help with system and software problems.

Finally, my deep gratitude goes to my family. I thank my husband Yongsheng Yu for his support for my PhD studies and the tremendous help in keeping the family running. I thank my dear son, Chen Yu, for the love and joy that he brings to my life. I cannot imagine how I can make it without the love, support and understanding of my father, Shunxiang Zhang, and my mother, Yingzhu Lin. I dedicate this thesis to them.

ABSTRACT

ROUTING IN MOBILE DTNS: PERFORMANCE MODELING, NETWORK CODING BENEFIT, AND MOBILITY TRACE MODELING

SEPTEMBER 2007

XIAOLAN ZHANG

B.S., PEKING UNIVERSITY, BEIJING, CHINA

M.S., UNIVERSITY OF MASSACHUSETTS AT AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Jim Kurose and Professor Don Towsley

We study three related problems on unicast routing in Disruption-Tolerant Networks (DTNs), i.e., resource-challenged networks where contemporaneous end-to-end connectivity cannot be assumed. We particularly focus on mobility-induced DTNs with opportunistic contacts.

First, we propose a unified framework based on Ordinary Differential Equations (ODEs) to study the performance of a class of epidemic style routing schemes. Derived as the limit of Markov process model under proper scaling, the ODE models capture the propagation and recovery process of data packets under different schemes. We derive a rich set of closed-form results using the ODE models, and quantitatively characterize the performance trade-off achieved by different schemes. We also show that compared to the Markovian model, the ODE models have the additional advantages of analytic tractability and scalability in numerical solution.

Next, we investigate the benefit of applying Random Linear Coding (RLC), a special type of network coding, to epidemic style routing in resource-constrained DTNs. We explore different ways to apply network coding, and study both the case where there is a single block of packets propagating through the network, and the case where blocks of packets arrive continuously to multiple unicast flows. Our results show that due to its increased randomness, the RLC-based scheme achieves the minimal block delivery delay with high probability and improves the block delivery delay versus number of transmission trade-off. The relative benefit of network coding is even more significant when the node buffer is limited.

Last, we study mobility traces taken from UMass DieselNet, an operational bus-based DTN. We analyze the bus-to-bus contact traces in order to develop a generative model that can be used to generate synthetic traces with similar DTN routing performance as the original trace. Focusing on inter-contact times, we show that an aggregate model for the inter-contact time is too coarse a model to accurately capture DTN routing performance. We then construct a route-level inter-contact time model based on the trace, which captures interesting mobility structure within the trace and is shown to capture epidemic routing performance more accurately than the aggregate model.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

The focus of this thesis is Disruption Tolerant Networks (DTNs). The general field of DTNs, as defined by the Delay Tolerant Networking Research Group (DTNRG) [1] "is concerned with *how to address the architectural and protocol design principles arising from the need to provide interoperable communications with and among extreme and performance-challenged environments where continuous end-to-end connectivity cannot be assumed*". Application domains of DTNs include mobile wireless sensor networks [61, 100] for wildlife tracking, underwater sensor networks [87, 92], disaster relief team networks, networks for remote areas or rural areas in developing countries [2, 31, 6], vehicular networks [18, 53] and Pocket-Switched Networks [50]. These networks are subject to intermittent connectivity and disconnection of nodes due to limitations of power, node mobility, sparse node density, and equipment failures.

In this thesis, we focus on DTNs where node mobility is the main factor leading to the disruption of contacts, in other words, the communication links between nodes going up and down as nodes move in and out of range of each other. For some networks, such as Inter-Planetary Networks [19], the node-to-node contacts are deterministic because the mobile nodes move along fixed trajectories according to deterministic schedules (e.g., the Low-Earth Orbiting Satellite (LEO) or planets). For most applications, however, mobile node movement is either random, such as human beings carrying PDAs or animals attached with sensors, or semi-random, such as buses whose movement exhibits random effects due to traffic and road conditions. As a result, the contacts between nodes are unpredictable and opportunistic, a.k.a.,

opportunistic networks. Our emphasis in this thesis is on the latter case, whose opportunistic nature renders it the more challenging case; for the former case, [55] has proposed various routing algorithms based on different assumptions about available knowledge.

Many challenges have to be addressed for DTNs (the interested user is referred to [50] for an overview), and several previous works have studied DTNs from the point of view of power management [62], security [17], and capacity building through deploying static nodes [119] or specialized mobile nodes [118, 77]. This thesis will focus on the routing aspect which, as the key challenge for DTNs, has attracted the most interest from the research community. In the remainder of this chapter, we first give an overview about unicast routing in DTNs, reviewing previously proposed schemes from four different perspectives. We then discuss research efforts that have addressed the performance evaluation of DTN routing schemes and the importance of studying real DTN mobility traces. Last, we highlight the contribitions made in this thesis and present an outline of the thesis.

1.1 Unicast Routing Schemes for DTNs

In the past several years, numerous routing schemes [110, 57] have been proposed for DTN routing. Unlike traditional routing protocols which assume the existence of an end-to-end path between sender and receiver throughout the lifetime of their communications, DTN routing has adopted a so-called "store-carry-forward" paradigm. Under this paradigm, each node in the network *stores* a packet that has been forwarded to it by other node, *carries* the packet while it moves around, and *forwards* it to other relay nodes or the destination node when they come within transmission range. Such a paradigm, also called "mobility-assisted routing", is similar in spirit to the 2-hop routing scheme proposed by Grossglauser and Tse [40]. Proposed for dense mobile ad hoc networks, the 2-hop routing scheme takes advantage of node mobility

and attains a constant per-session throughput as the network density grows (in sharp constrast to the diminishing throughput under fixed networks as shown by Gupta and Kumar [41]).

Generally, previously proposed DTN routing schemes differ in the following aspects: *(i)* single-copy or multi-copy schemes, *(ii)* stateful or stateless schemes, *(iii)* non-coding or coding-based schemes, *(iv)* consideration of resouce constraints. We elaborate on these four aspects of DTN routing schemes next with some examples. Interested readers are also referred to [60] which provides a nice review of the DTN routing schemes.

First, routing schemes can be classified as single-copy or multi-copy schemes. Under a single-copy scheme, at any point of time, there is a single copy of the packet in the network; each packet is *forwarded (not copied)* along a single path. Under multi-copy schemes, there can be multiple simultaneous copies of a packet in the network; a packet is *copied (i.e., duplicated)* to other nodes, allowing simultaneous use of multiple paths to the destination. The single-copy schemes [103, 57] generally incur less transmission overhead and place less demand on buffer space; the challenge lies in the forwarding decision, which usually needs to take into account meeting history and buffer availability. Under opportunistic node-to-node contacts, it's beneficial to use multi-copy schemes to search for a path to the destination. Compared to single-copy schemes, multi-copy schemes enjoy better delivery performance (i.e., lower delivery delay and higher delivery probability), sometimes at the expense of more transmission overhead and buffer occupancy; furthermore, a recovery scheme is usually deployed to delete obsolete copies once a packet is first delivered, to avoid useless transmissions. The majority of routing schemes previous proposed are multi-copy schemes. For example, epidemic routing proposed by Vahdat and Becker [110] essentially floods the whole network in order to deliver a packet. Using all transmission opportunities to achieve minimum delivery delay, epidemic routing incurs maximum

resource consumption, and causes congestion in loaded networks. Many variations of epidemic routing that trade-off delivery delay for resource consumptions have been subsequently proposed and studied, for example, K-hop, probabilistic forwarding [43] and spray-and-wait [104, 101, 102] schemes.

Secondly, different routing schemes make different assumptions about node mobility, and vary in whether and how they leverage mobility properties. Generally speaking, *stateful routing* schemes assume heterogeneous node mobility and make use of the heterogeneity in the forwarding/copying decision. For example, a couple of routing schemes including probabilistic routing [82], history-based routing [61], utility-based routing [22], MaxProp [18], Mobility Pattern Space Routing [76], and [103] have assumed that two nodes that meet more often (or more recently) in the past are more likely to meet again soon; and thus the forwarding/copying decision are made based on history information about past meetings. More recently, Hui *et al.* [52, 51] have studied the identification of social structure within human mobility traces, and demonstrated the benefit of routing schemes that leverage such social structures. Routing schemes such as epidemic routing [110], K-hop, probabilistic forwarding [43] and spray-and-wait [104, 101, 102], do not assume or leverage any structure within the node mobility, and do not make use of such information; they fall into the category of *stateless routing* schemes.

Thirdly, while most previously proposed schemes forward or duplicate packets in the network without modifying packet contents, a number of source coding [111] or network coding based schemes [112, 116] have also been proposed and shown to be beneficial. Source coding based schemes [111, 24] source-erasure-codes a source packet into a large number of blocks, and forwards these blocks via a large number of paths, in order to increase path diversity and hence improve worst case performance using a fixed overhead. For DTNs with resource (bandwidth and buffer) constraints, Random Linear Coding (a special form of randomized network coding) based schemes

are shown to improve block delivery delay versus transmissions overhead trade-off due to its increased randomness [116].

Finally, previous proposed routing schemes differ in their assumptions about resource constraints. Real DTNs are often subject to sometimes severe resource constraints. First, transmission bandwidth is often limited due to the low data rates supported by wireless radio communication, and the limited duration of node-to-node encounters. In addition, applications such as mobile sensor networks often use small battery-powered nodes (sometimes with solar panels to harvest solar power), hence energy and memory capacity are also scarce resources. Considering ZebraNet [61] as an example, the collars attached to zebras use a low-power, short range radio (100m, 19.2Kbps) for peer-to-peer communication, and only have 640KB of user-accessible FLASH RAM memory. While most early routing schemes have ignored resource constraints, or only considered buffer constraints, several recent works [61, 18, 12] have addressed transmission scheduling and buffer management problems. In particular, Balasubramanian *et al.* [12] formulated the DTN routing under resource constraints as a utility-driven resource allocation problem that can be configured to explicitly optimize certain routing metrics such as average delivery delay or maximum delivery delay. Using a per-packet utility function derived from the routing metric, a heuristic algorithm is developed to schedule packet transmission and manage buffer.

1.2 Performance and Mobility Traces Modeling

Early performance evaluations of DTN routing schemes have been mainly carried out through simulation [110]; recent years have seen considerable analytic modeling efforts [101, 39, 104, 103].

Generally, these simulation and modeling works assume synthetic mobility models such as random waypoint, random direction, and random walk models (interested readers are referred to [20] for a survey on mobility models). Most of these syn-

thetic models assume independent and identically random movements of the mobile nodes, and result in memoreless property or exponential tail of the induced *meeting times* distributions [105, 9, 59]. Using such synthetic mobility models makes analytic modeling possible, and in some cases it allows the derivation of closed-form results for performance metrics of interest [39, 117]. The drawback of using such synthetic mobility models is that these models haven't been validated using real DTN mobility traces, and the independent and identical assumption is questionsable. Hence it is unclear how accurately the performance predictions reflect what would be experienced in real deployed systems.

Although some synthetic mobility models proposed in the past have accounted for grouping behavior (a.k.a. group mobility models [20]), social structure of nodes [89] or geographical constraints (such as streets), they are generally not widely adopted, partly due to the lack of validation results for such models as well.

For the above reasons, the collection, analysis and modeling of mobility traces of real DTN networks is a crucial problem. Several recent studies [21, 18, 107, 49, 64, 52, 51] have characterized traces collected from actual mobile networks with intermittent connectivity or adapted from traces collected from wireless LAN, and/or evaluated the impact of the measured mobility on DTN applications through simulation or analysis. These works characterized certain certain aspects of traces, e.g., the *aggregate* inter-contact time, without considering which aspects of the underlying mobility are the most important factors in determining DTN performance and therefore need to be captured and modeled accurately.

1.3 Our Contributions

In this thesis, we study three related fundamental problems in DTNs.

First, we propose an Ordinary Differential Equation (ODE) based framework for the modeling of epidemic style routing. We derive the ODE models from Markovian

models under appropriate scaling (when the network size increases). We also show that such models allow one to derive or calculate important performance metrics (i.e., delivery delay and total number of copies generated) much more easily than Markovian models. Although we mainly consider networks where there is no bandwidth constraint for both the cases of with or without buffer constraints, extending the model to further consider bandwidth constraints and contention is not difficult.

Our second contribution is the study of the benefit of network coding in DTNs. For realistic DTNs where bandwidth and buffer are constrained, we compare the performance of the non-coding based approach, and Random Linear Coding (abbreviated as RLC, a specific form of randomized network coding) based scheme. We find that an RLC based scheme can decrease the block delivery delay, and improve the block delivery delay versus the number of transmissions made trade-off. The relative benefit of the RLC scheme is especially significant when both bandwidth and buffer space are constrained.

The third contribution of this thesis is a careful modeling study of the mobility trace collected from UMass DieselNet, an operational bus based DTN. We analyze the bus-to-bus contact traces and characterize the contact processes between buses and its impact on DTN routing performance. We find that the all-bus-pairs aggregated inter-contact times show no discernible pattern. However, the inter-contact times aggregated at a route level exhibit periodic behavior. Based on the analysis of the deterministic inter-meeting times for bus pairs running on route pairs, and considering the variability in bus movement and the random failures to establish connections, we construct generative route-level models that capture the above behavior, allowing one to generate synthetic DTN mobility traces. Through trace-driven simulations of epidemic routing, we find that the epidemic performance predicted by traces generated with this finer-grained route-level model are much closer to the actual performance that would be realized in the operational system than traces generated

using the coarse-grained all-bus-pairs aggregated model. This suggests the importance of choosing the right level of model granularity when modeling mobility-related measures such as inter-contact times in DTNs.

1.4 Structure of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we presente our performance modeling studies of epidemic style routing using an ODE based framework. In Chapter 3, we explore the benefit of applying network coding to unicast applications in DTNs. In Chapter 4, we present our modeling studies of the mobility trace collected from UMass DieselNet. We draw conclusions and discuss future research directions in Chapter 5.

CHAPTER 2

PERFORMANCE MODELING OF EPIDEMIC ROUTING

2.1 Introduction

Epidemic routing [110] has been proposed as an approach for routing in sparse and/or highly mobile networks in which there may not be a contemporaneous path from source to destination. It adopts the "store-carry-forward" paradigm – a node receiving a packet buffers and carries that packet as it moves, passing the packet on to new nodes that it encounters. Analogous to the spread of infectious diseases, each time a packet-carrying node encounters a node that does not have a copy of that packet, the carrier is said to *infect* this new node by passing on a packet copy; newly infected nodes, in turn, behave similarly. The destination receives the packet when it first meets an infected node. When the traffic load is very low, epidemic routing achieves the minimum delivery delay at the expense of increased use of resources such as buffer space, bandwidth, and transmission power. However this also leads to link and/or storage congestion when the network is loaded. Variations of epidemic routing have recently been proposed that exploit the tradeoff between delivery delay and resource consumption, including K-hop schemes [99, 39], probabilistic forwarding [82, 43], and spray-and-wait [104, 101]. These different schemes differ in their "infection process", i.e., the spreading of a packet in the network. They need to be combined with a so-called "recovery process" that deletes copies of a packet at infected nodes, following the successful delivery of the packet to the destination. Different recovery schemes have been proposed: some are simply based on timers, others actively spread

the information that a copy has been delivered to the destination throughout the network [43].

Early efforts evaluating the performance of epidemic routing schemes used simulation [110, 61, 82]. More recently, Markovian models have been developed to study the performance of epidemic routing [100, 39, 43], 2-hop forwarding [39], and spray-and-wait [104, 101]. Recognizing the similarities between epidemic routing and the spread of infectious diseases, [100, 43] used ordinary differential equation (ODE) models adapted from infectious disease-spread modeling [27] to study the source-to-destination delivery delay under the basic epidemic routing scheme, and then adopted Markovian models to study other performance metrics.

In this chapter, we develop a rigorous, unified framework, based on Ordinary Differential Equations (ODEs), to study epidemic routing and its variations. The starting point of our work is [39], where the authors considered common node mobility models (e.g., random waypoint and random direction mobility) and showed that nodal inter-meeting times are approxmiately exponentially distributed when transmission ranges are small compared to the network area, and node velocity is sufficiently high. This observation suggests that Markovian models of epidemic routing can lead to quite accurate performance predictions; indeed [39] developed Markov chain models for epidemic routing and 2-hop forwarding, deriving the average source-to-destination delivery delay and the number of extant copies of a packet at the time of delivery. An analytical study of such Markov chain models is quite complex for even simple epidemic schemes, and more complex schemes have defied analysis thus far. Moreover, numerical solution of such models becomes impractical when the number of nodes is large.

We develop ODEs as a fluid limit of Markovian models such as [39], under an appropriate scaling as the number of nodes increases. Through the chapter we show that the ODE approach is a valid tool for investigating epidemic style routing. In fact

this approach allows us to derive closed-form formulas for the performance metrics considered in [39], obtaining matching results. More importantly, we are also able to use the ODE framework to further model the recovery process, to study more complex variants of epidemic routing, and to model the performance of epidemic routing with different buffer management schemes under buffer constraints. While different recovery processes are also studied in [43] using Markov chains, simulation is first needed to determine a number of model parameters. Many of our ODE models can be analytically solved, providing closed-form formulas for the performance metrics of interest; in cases where we resort to numerical solution, the computational complexity does not increase with the number of nodes. The drawback of our ODE models is that they provide the moments of the various performance metrics of interest, while numerical solution of Markov chain models can provide complete distributions (e.g., for the number of packet copies in the system). Simulation results show good agreement with the predictions of our ODE models.

The main purpose of this work is to show how ODE models can be advantageously employed to study the performance of various epidemic style routing schemes, rather than to provide final conclusions about the merits of specific schemes. Nevertheless we have obtained insights into different epidemic routing schemes through our models. In particular, we have identified rules of thumb for configuring these schemes, we have shown the existence of a linear relation between total number of copies sent and the buffer occupancy under certain schemes, and we have demonstrated that the relative benefit of different recovery schemes depends strongly on the specific infection process. Finally our analysis of buffer-constrained epidemic routing suggests that sizing node buffers to limit packet loss is not vital as long as an appropriate buffer management scheme is used.

The remainder of this chapter is structured as follows. Basic epidemic routing and our basic ODE model are described and derived in Section 2.2, allowing one to

characterize the source-to-destination delivery delay, the number of copies made for a packet, and the average buffer occupancy. In Section 2.3, it is shown how the ODE model can be easily extended to model three important variations of basic epidemic routing, K-hop forwarding, probabilistic forwarding and limited-time forwarding, to global timeout scheme, and to model signaling overheads. In Section 2.4, we perform validation for these models through simulation. We use the proposed models to characterize the tradeoff between delivery delay and resource consumption (buffer occupancy, number of copies made) in Section 2.5. In Section 2.6, we integrate the ODE models with Markov and fluid buffer models to study the effect of finite buffers, and compare different buffer management strategies. In Section 2.7, we review related works and compare our work with them. Finally, in Section 2.8 we summarize the chapter.

2.2 Basic Epidemic Routing

In this section we develop our ODE model for basic epidemic routing [110], after briefly describing epidemic routing and the scenario we are considering. We then use the model to study three different recovery techniques for deleting packet copies after the delivery of the packet.

We consider a set of $N+1$ nodes, each with a finite transmission range, moving in a closed area, and different source-destination pairs. We say that two nodes "meet" when they come within transmission range of each other, at which point they can exchange packets. Let us focus on a single packet. The analogy with disease spreading is useful in describing epidemic routing. The source of the packet can be viewed as the first carrier of a new disease, the first *infected* node, which copies the packet to (infects) every node it meets. These new infected nodes act in the same way. As a result, the population of *susceptible* nodes (i.e., nodes without a copy of the packet) decreases over time. Once a node carrying the packet meets the destination,

notation	meaning
N	number of nodes in the network, excluding the source
β	pair-wise meeting rate
B	node buffer size (in packets)
λ	per-flow packet arrival rate
K	Maximum number of hops in K-hop scheme
p	Forwarding probability under probabilistic forwarding scheme
T	TTL (Time-To-Live) value for global timeout scheme
μ	Exponential time out value for limited time forwarding
T_d	packet delivery delay
L	average packet lifetime
G	number of times a packet is copied during its lifetime
C	number of copies a packet in the network at delivery time
Q	per-node data packet buffer occupancy
G_h	number of times the packet header is copied during the packet's lifetime
G_a	number of times the anti-packet is copied during the packet's lifetime
Q_{anti}	per-node anti-packets buffer occupancy
$i(t)$	the average fraction of infected nodes at time t
$r(t)$	the average fraction of recovered nodes at time t
$I(t)$	the average number of infected nodes at tiem t
$R(t)$	the average number of recovered nodes at tiem t
$P(t)$	the CDF for delivery delay T_d

Table 2.1. Tables of notations used in Chapter 2

it passes the packet on to the destination, deletes the packet from its own buffer, and retains "packet-delivered" information (an "anti-packet"), which will prevent it from receiving another copy of this packet in the future. Such a node is said to have *recovered* from the disease. Here the recovery process simply relies on meeting with the destination. We will shortly consider more sophisticated recovery schemes. Table 2.1 explains the notations used throughout this chapter.

Consider now many packets spreading at the same time in the network. We assume that when two nodes meet they can exchange an arbitrary number of packets, and each node has enough buffer to store all packets (the latter assumption is relaxed in Section 2.6), thus allowing different infections to be considered independently. We

also assume a mechanism exists so that nodes never exchange a packet if both nodes are already carrying a copy of that packet (more details in Section 2.3.3).

2.2.1 ODE Models for Basic Epidemic Routing

As noted earlier, Groenevelt *et al.* [39] showed that the pairwise meeting time between nodes is nearly exponentially distributed, if nodes move in a limited region (of area A) according to common mobility models (such as the random waypoint or random direction model [13]) and if their transmission range (d) is small compared to A, and their speed is sufficiently high. The authors also derived the following estimate of the pairwise meeting rate β:

$$\beta \approx \frac{2wd\mathrm{E}[V^*]}{A}, \tag{2.1}$$

where w is a constant specific to the mobility model, and $\mathrm{E}[V^*]$ is the average relative speed between two nodes. Under this approximation, [39] showed that the evolution of the number of infected nodes can be modeled as a Markov chain.

We introduce our modeling approach starting from the Markov model for basic epidemic routing before the delivery of a copy to the destination. Given $n_I(t)$, the number of infected nodes at time t, the transition rate from state n_I to state $n_I + 1$ is $r_N(n_I) = \beta n_I(N - n_I)$, where N is the total number of nodes in the network (excluding the destination). If we rewrite the rates in a "density dependent form", as $r_N(n_I) = N\lambda(n_I/N)(1 - n_I/N)$ and assume that $\lambda = N\beta$ is constant, we can apply Theorem 3.1 in [75] to prove that, as N increases, the fraction of infected nodes (n_I/N) converges asymptotically to the solution of the following equation[1]:

$$i'(t) = \lambda i(t)(1 - i(t)), \text{ for } t \geq 0 \tag{2.2}$$

[1] Formally, $\forall \epsilon > 0$, $\lim_{N\to\infty} \mathrm{Prob}\{|\sup_{s\leq t}\{n_I(s)/N - i(s)\}| > \epsilon\} = 0$

with initial condition $i(0) = \lim_{N\to\infty} n_I(0)/N$. The average number of infected nodes then converges to $I(t) = Ni(t)$ in the sense of footnote 1. The following equation can be derived for $I(t)$ from Eq.(2.2):

$$I'(t) \;=\; \beta I(N - I), \tag{2.3}$$

with initial condition $I(0) = Ni(0)$. Such an ODE, which, as we have shown, results as a fluid limit of a Markov model as N increases, has been commonly used in epidemiology studies, and was first applied to epidemic routing in [100] as a reasonable approximation.

We remark that 1) the initial population of infected nodes must scale with N, and 2) the pairwise meeting rate must scale as $1/N$. Eq.(2.1) provides insight into the physical interpretation of this meeting rate scaling: in particular if the area A increases with N, keeping node density constant, then β scales with $1/A$, i.e.,$1/N$. In the following we will consider Eq.(2.3) with initial condition $I(0) = 1$, which corresponds to an initial fraction of infected nodes $i(0) = 1/N$. Despite the "small" number of initial infected nodes, we will see via simulation that the approximation is good. We also note that Eq.(2.3), as well as other related equations we will derive shortly, can also be obtained in a different manner from Markovian models by neglecting terms related to higher moments (the details are given in Appendix A).

2.2.2 Delay under Epidemic Routing

Let T_d be the packet delivery delay, i.e., the time from when a packet is first generated at the source to the time when it is first delivered to the destination, and denote its cumulative distribution function (CDF) by $P(t) = \text{Prob}(T_d < t)$. Under the same scaling and approximations considered earlier, we can derive the following equation for $P(t)$: $P'(t) = \lambda i(1 - P)$, where $i(t)$ is the solution of Eq.(2.2). Let us

consider $P_N(t)$, the CDF of T_d when the number of nodes in the system is $N+1$, i.e., there are N nodes plus one destination node. We have

$$
\begin{aligned}
&P_N(t+dt) - P_N(t) = \text{Prob}\{t \le T_d < t+dt\} \\
&= \text{Prob}\{\text{destination meets an infected node in } [t, t+dt] \mid T_d > t\} \\
&\qquad \times \text{Prob}\{T_d > t\} \\
&= \text{Prob}\{\text{destination meets one of the } n_I(t) \text{ infected nodes in } [t, t+dt]\,\} \\
&\qquad \times (1 - P_N(t)) \\
&= \text{E}[\text{Prob}\{\text{destination meets one of the } n_I(t) \text{ infected nodes in } [t, t+dt] \mid n_I(t)\}] \\
&\qquad \times (1 - P_N(t)) \\
&\approx \text{E}[\beta n_I(t) dt]\,(1 - P_N(t)) \\
&= \beta \text{E}[n_I(t)](1 - P_N(t)) dt \\
&= \lambda \frac{\text{E}\,[n_I(t)]}{N} (1 - P_N(t)) dt.
\end{aligned}
$$

Note that $n_I(t)$ is the number of infected nodes at time t, given that the destination has not received a copy of the packet. It implicitly accounts for the condition $T_d > t$. The following ODE for $P_N(t)$ follows from the above equation:

$$
P'_N(t) \;=\; \lambda \text{E}\left[\frac{n_I(t)}{N}\right](1 - P_N(t)).
$$

As N increases, $\text{E}[n_I(t)/N]$ converges to $i(t)$, and $P_N(t)$ converges to the solution of $P'(t) = \lambda i(t)(1 - P(t))$.

For a finite population of size N we can consider:

$$
P'(t) \;=\; \beta I(t)(1 - P(t)). \tag{2.4}
$$

Eq.(2.4) was proposed in [100], based on an analogy with a Markov process. Solving Eq.(2.3) and Eq.(2.4) with initial conditions, $I(0) = 1, P(0) = 0$ yields

$$I(t) = \frac{N}{1 + (N-1)e^{-\beta N t}}, \quad P(t) = 1 - \frac{N}{N - 1 + e^{\beta N t}}.$$

From $P(t)$, the average delivery delay can be explicitly found as:

$$\mathrm{E}[T_d] = \int_0^\infty (1 - P(t))dt = \frac{\ln N}{\beta(N-1)}. \tag{2.5}$$

The average number of copies of a packet in the system when the packet is delivered to the destination under epidemic routing, $\mathrm{E}[C_{ep}]$, can also be derived, as it coincides with the average number of infected nodes in the system, apart from the source, when the packet is delivered. Let $Q(t) := 1 - P(t)$, we have:

$$\begin{aligned}
\mathrm{E}[C_{ep}] &= \int_0^\infty I(t)P'(t)dt - 1 \\
&= \beta \int_0^\infty I^2(t)Q(t)dt \text{ , replacing } P'(t) \text{ by Eq.(2.4)} \\
&= \beta \int_0^\infty \frac{I^2(t)Q(t)}{-\beta I(t)Q(t)} d(Q(t)), \text{ by Eq.(2.4)} \\
&= \int_0^\infty I'(t)Q(t)dt + 1, \text{ by integrating by parts.}
\end{aligned}$$

By replacing $I'(t)$ with Eq.(2.3), and considering that $\int_0^\infty \beta I(t)Q(t)dt = P(\infty) - P(0) = 1$, we get: $C_{ep} = \frac{N-1}{2}$.

Using a Markov chain model, [39] obtained the same results for the number of copies, computed the Laplace-Stieltjes Transform (LST) of the delay distribution, and from the LST found the following asymptotic expression for the average delay as $N \to \infty$: $\frac{1}{\beta(N-1)}(\ln N + \gamma + O(\frac{1}{N}))$, matching Eq.(2.5). We note that the derivation is much simpler using our ODE model.

2.2.3 Recovery from Infection

In the previous two subsections, we derived the ODE models to model the infection process of epidemic routing, and obtained delivery delay, and the number of copies made at delivery time. In this section, we further consider the recovery process by studying several recovery schemes previously proposed in [43].

Clearly, once a node delivers a packet to the destination, it should delete the packet from its buffer to save storage space and prevent the node from infecting other nodes. Moreover, to avoid being reinfected by the packet, the node can keep record about the packet's delivery. We refer to this information stored at the node as "anti-packet", and refer to this scheme of handling already-delivered packets as the IMMUNE scheme. A more aggressive approach toward deleting obsolete copies is to propagate anti-packets among nodes. An anti-packet can be propagated only to infected nodes (which we will refer to as the IMMUNE_TX scheme), or to both infected and susceptible nodes (VACCINE scheme). We study the following two metrics for epidemic routing under these different recovery schemes. One is the average number of times a packet is copied during its lifetime, excluding the copy to the destination, denoted as $\mathrm{E}[G]$. This value is greater than or equal to $\mathrm{E}[C]$, because more copies can be made after the delivery to the destination. This metric is strongly related to the bandwidth requirement, and transmission power consumption of a specific scheme. The other is the average buffer occupancy at each node $\mathrm{E}[Q]$, for which we are going to derive an expression under a specific traffic pattern. The two metrics are related to each other and they both depend on the specific recovery process.

In order to study these two metrics, we next derive ODE models that take into account the above recovery processes as the limit of Markov models. We note that the derivations require that we scale the number of destinations n_D in a manner similar to the scaling of the number of initially infected nodes, i.e. $\lim_{N\to\infty} n_D/N = d$.

IMMUNE Recovery. Let $n_R(t)$ denote the number of recovered nodes at time t, then the system state can be denoted as $(n_I(t), n_R(t))$. The transition rates (i.e., the infection and recovery rate) can be expressed as:

$$\begin{aligned} r_N((n_I(t), n_R(t)), (n_I(t)+1, n_R(t))) &= \beta n_I(t)(N - n_I(t) - n_R(t)) \\ r_N((n_I(t), n_R(t)), (n_I(t)-1, n_R(t)+1)) &= \beta n_I(t) n_D. \end{aligned}$$

The transition rates can be written in a "density dependent" form, given that the number of destinations n_D scales in a manner similar to the scaling of the number of initially infected nodes. Then by Theorem 3.1 in [75], we get that, as N increases, the fraction of infected nodes (n_I/N) and recovered nodes (n_R/N) converge asymptotically to the solution of the following equations:

$$\begin{aligned} i'(t) &= \lambda i(t)(1 - i(t) - r(t)) - \lambda i(t) d \\ r'(t) &= \lambda i(t) d \end{aligned}$$

where $d = n_D/N$, and the initial conditions are $i(0) = \lim_{N\to\infty} n_I(0)/N, r(0) = 0$.

The number of infected and recovered nodes then converge to $I(t) = Ni(t), R(t) = Nr(t)$ respectively in the sense of footnote 1. The following equation can be derived for $I(t), R(t)$ from the previous ODEs:

$$I'(t) = \beta I(N - I - R) - \beta I n_D \tag{2.6}$$

$$R'(t) = \beta I n_D \tag{2.7}$$

with initial condition $I(0) = Ni(0), R(0) = 0$. We consider $I(0) = 1, R(0) = 0, n_D = 1$.

The above model allows us to evaluate the average number of times that a packet is copied during its lifetime, $\mathrm{E}[G_{ep}]$. In fact the total number of copies made of

a packet equals the number of nodes that have ever been infected, i.e., $\mathrm{E}[G_{ep}] = \lim_{t\to\infty}(I(t)+R(t)) - I(0)$. As $R(t)$ is a strictly increasing function of t, $I(R)$ is well defined. Dividing Eq.(2.6) over Eq.(2.7) yields (we assume $n_D = 1$):

$$dI/dR = N - I - R - 1.$$

Solving this ODE with initial condition $I(R=0) = 1$ yields

$$I(R) = (-N+1)e^{-R} - R + N.$$

As $\lim_{t\to\infty} I(t) = 0$, we can solve $I(R) = 0$ for R to find $\lim_{t\to\infty} R(t)$. For N large enough ($N > 10$), the solution gives $\lim_{t\to\infty} R(t) \approx N$. So we get $\mathrm{E}[G_{ep}] = \lim_{t\to\infty} I(t) + R(t) - 1 \approx N - 1$.

IMMUNE_TX Recovery. For IMMUNE_TX the transition rates are (omitting the dependence from time, t):

$$\begin{aligned} r_N((n_I, n_R), (n_I+1, n_R)) &= \beta n_I (N - n_I - n_R) \\ r_N((n_I, n_R), (n_I-1, n_R+1)) &= \beta n_I (n_R + n_D) \end{aligned}$$

The limiting equations are:

$$\begin{aligned} i'(t) &= \lambda i(t)(1 - i(t) - r(t)) - \lambda i(t)(r(t) + d) \\ r'(t) &= \lambda i(t)(r(t) + d) \end{aligned}$$

The following equations can be immediately derived:

$$I'(t) = \beta I(N - I - R) - \beta I(1 + R) \tag{2.8}$$

$$R'(t) = \beta I(1+R) \tag{2.9}$$

We divide Eq.(2.8) over Eq.(2.9) to get an ODE of $I(R)$. Solving the ODE yeilds

$$I(R) = \frac{-R^2 + (N-1)R + 1}{R+1}.$$

As $\lim_{t\to\infty} I(t) = 0$, we find $\lim_{t\to\infty} R(t)$ by solving $I(R) = 0$ for R. $I(R) = 0$ has two roots $(N-1 \pm \sqrt{N^2-2N+5})/2$. Discarding the negative root, we have $\lim_{t\to\infty} R(t) = (N-1+\sqrt{N^2-2N+5})/2$. Therefore, for IMMUNE_TX scheme, we have

$$\mathrm{E}[G_{ep}(N)] = \lim_{t\to\infty}(I(t)+R(t)-1) = \frac{N-3+\sqrt{N^2-2N+5}}{2}.$$

VACCINE Recovery. For VACCINE we need to specify how many destination nodes have received the packet, let n_{DR} denote this number[2]. We assume that all the destinations have to receive the packets from an infected node[3]. The transition rates are:

$$\begin{aligned}
r_N((n_I, n_R, n_{DR}), (n_I+1, n_R, n_{DR})) &= \beta n_I(N - n_I - n_R) \\
r_N((n_I, n_R, n_{DR}), (n_I-1, n_R+1, n_{DR})) &= \beta n_I(n_R + n_{DR}) \\
r_N((n_I, n_R, n_{DR}), (n_I-1, n_R+1, n_{DR}+1)) &= \beta n_I(n_D - n_{DR}) \\
r_N((n_I, n_R, n_{DR}), (n_I, n_R+1, n_{DR})) &= \beta(N - n_I - n_R)(n_R + n_{DR}).
\end{aligned}$$

[2]There is no such a need for the previous schemes because only a destination can recover an infected node. Hence even if the destination has not received the packet, the destination receives it when it meets the infected node.

[3]Different assumptions can be made, for example a destination could receive the packet from another destination, or a destination could receive the antipacket from a recovered node and propagate it without having received the packet. The latter case is meaningful when we deal with an anycast communication (the packet has to reach at least one of the destinations) or if we can rely on the fact all the destinations will receive a copy of the packet from the destination that started the recovery process. These different assumptions lead to minor differences in the final equations.

The limiting equations are as follows, where $d_r(t) = \lim_{N\to\infty}(n_{DR}/N)$:

$$\begin{aligned} i'(t) &= \lambda i(t)(1 - i(t) - r(t)) - \lambda i(t)(r(t) + d) \\ r'(t) &= \lambda i(t)(r(t) + d) + \lambda(1 - i(t) - r(t))(r(t) + d_r(t)) \\ d_r'(t) &= \lambda i(t)(d - d_r(t)). \end{aligned}$$

If we consider the average populations ($Ni(t)$,$Nr(t)$ and $Nd_r(t)$), and assume $n_D = 1$, we observe that $Nd_r(t)$ satisfies the same ODE as $P(t)$, and derive the following equations:

$$I'(t) = \beta I(t)(N - I(t) - R(t)) - \beta I(t)(R(t) + 1) \quad (2.10)$$

$$R'(t) = \beta I(t)(1 + R(t)) + \beta(N - I(t) - R(t))(R(t) + P(t)) \quad (2.11)$$

$$P'(t) = \beta I(t)(1 - P(t)). \quad (2.12)$$

Let $C(t)$ be the number of nodes that are ever infected by the packet, then we have

$$C'(t) = \beta I(t)(N - I(t) - R(t)) \quad (2.13)$$

Solving the ODEs (Eq.(2.10),(2.11),(2.12), (2.13)) allows us to evaluate the number of times a packet is copied during its lifetime (excluding the copy to the destination), as $G_{ep_vac} = C(\infty)$.

Buffer Occupancy. We next consider the average buffer occupancy E[Q], in the case of $N + 1$ unicast flows, with each node being the source of one flow and the destination for one other flow. The packet generation process in each flow is a

Poisson process with rate λ. We denote by L the average packet lifetime (the time from when the packet is generated by the source node to when all copies of the packet are removed from the network). As the total arrival rate of new packets to the system is $(N+1)\lambda$, the average number of packets in the system is $(N+1)\lambda L$ by Little's Law.

The average number of copies of a packet in the system during its lifetime can be evaluated as $\int_0^\infty I(t)dt/L$, where $I(t)$ is the solution to the ODEs that account for the recovery process.

Therefore the average total buffer occupancy in the whole network is given by $\mathrm{E}[Q_{total}] = (\int_0^\infty I(t)dt/L)(N+1)\lambda L = \int_0^\infty I(t)dt(N+1)\lambda$, and the per-node average buffer occupancy is thus $\mathrm{E}[Q] = \lambda \int_0^\infty I(t)dt$.

Next, we demonstrate that modeling a node's buffer as an $M/M/\infty$ queue gives the same result and shows a linear relationship between the average buffer occupancy and the number of copies made under the IMMUNE scheme. In fact, given that each packet is copied $\mathrm{E}[G_{ep}]$ times during its lifetime, each flow generates relay traffic at rate $\mathrm{E}[G_{ep}]\lambda$, and the total rate of relay traffic in the network is $\mathrm{E}[G_{ep}]\lambda(N+1)$ (as there are $N+1$ flows). This traffic is equally divided among the $N+1$ nodes, hence the arrival rate of *r*elay packets to each node is $\mathrm{E}[G_{ep}]\lambda$, and the total packet arrival rate is $\lambda(1+\mathrm{E}[G_{ep}])$. If a copy is deleted only when the node meets the destination[4], the service rate is $1/\beta$. Under the $M/M/\infty$ queue, the average buffer occupancy is $\mathrm{E}[Q] = \frac{\lambda}{\beta}(1+\mathrm{E}[G_{ep}])$.

2.3 Extended Model

The schemes in the previous section all share the same infection process: they propagate a packet among nodes in a flooding/epidemic manner, but differ in the way

[4]This is the case under IMMUNE for the basic epidemic routing, and also for the probabilistic and K-hop forwarding schemes we consider in Section 2.3.1.

they counteract the infection after the packet has been delivered to the destination. As results in Table 2.2 show, this can lead to substantial differences in terms of buffer occupancy and the total number of copies made of a packet. Depending on the specific applications, it might be preferable to trade off timely delivery for savings in resource consumption, by changing the way packets are propagated among nodes. We describe in Section 2.3.1 K-hop forwarding, probabilistic forwarding and limited-time forwarding that allow us to achieve such tradeoff. In Section 2.3.2, we introduce the global timeout scheme that naturally addresses the problem of deleting anti-packets. We discuss how ODE models can be used to model signaling overhead in Section 2.3.3. All the ODEs models can be derived as limits of Markovian models, similarly to what we have shown in Section 2.2.1. We do not detail the derivations, but only stress the peculiarities (if any) to be taken into account when applying the limiting theorem.

2.3.1 Trade-off Schemes

In this section, we present ODEs models for the following variations of the basic epidemic routing scheme, i.e., K-hop forwarding, probabilistic forwarding and limited-time forwarding. They all allow one to trade off timely delivery delay for savings in transmission bandwidth and buffer occupancy.

2.3.1.1 K-hop Forwarding

Under K-hop forwarding, a packet can traverse at most K hops to reach the destination. We can use ODE models to model the K-hop forwarding scheme, as we demonstrate in the following for $K = 2$ case.

Under 2-hop forwarding, the source copies the packet to every node it meets until it meets the destination; relay nodes do not copy the packet to any other node except the destination. As the packet spreads at a rate proportional to the number of

susceptible nodes, the following equations model the spread process and the delivery delay:

$$\begin{aligned} I'(t) &= \beta(N-I) \\ P'(t) &= \beta I(1-P) \end{aligned}$$

with initial conditions: $I(0) = 1$ and $P(0) = 0$. Note that to derive the previous equations from the Markovian model similarly to what we did in Section 2.2.1, the number of source nodes need to scale with N.

The above ODE system can be solved explicitly, from which we can then derive an asymptotic expression for the average delivery delay $\mathrm{E}[T_d] \approx \frac{1}{\beta}\int_0^\infty e^{-(N-1)t^2/2}dt = \frac{1}{\beta}\sqrt{\frac{\pi}{2}}\frac{1}{\sqrt{N-1}}$ (derivation detail is given in Appendix B). As to the average number of copies until delivery, C_{2hop}, a derivation similar to that of epidemic routing in Section 2.2.1 yields $C_{2hop} = \beta N\mathrm{E}[T_d] - 1 \sim_{N\to\infty} \sqrt{\frac{\pi}{2}}\sqrt{N}$. These results again match those obtained in [39] using a Markov Chain model.

We can apply analysis similar to Section 2.2.3 to study the number of copies made and the average buffer occupancy for given recovery schemes. For IMMUNE recovery, we obtain more accurate model through the following derivations. Let G_{2hop} be the number of times a packet is copied during its life time (excluding the copy to the destination) under this scheme. For each packet, the source node copies the packet to every relay node it meets before it meets the destination, therefore $G_{2hop}(N)$ equals the number of nodes the source node meets before meeting the destination. As the inter-meeting times between all node pairs are independently identically distributed (i.i.d.) exponential random variables, the destination node is equally likely to be the i-th node to meet the source node, for $i = 1, ..., N$. Therefore we have $Pr(G_{2hop} = i) = \frac{1}{N}$, for $i = 0, ..., N-1$, and hence $\mathrm{E}[G_{2hop}] = \frac{N-1}{2}$. Given $G_{2hop}(N)$, we can derive the average buffer occupancy using a $M/M/\infty$ model with the departure rate β, by an approach similar to what we described in Section 2.2.3.

Schemes	$I(t)$ $P(t)$	$\mathrm{E}[T_d]$	$\mathrm{E}[C], \mathrm{E}[G]$	$\mathrm{E}[Q]$
Epidemic routing	$I(t) = \frac{N}{1+(N-1)e^{-\beta Nt}}$ $P(t) = 1 - \frac{N}{N-1+e^{\beta Nt}}$	$\frac{\ln N}{\beta(N-1)}$	$\mathrm{E}[C] = \frac{N-1}{2}$ $\mathrm{E}[G] \approx N-1$ (IM) $\mathrm{E}[G] = \frac{N-3}{2}$ $+\frac{\sqrt{N^2-2N+5}}{2}$ (IM_TX)	$\approx N\lambda/\beta$ (IM) $\approx \lambda\frac{N-1+\sqrt{N^2-2N+5}}{2\beta}$ (IM_TX)
2-hop forwarding	$I(t) = N-(N-1)e^{-\beta t}$ $P(t) = 1 - e^{N-1-\beta Nt-(N-1)e^{-\beta t}}$	$\frac{1}{\beta}\sqrt{\frac{\pi}{2}}\frac{1}{\sqrt{N-1}}$	$\mathrm{E}[C] = \sqrt{\frac{\pi}{2}}\sqrt{N}$, $G = \frac{N-1}{2}$ (IM)	$\frac{\lambda(N+1)}{2\beta}$ (IM)
Prob. forwarding	$I(t) = \frac{N}{1+(N-1)e^{-p\beta Nt}}$ $P(t) = 1 - (\frac{N}{N-1+e^{p\beta Nt}})^{1/p}$	$[\frac{\ln(N)}{\beta(N-1)}, \frac{\ln(N)}{\beta p(N-1)}]$	$\mathrm{E}[C] = \frac{p(N-1)}{1+p}$	
Limited-time forwarding (no reinfection)	$I(t) = \frac{a_2+e^{\beta(a_2-a_1)t}+A}{A+e^{\beta(a_2-a_1)t}}$ $a_{1,2} = \frac{(\beta N-\mu)\mp\sqrt{(\beta N-\mu)^2+4\beta\mu}}{2\beta}$ $a_1<0, a_2>0$, $A = \frac{a_2-1}{1-a_1}$	$\underset{N\to\infty}{\sim} \frac{1}{\beta}\frac{\ln(N-\frac{\mu}{\beta})}{N-\frac{\mu}{\beta}}$ $\underset{\mu\to\infty}{\sim} \frac{\mu-N\beta}{\beta\mu}$ $\underset{N=\frac{\mu}{\beta}\to\infty}{\sim} \frac{\pi}{2\beta\sqrt{N-2}}$		
Global timeout scheme	$P(t) = 1 - \frac{N}{N-1+e^{\beta Nt}}, t \le T$ $P(t) = 1 - e^{\beta b(T-t)}$ $\frac{N}{N-1+e^{\beta NT}}, t>T, b = I(T)$	$\frac{\ln(N)}{\beta(N-1)}$ $-\frac{\ln(1+(N-1)e^{-\beta NT})}{\beta(N-1)}$ $+\frac{1}{\beta e^{\beta NT}}$	$\mathrm{E}[C] = \frac{N-1}{2}$ $-\frac{N^2(N-1)}{2(e^{\beta TN}+N-1)^2}$	
Global timeout scheme (2)	$P(t) = 1 - \frac{N}{N-1+e^{\beta Nt}}, t \le T$ $P(t) = 1 - \frac{Ne^{\beta(T-t)}}{N-1+e^{\beta NT}}, t>T$	$\frac{\ln(N)}{\beta(N-1)}$ $-\frac{\ln(1+(N-1)e^{-\beta NT})}{\beta(N-1)}$ $+\frac{N}{\beta(e^{\beta TN}+N-1)}$	$\mathrm{E}[C] = \frac{N-1}{2}$ $-\frac{N^2(N-1)}{2(e^{\beta TN}+N-1)^2}$	

Table 2.2. Summary of closed-form expressions obtained for different schemes

2.3.1.2 Probabilistic Forwarding

Probabilistic forwarding is similar to epidemic routing except that when two nodes meet, each node accepts a relay packet with probability p. When $p = 0$, the probabilistic forwarding degenerates to direct source-destination delivery; when $p = 1$, epidemic routing is performed. Varying p in the range $(0, 1)$ allows a trade-off between storage/transmission requirements and delivery delay. We can model the delivery delay using the following ODEs:

$$\begin{aligned} I'(t) &= \beta p I(N - I) \\ P'(t) &= \beta I(1 - P) \end{aligned}$$

with $I(0) = 1, P(0) = 0$. We derive a closed-form solution for this ODEs, from which we then derive bounds for the average delay (details given in Appendix B, and the results are given in Table 2.2). Close-form formula for the number of copies at delivery time is derived using similar technique described in Section 2.2.2: $C_{prob} = \frac{p(N-1)}{1+p}$.

Similar to the basic epidemic routing case, we extend the ODE models to consider recovery process, and calculate the average total number of copies made of a packet (G_{prob}) and the average buffer occupancy for probabilistic forwarding under different recovery schemes.

2.3.1.3 Limited-time Forwarding

Under limited-time forwarding, when a *relay* node accepts a packet copy, it starts a timer with duration drawn from an exponential distribution with rate μ. When the timer expires, the copy is deleted from the buffer. To guarantee the eventual delivery of each packet, a node does not time out a packet for which it is the original source. The choice of timeout value allows us to trade off the delivery delay against storage and number of transmissions.

When a packet copy in a node times out, the node can either store an anti-packet (so that it will not be infected by the packet again), or keep no information (in which case it become susceptible to the packet again).

The former scheme can be modeled using the following ODEs, where $T(t)$ is the number of timed out nodes at time t. As before these ODEs can be derived as the limit of a Markovian model[5].

$$\begin{aligned} I'(t) &= \beta I(N - I - T) - \mu(I - 1) \\ T'(t) &= \mu(I - 1) \\ P'(t) &= \beta I(1 - P) \end{aligned}$$

We numerically solved this ODEs, and evaluate the average delivery delay as $\mathrm{E}[T_d] = \int_0^\infty (1 - P(t))dt$. Similar to epidemic routing, by extending the ODEs to include recovery processes, we are able to evaluate numerically the average total number of copies made for a packet during its lifetime $\mathrm{E}[G]$, and the average buffer occupancy $\mathrm{E}[Q]$.

The latter scheme can be studied using the following ODEs:

$$\begin{aligned} I'(t) &= \beta I(N - I) - \mu(I - 1) \\ P'(t) &= \beta I(1 - P) \end{aligned}$$

The above ODEs can be solved explicitly and an asymptotic expression for the average delay can be found (see Table 2.2 for the results, and Appendix B for derivation detail). We found that if $\mu \geq N\beta$ the number of infected nodes goes to zero as $t \to \infty$. In this case limited-time forwarding can perform recovery via timeout and there is no

[5]There is no need to scale the timer rate μ, while we need to scale β as we noted in Section 2.2.1.

need for explicitly transmitted anti-packets, and the epidemic spreading will eventually die out. The asymptotic delay when $\mu = N\beta$ equals $\frac{\pi}{2\beta\sqrt{N-2}}$ (derivation detail is given in Appendix B).

2.3.2 Handling Anti-packets: Global Timeout Scheme

Under the recovery schemes, IMMUNE, IMMUNE_TX and VACCINE, nodes store and propagate anti-packets to delete obsolete packet copies to save buffer space and reduce the number of copies of each packet sent. Although anti-packets are typically much smaller than data packets, a mechanism is needed to delete anti-packets: otherwise, the buffer space taken up by anti-packets will grow arbitrarily large. In this section, we describe a global timeout scheme for deleting anti-packets.

Under the global timeout scheme, there is a global timer associated with each packet, acting upon the copies and anti-packets for the packet stored at all the nodes. Before the timer expires, the packet is propagated according to the specific forwarding scheme employed. When the timer expires, all anti-packets will be deleted; the infected nodes keep their copies of the packet, but can only forward the copy to the destination. Notice that as there is no relaying of packets after time T, nodes no longer keep anti-packets. Varying the timeout value T allows a tradeoff between delivery delay and resource consumption. This scheme is similar to spray and wait [104] in that both schemes have two phases: an epidemic style forwarding phase and a direct delivery phase. While spray and wait limits the spread of packets by specifying the maximum number of copies, our scheme limits the spread by setting a duration.

As [100] suggested, a global timer can be implemented as follows[6]. The source node sets a TTL (Time-To-Live) field to duration T for each source packet generated. The TTL field is decreased as time passes. Whenever the packet is copied to another

[6]Under the scheme they considered, when the packet timer expires, all copies and anti-packets of the packet are deleted from the network. We note that there is a non-zero probability that the packet is not delivered to the destination.

node, the new copy's TTL field is set to the remaining TTL field of the old copy; when an anti-packet is generated at the destination, its TTL field is set to the same value as the data packet being delivered.

We now demonstrate how ODEs can be used to model this global timeout scheme using the example where it is deployed for the basic epidemic routing with IMMUNE recovery. As usual, let $I(t)$ be the average number of infected nodes at time t, given that the packet has not been delivered; and $P(t)$ the CDF of delivery delay. Before time T, the packet propagates according to epidemic routing; after the timer expires, the packet can be only forwarded to the destination. Therefore, $I(t)$ and $P(t)$ satisfy the following ODEs:

$$I'(t) = \beta I(t)(N - I(t)), \text{ for } 0 \leq t \leq T \tag{2.14}$$

$$I'(t) = 0, \text{ for } t > T \tag{2.15}$$

$$P'(t) = \beta I(t)(1 - P(t)) \tag{2.16}$$

The initial conditions are $I(0) = 1, P(0) = 0$. When deriving these ODEs from the Markovian model, one has to take into account that the system has time-dependent transition rates (in particular they changes at time T). Nevertheless the same kind of convergence holds. It can be proven by applying Theorem 3.1 in [75] separately to the system trajectories before time T and after time T and then by appropriately joining them.

The above ODEs can be explicitly solved, and allow us to derive the average delivery delay and the number of copies made for a packet at delivery time (see Table 2.2 for the results).

As to the total number of copies made for a packet, we observe that after time T, the packet can only be forwarded to the destination, hence the total number of copies made for a packet (exclude the copy to the destination) is given by $G_{gt} =$

$I(T) + R(T) - I(0)$ under IMMUNE and IMMUNE_TX scheme, where $I(t)$ and $R(t)$ are solutions to Eq.(2.6,2.7) and Eq.(2.8,2.9) respectively. Under VACCINE, the total number of copies made is given by $C(T)$, where $C(t)$ is solution to Eq.(2.13) in Section 2.2.3. For the average buffer occupancy, E$[Q]$, the following equation derived in Section 2.2.3 still applies: E$[Q] = \lambda \int_0^\infty I(t)dt$.

An alternative scheme is to delete all anti-packets and copies of the packet, except the copy at the source node, when the timer expires. Compared to the previous scheme, this scheme saves buffer space but incurs larger delivery delay. Under this scheme, $I(t)$ and $P(t)$ satisfy the following ODE:

$$\begin{aligned} I'(t) &= \beta I(t)(N - I(t)), \text{ for } 0 \le t \le T \\ P'(t) &= \beta I(t)(1 - P(t)), t \le T \\ P'(t) &= \beta(1 - P(t)), t > T \end{aligned}$$

Also, we have $I(t) = 1$ for $t > T$, as all infected nodes except the source node will delete the copy when the timer expires. We derive the closed-form solution to the above ODEs, and obtained explicit formula for the average delivery delay (see the global timeout (2) in Table 2.2). The average number of copies made at delivery time, $E[C]$, and during a packet's lifetime, $E[G]$, are the same as those under the original global timeout scheme.

2.3.3 Signaling Overhead

We have so far studied the number of copies made for a packet and the average storage occupancy incurred by data packets, but ignored signaling overhead. We now discuss the signaling overhead in epidemic style routing, assuming the global timeout scheme is used to delete anti-packets.

We assume that when two nodes move into the transmission range of each other, they perform the following steps of communication:

1. exchange identification information, i.e. node ID,

2. exchange header information of data packets,

3. exchange anti-packets information,

4. exchange data packets.

The transmission cost of step (4) has already been studied in as the average total number of copies made for a packet. The amount of information exchanged in step (2) is dependent on the specific forwarding scheme. For example, under K-hop, the packet header for a packet with hop count $K-1$ does not need to be sent to other relay nodes. Likewise, the amount of information exchanged in step (3) is dependent on the specific recovery schemes. For example, while IMMUNE_TX only propagates anti-packets to infected nodes, VACCINE propagates anti-packets to all nodes.

Extending ODE models to account for signaling overhead is straightforward. Next we present the ODE model for the case where the basic epidemic routing with IMMUNE recovery is employed with the global timeout scheme with timer duration T.

Let $I(t)$ and $R(t)$ be the average number of infected and recovered nodes respectively at time t, taking into account the recovery process. We have:

$$\begin{aligned} I'(t) &= \beta I(N-I-R) - \beta I,\ t \leq T \\ I'(t) &= -\beta I,\ t > T \\ R'(t) &= \beta I,\ t \leq T \end{aligned}$$

Since all anti-packets are deleted after the timer expires, we have $R(t) = 0$ for $t > T$. The average per-node buffer occupancy of anti-packets, Q_{anti}, is given by $Q_{anti} = \lambda \int_0^T R(t)dt$, following an argument similar to that in Section 2.2.3. Here λ is the packet arrival rate.

Now let us consider the overhead of exchanging packet headers and anti-packets. Let $H(t)$ and $A(t)$ respectively denote the average number of packet headers and anti-packets that are exchanged among all the nodes up to time t, we have:

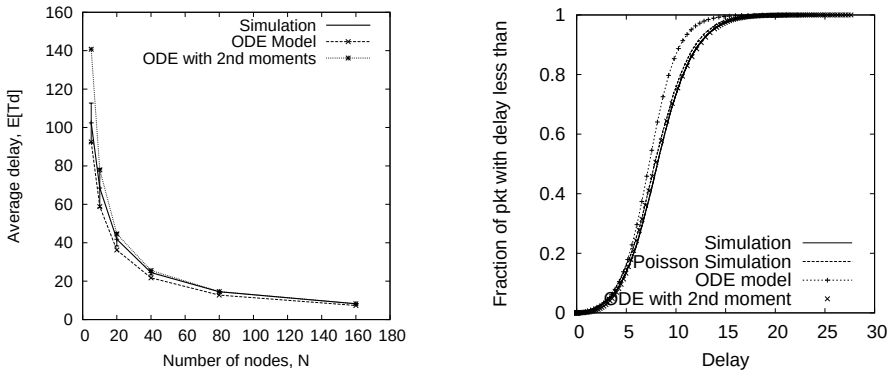
$$\begin{aligned} H'(t) &= \beta I(N+1),\ t \leq T \\ H'(t) &= \beta I,\ t > T \\ A'(t) &= \beta I \end{aligned}$$

Intuitively, before time T, the packet header is sent by every infected node to every node they meet[7]. After the timer expires, the packet header is sent only to the destination node. Anti-packets are transmitted by the destination node to infected nodes when they meet under IMMUNE, before or after T. For any packet, the average total number of times the packet header and anti-packet is exchanged is given by $G_h = H(\infty)$ and $G_a = A(\infty)$ respectively. Numerical techniques can be used to evaluate these metrics.

2.4 Model Validation

We have developed a simulator that simulates various routing schemes and recovery schemes under random waypoint and random direction models. The results we present here are for a specific setting considered in [39]: $N+1$ nodes move within a 20×20 terrain according to the random direction model [13, 42]. Each node chooses an initial direction, speed and travel time, and then travels in that direction with a given speed for the chosen travel time. When the travel time expires, the node chooses a new direction, speed, and travel time at random, independently of all previous directions, speeds and travel times. If a node hits the boundary of the terrain,

[7]For the purpose of clarity, we ignore some optimizations that can be used to save overhead. For example, when two infected nodes for packet i meet, after one node sends its packet headers, the other node, knowing the previous node has packet i, need not send packet i header to B.



(a) Average delay for different N (b) CDF of delay with $N = 160$

Figure 2.1. Delay under epidemic routing

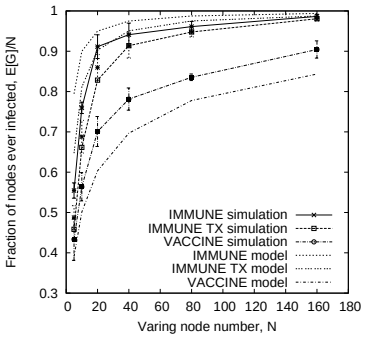
it wraps around at the other side of the terrain. The node speed is chosen uniformly in the range $[4, 10]$, and the mean travel duration is $1/4$. The transmission range of the nodes is chosen to be 0.1. The pair-wise meeting rate for this setting is found to be $\beta = 0.00435$ using the formula in [39].

We simulate $N+1$ unicast flows, with each node being the source of one flow, and the destination of another flow. Each flow generates packets according to a Poisson process with rate $\lambda = 0.01$. The simulation is run long enough such that at least 500 packets are generated and delivered. We then use the 500 observations to calculate the mean and 95% confidence interval for average packet delivery delay and the average total number of copies made for a packet. Average buffer occupancy is calculated after removing the initial transient period from the trace. These simulation results are then compared with the ODE models predictions. We report the *relative modeling error*, defined as $(V_s - V_m)/V_m$, where V_m is the model predicted value and V_s is the simulation result. We calculate the 95% confidence interval for the relative modeling errors using the 95% confidence interval for V_s. We do not consider signaling overhead as we expect similar model prediction performance for these metrics.

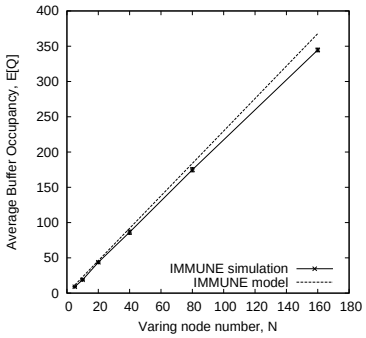
We first consider basic epidemic routing. We vary N between 5 and 160, and plot the mean and 95% confidence interval of packet delivery delay obtained from

simulation, and the model predicted average delay in Figure 2.1(a). We find that the model accurately predicts the average delivery delay, capturing the performance trend as N increases. Figure 2.1(b) compares, for $N = 160$, the CDF of packet delivery delay obtained from simulation with the one predicted by Eq.(2.4). It shows that ODE model under predicts the packet delivery delay. To investigate modeling error, we run another simulation with nodes meeting according to a Poisson process with rate $\beta = 0.00435$ (i.e., we set the meeting rate in the simulation to exactly match the model's meeting rate) and the results of the two simulations are very close (see the curve labeled as "Poisson Simulation" in Figure 2.1(b)). This suggests that the error introduced by the Poisson meeting process approximation is negligible. We conjecture that the prediction errors are mainly due to the small number of initially infected nodes and/or the small total number of nodes. This is confirmed by simulations where we vary the number of initially infected nodes, and observe that the modeling error becomes smaller when the number of initially infected nodes is large. We also use a moment-closure technique to derive an ODE system involving second moments using the MVN method (details are given in Appendix A). The modified ODE provided a better prediction for mean and CDF of packet delivery delay (Figure 2.1).

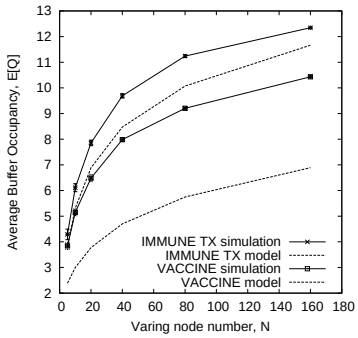
For epidemic routing with different recovery schemes, Figure 2.2 plots $\mathrm{E}[G_{ep}]/N$, and the average buffer occupancy $\mathrm{E}[Q]$ as predicted by the model and obtained from simulation. We find that the ODE models are more accurate for IMMUNE than for VACCINE. In some sense, any error in the infection process modeling is amplified by the exponentially fast recovery of VACCINE. We observe that IMMUNE_TX only slightly reduces the number of copies sent for each packet, while VACCINE further reduces the number of copies sent. The reduction in buffer requirements is similar for IMMUNE_TX and VACCINE.



(a) $\mathrm{E}[G_{ep}]/N$

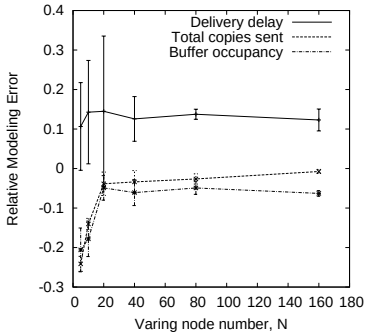


(b) Buffer occupancy under IMMUNE

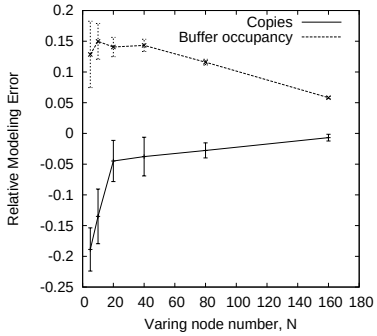


(c) Buffer occupancy under IMMUNX_TX and VACCINE

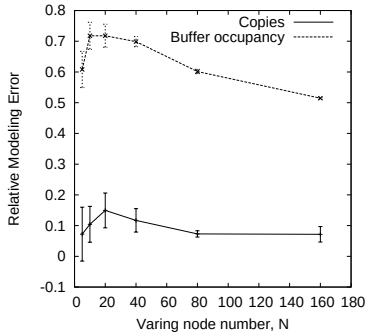
Figure 2.2. Copies sent and buffer occupancy under epidemic routing



(a) IMMUNE recovery



(b) IMMUNE_TX recovery

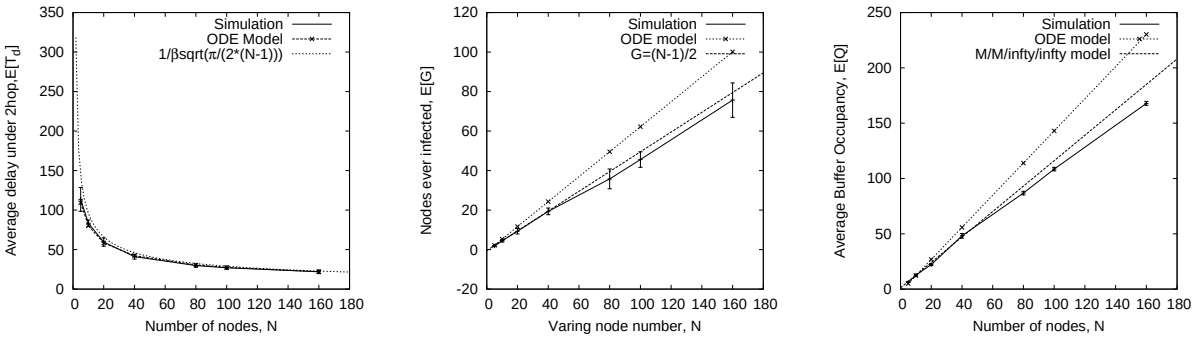


(c) VACCINE recovery

Figure 2.3. Relative modeling errors for epidemic routing with different recovery schemes

Figure 2.3 plots the relative modeling errors for delivery delay, number of copies made and buffer occupancy, for epidemic routing under the three recovery schemes. We observe that as N increases, the error decreases. While ODE models over predict the copies sent and average buffer occupancy for IMMUNE recovery, they under predict buffer occupancy for IMMUNE_TX recovery, and under predict both metrics for VACCINE recovery.

Next, we present validation results for the forwarding schemes introduced in Section 2.3, focusing on the following three metrics, average delay ($\mathrm{E}[T_d]$), average buffer occupancy ($\mathrm{E}[Q]$), and average total number of copies transmitted ($\mathrm{E}[G]$) under



(a) Delivery Average (b) Copies transmitted (c) Buffer occupancy

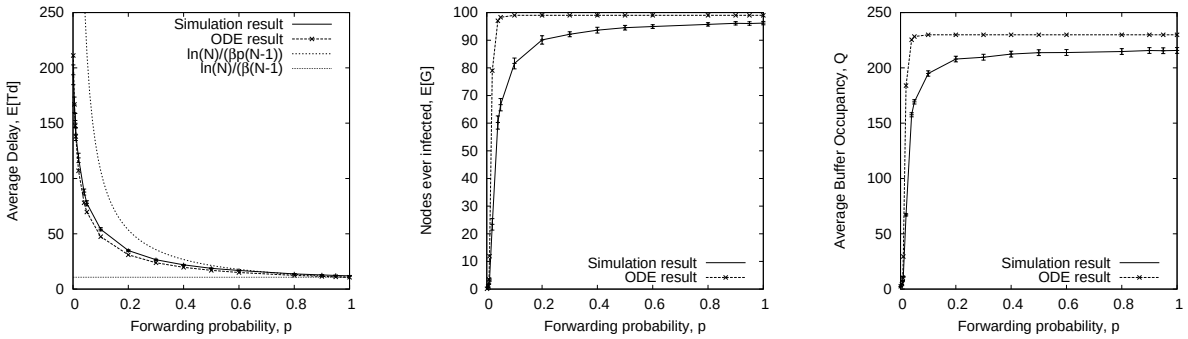
Figure 2.4. Average delivery delay, copies transmitted and buffer occupancy under 2-hop forwarding

IMMUNE recovery. We expect the prediction errors to be slightly larger for IMMUNE_TX and VACCINE recovery as observed for epidemic routing.

For the 2-hop forwarding, Figure 2.4 compares the three metrics as the number of nodes N is varied, showing a good match between the modeling results and simulation results. Figure 2.6(a) plots the relative prediction error.

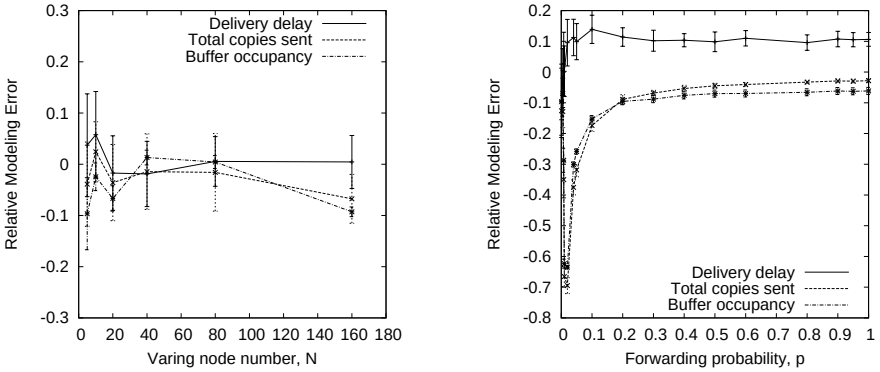
For probabilistic forwarding scheme, Figure 2.5 plots the three metrics, comparing the model prediction with simulation results for $N = 100$ under varying forwarding probability. Figure 2.6(b) plots the relative prediction error for probabilistic forwarding. We observe a larger prediction error for $p \in [0.01, 0.1]$, and the error decreases as p increases and approaches to 1. We conjecture the large prediction error in $p \in [0.01, 0.1]$ is due to the larger variance when p takes a value in this range (as shown in Appendix A). Like epidemic routing, the ODE models underpredicts the average delay, whereas overpredicts the average number of copies sent and the average buffer occupancy.

For limited time forwarding (with no reinfection after timeout) for $N = 100$ under varying average timeout value, $1/\mu$, Figure 2.7 plots the three metrics as predicted by the model and as obtained through simulation, and Figure 2.9.(a) plots the relative modeling error. We observe that the relative modeling error decreases as the average



(a) Delivery delay (b) Copies transmitted (c) Buffer occupancy

Figure 2.5. Average delivery delay, copies transmitted and buffer occupancy under probabilistic forwarding, $N = 100$

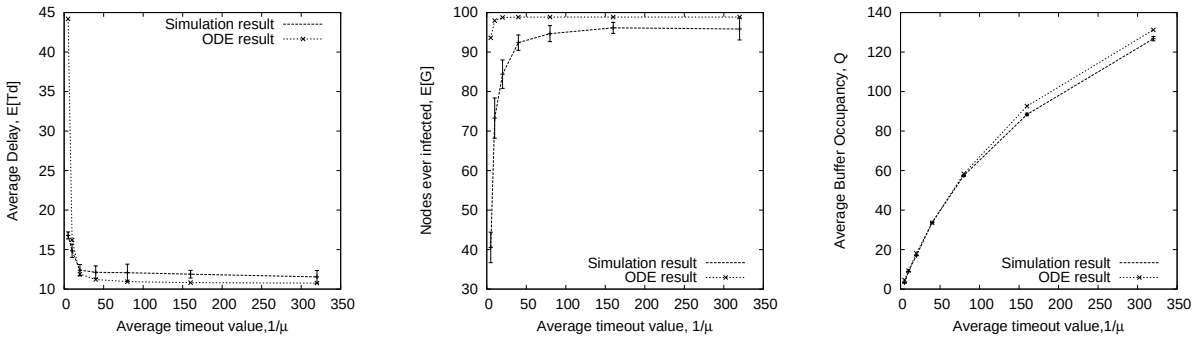


(a) 2-hop forwarding (b) Probabilistic forwarding, $N = 100$

Figure 2.6. Relative modeling error for 2-hop and probabilistic forwarding

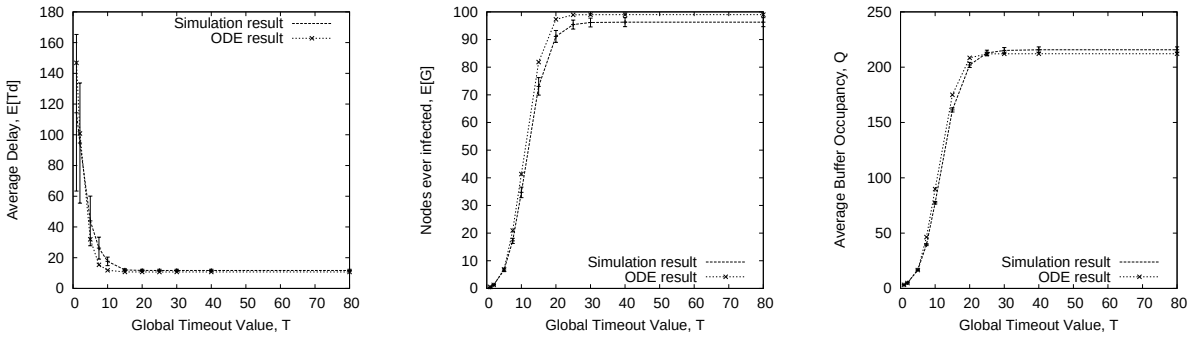
timeout value increases. This is expected because the higher the number of infected nodes, the better is the fluid approximation. As in the case of epidemic routing, the model underpredicts the delay, and overpredicts the number of copies sent and the average buffer occupancy.

Finally, for epidemic routing with IMMUNE recovery and global timeout mechanism, Figure 2.8 plots the three metrics under $N = 100$ as the global timeout value, T is varied. Figure 2.9(b) plots the corresponding relative prediction errors. We observe that the ODE model underpredicts the average delay, and overpredicts the average number of copies sent and the average buffer occupancy, as the case for epidemic



(a) Delivery delay (b) Copies transmitted (c) Buffer occupancy

Figure 2.7. Average delivery delay, copies transmitted and buffer occupancy under limited-time forwarding, $N = 100$



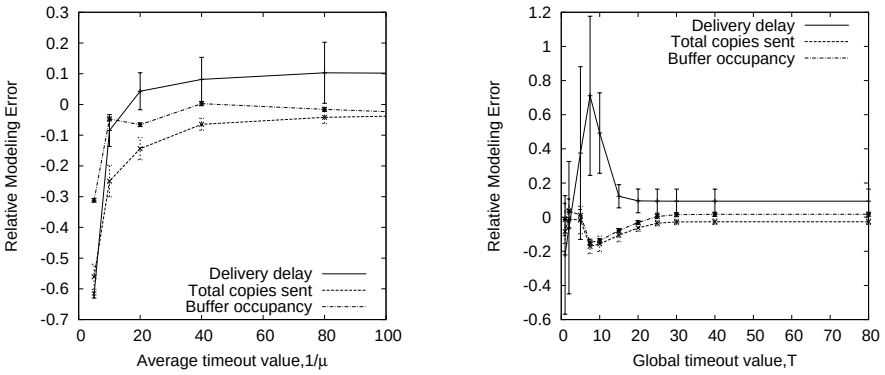
(a) Delivery delay (b) Copies transmitted (c) Buffer Occupancy

Figure 2.8. Average delivery delay, copies transmitted and buffer occupancy under global timeout scheme, $N = 100$

routing. The relative modeling error decreases as the timeout value T increases, as is the case for limited time forwarding.

2.5 Performance Trade-offs

In this section, we show how the ODE models we derived can be employed to quantitatively explore tradeoffs between delivery delays and resource consumption under different forwarding and recovery schemes, and to determine configuration criteria. It is not our intent to exhaustively explore all the possible dimensions of epidemic routing (forwarding schemes, recovery schemes, methods to manage anti-packets) in



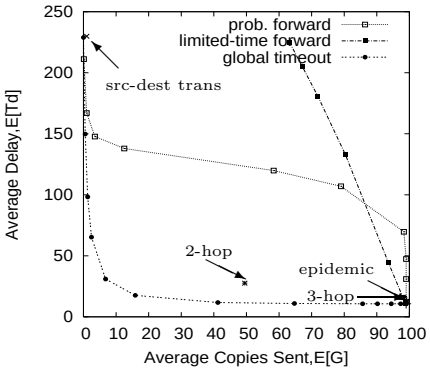
(a) Limited-time forwarding (b) Global timeout scheme

Figure 2.9. Relative modeling errors for limited-time and global timeout scheme, $N = 100$

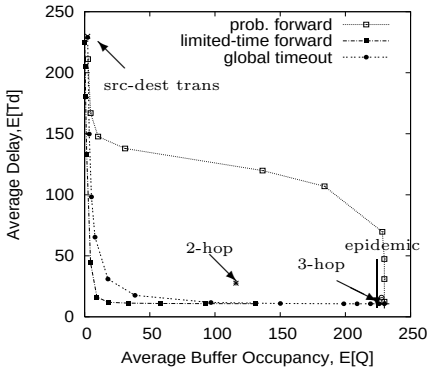
order to determine the best candidate under a specific scenario and a performance optimization goal.

We discuss the average delay versus average number of copies transmitted and average delay versus average buffer occupancy tradeoff achieved by different forwarding schemes under IMMUNE (Section 2.5.1) and VACCINE (Section 2.5.2). We ignore signaling overhead in this discussion, because doing so requires that we exactly account for the deletion of anti-packets, for example, by using the global timeout scheme. For each forwarding and recovery scheme and the particular parameter setting, choosing a different global timer T results in a different tradeoff between delivery delay and resource consumptions. Furthermore there is no optimal choice of T unless an optimization goal is given. The latter optimization consideration is beyond the scope of this chapter. The reader interested in this topic can refer to our work [90].

The results presented in this section are mainly based on numerical solution of the ODEs models previous proposed (for $N = 100$, $\beta = 0.00435$, $\lambda = 0.01$), but we also employ asymptotic results for qualitative considerations.



(a) delay vs number of copies sent tradeoff



(a) delay vs buffer occupancy tradeoff

Figure 2.10. Performance tradeoffs for various schemes with IMMUNE recovery

Probability (p,%)	0.1	0.5	0.8	1	1.5	2	5	10	20	80			
Timeout ($1/\mu$)	0.1	0.5	1	2	5	10	20	40	80	160	320		
Global timer (T)	0.01	1	2	3	5	7	10	12	15	16	18	20	80

Table 2.3. Settings considered for probabilistic, limited-time forwarding, and global timeout scheme

2.5.1 Performance Trade-off Under IMMUNE

Figure 2.10(a) and (b) respectively plot the delay-versus-number-of-copies-sent and the delay-versus-buffer-occupancy trade-offs achieved by different forwarding schemes when IMMUNE recovery is employed. In the figure, there are four singleton points corresponding to direct source-destination transmission, 2-hop, 3-hop forwarding, and epidemic routing. Three curves have been obtained for probabilistic forwarding, limited-time forwarding (without reinfection), and global timeout scheme respectively; for these curves, each point corresponds to a different value of the forward probability p, the mean timeout interval $1/\mu$ or the global timeout T respectively. All these parameter values are shown in Table 2.5.1.

Let us first consider the delay-versus-number-of-copies-sent trade-off. One can reduce the average number of copies sent by decreasing p, $1/\mu$, or T, but with an increase in average delay. $p \to 1$, $1/\mu \to \infty$, and $T \to \infty$, whereas $p \to 0$, $1/\mu \to 0$, or $T \to 0$ correspond to a no-relaying scenario where the packet is only delivered

directly from the source to the destination. The only difference is for $1/\mu \to 0$, the number of copies converges to $N/2$ (which is the average number of nodes the source uselessly infects before meeting the destination). A global timeout scheme appears to be the best choice when limiting the average number of copies transmitted is the main concern. As a rule of thumb, one can choose $T \approx \mathrm{E}[T_d]/2$ (= 5.0 in this specific setting), where $\mathrm{E}[T_d]$ is the average delay under epidemic routing. In this setting, this choice significantly reduces the number of copies sent from nearly 100 to 6.8, with the average delivery delay increased from around 10 to around 30.

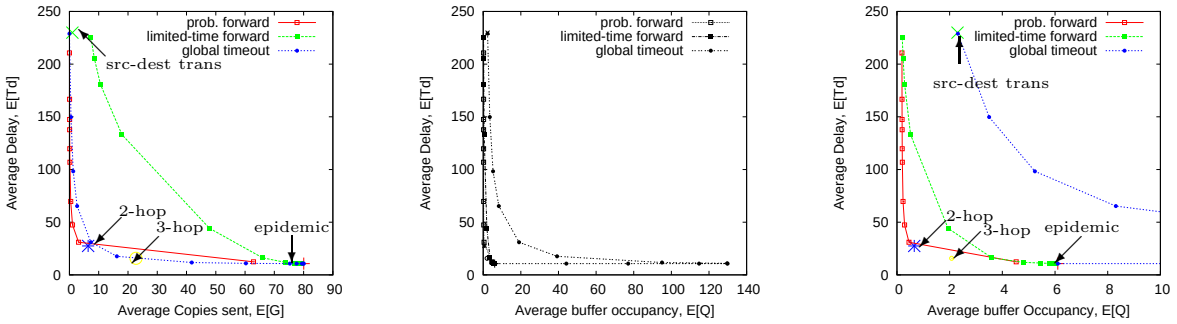
Figure 2.10(b) shows that for probabilistic and K-hop forwarding, the delay-versus-buffer-occupancy tradeoffs are similar to the delay-versus-copies tradeoffs. This is due to the proportionality between the number of copies sent and the buffer occupancy that we have shown in Section 2.2.3 for epidemic routing under IMMUNE. This relation holds for all schemes where copies are deleted only after the infected node's meeting with the destination, hence also for probabilistic and K-hop forwarding under IMMUNE, but not for limited-time or global timeout forwarding. We observe that limited time forwarding is the best choice when limiting buffer occupancy is of primary concern. With a value of $1/\mu \approx 2\mathrm{E}[T_d]$ (=20 in this specific setting), the average buffer occupancy is decreased to about one tenth of that of epidemic routing, with a small increase in the average delivery delay. The delay-versus-buffer tradeoff achieved by the global timeout scheme is very close to that of limited time forwarding.

2.5.2 Performance Improvement by VACCINE

Figure 2.11 shows the delay-versus-copies and delay-versus-buffer-occupancy tradeoffs under various forwarding schemes when VACCINE recovery is employed. For the delay-versus-copies tradeoff (Figure 2.11(a)), compared to IMMUNE recovery, VACCINE recovery decreases the average number of copies sent for a packet and the average buffer occupancy for each forwarding scheme. However, for different schemes,

different amounts of improvements are achieved by VACCINE recovery: in particular, the largest improvement is achieved for probabilistic forwarding, followed by K-hop forwarding, and then limited-time forwarding and global timeout scheme. The relatively small improvements for limited-time forwarding and global timeout scheme are due to their intrinsic recovery features: nodes delete packet copies when the timer expires and they cannot be reinfected. The explanation is more complex for the probabilistic and K-hop forwarding schemes. Because of the two counteracting processes – the counter-infection recovery process due to anti-packets spreading and the continuing packet infection – the total recovery speed depends not only on the recovery scheme but also on the specific infection process. Given the same average delivery delay, when the recovery process starts, the average number of nodes infected and the infection rates are higher under probabilistic forwarding (its infection rate is exponential, hence in the long term it is faster than K-hop). For this reason, we expect the IMMUNE recovery process to be significantly "longer" for probabilistic forwarding than for K-hop forwarding, leading to larger buffer occupancies and more copies transmitted for a packet (as shown in Figure 2.10). Conversely under VACCINE, the recovery process is much shorter; the buffer occupancy is mainly determined by the initial infection process (before the delivery), and the difference in the copies transmitted and the buffer occupancy under probabilistic forwarding and K-hop scheme becomes much smaller, as shown in Figure 2.11.

Figure 2.11(b) and Figure 2.11(c) illustrate the delay-versus-buffer-occupancy tradeoff for various forwarding schemes under VACCINE recovery, where Figure 2.11(c) focuses on the small buffer occupancy range. Comparing Figure 2.11(c) to Figure 2.11(a), we find that the delay-versus-buffer tradeoff is similar to the delay-versus-copies tradeoff for all schemes except the global timeout scheme. For the global timeout scheme, as Figure 2.11.(b) shows, as T increases, the average delay decreases monotonically; whereas, the buffer occupancy increases first and then de-



(a) delay vs number of copies sent tradeoff

(b) delay vs buffer occupancy tradeoff

(c) delay vs buffer occupancy tradeoff

Figure 2.11. Performance tradeoffs for various schemes with VACCINE recovery

creases. To see why this is the case, Figure 2.12 plots the numerical solutions for $I(t)$ in ODEs given by Eq.(2.14) and Eq.(2.15), the number of infected nodes at time t, for several values of T. Basically, under the global timeout scheme, the recovery process after time T is essentially the IMMUNE recovery, which is much slower than VACCINE recovery. As a result, increasing the timeout value T not only leads to a longer epidemic spread phase, but also results in a faster overall recovery process. When T is smaller than a certain threshold (which is around 15 for the specific setting considered here), the former effect outweighs the latter one, leading to a larger buffer occupancy (as illustrated by $T = 2$ and $T = 10$ curves); when T increases further, the latter effect becomes the dominant factor, leading to a small average buffer occupancy (as illustrated by the $T = 20, 40$ curves)[8].

2.6 Epidemic Routing under Buffer Constraints

Thus far, we have assumed that each node has sufficient buffer space to store all packets. In reality, however, mobile nodes often have limited storage due to cost and form factor. Sizing the buffer to limit end-to-end packet losses due to buffer

[8]Recall that $Q = \lambda \int_0^\infty I(t)dt$, as derived in Section 2.2.3.

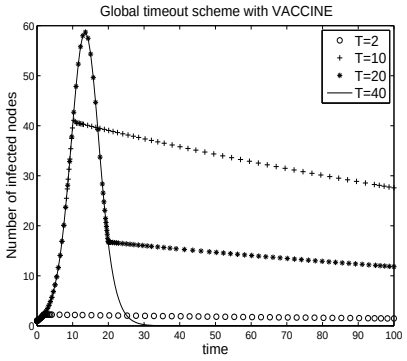


Figure 2.12. $I(t)$ for different T under global timeout scheme

overflow in store-carry-forward networks is difficult. For example, [43] studied buffer occupancy variability for the purpose of buffer sizing, but their model requires an empirical distribution obtained from simulation.

In this section, we examine the performance of epidemic routing under the constraint that each node can store at most B packets, and compare the performances of three buffer management strategies: *droptail*, *drophead* and *drophead_sp* (i.e., source prioritized drophead) under IMMUNE recovery. From the modeling point of view, we couple the forwarding models with Markovian or fluid queuing models. The coupling involves some common parameters, so that in general a fixed point problem has to be solved.

2.6.1 Droptail Scheme

Under *droptail* scheme, when the buffer of a node is full, the node will not accept any packet. Let P_d denote the probability that the buffer of a node is full. As packets are accepted by relay nodes with probability $1 - P_d$, the forwarding process is exactly the same as probabilistic forwarding with forwarding probability $1 - P_d$. The model proposed in Section 2.3.1.2 is hence directly applicable. Furthermore, with probability P_d, a source packet generated at a node finds a full buffer and is discarded, leading to a loss probability of P_d.

In order to estimate P_d, we model the buffer at a node as an $M/M/B/B$ queue, where B denotes the buffer size. Under IMMUNE recovery, a packet copy is deleted only when meeting the destination, therefore the service rate is β, the rate that the node meets the destination. The arrival rate is the sum of the source packet rate, λ, and the relay packet rate. The latter rate is given by $(N-1)\lambda(1-P_d)$, as there are $N-1$ relay flows each of rate λ, with each packet being lost at the source with probability P_d. We find P_d by solving a fixed point problem: given an arrival rate of $\lambda(1+(N-1)(1-P_d))$, and a service rate of β, the loss probability P_d of the $M/M/B/B$ queue can be calculated using the Erlang's B formula [4]: $P_d = \frac{\frac{A^B}{B!}}{\sum_{i=0}^{B}\frac{A^i}{i!}}$, where $A = \frac{\lambda(1+(N-1)(1-P_d)}{\beta}$ is the traffic offered in Erlangs.

2.6.2 Drophead Scheme

Under *drophead* scheme, when a node receives a new packet (source or relay) and its buffer is full, it discards the oldest packet in its buffer; and the node does not accept a copy of the discarded packet in the future.

Let G_{dh} be the number of times a packet is copied in the system, and $\overline{G}_{dh}$ its expected value. As the average arrival rate to a node is given by $(\overline{G}_{dh}+1)\lambda$, the packets in the buffer are pushed to the head of the buffer with this rate, in other words, the packets in the buffer *age* with this rate. As before, let's consider the spread process of a packet. Let $S(t)$ denote the average number of susceptible nodes at time t, $I_i(t)$ the average number of infected nodes where the copy of the packet is the i-th newest packet, $D(t)$ the average number of nodes that have deleted the packet (and would not accept the packet in the future) at time t. The following ODEs can be used to model the forwarding process.

$$\begin{aligned} S'(t) &= -\beta S \sum_{1\le i\le B} I_i \\ I_1'(t) &= \beta S \sum_{1\le i\le B} I_i - (\overline{G}_{dh}+1)\lambda I_1 \end{aligned}$$

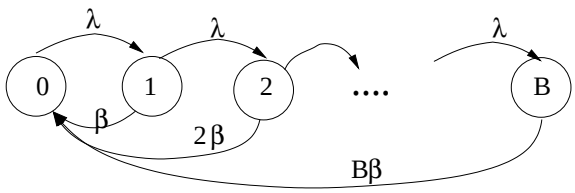


Figure 2.13. Markov chain for the number of source packets in a node's buffer under drophead_sp scheme. Source packets arrive to a node with rate λ. When the node encounters the destination of these packets (with rate β), all source packets in the buffer are delivered and deleted from the buffer.

$$
\begin{aligned}
I_j'(t) &= (\overline{G}_{dh}+1)\lambda(I_{j-1}-I_j), 2 \le j \le B \\
D'(t) &= (\overline{G}_{dh}+1)\lambda I_B \\
P'(t) &= \beta \sum_{1 \le i \le B} I_i(1-P)
\end{aligned}
$$

The initial conditions are: $S(0) = N-1, I_1(0) = 1, I_j(0) = 0$, for $j = 2, ...B$, $D(0) = 0, P(0) = 0$.

Note that if all copies of a packet are discarded before the packet's delivery to the destination, the packet is lost. We estimate the loss probability as $\lim_{t\to\infty} P(t)$.

We find $\overline{G}_{dh}$ by solving a fixed point problem: given $\overline{G}_{dh}$, we numerically solve the extended ODE model (obtained by adding recovery process to the above ODEs) and calculate the amount of flow from state S to I_1, i.e. $S(0) - S(\infty) = \overline{G}_{dh}$. We perform a binary search to find the fixed point $\overline{G}_{dh}$.

2.6.3 Source Prioritized Drophead Scheme

Source prioritized drophead scheme, in short, *drophead_sp*, is similar to *drophead* scheme, with the difference to give higher priority to source packets. Under this scheme, if a source packet arrives to a node with a full buffer, the node will first try to drop oldest relay packets, and then the oldest source packets. If a relay packet arrives to a full buffer, the node finds the oldest relay packet and delete it from the buffer; if all packets in the buffer are source packets, the relay packet is not accepted.

Let P_f be the probability that a node's buffer is filled with its own source packets. We use the Markov chain as shown in Figure 2.13 to model the number of source packets in a node's buffer. We then calculate the stationary distribution p_i (probability that there are i source packets in the buffer) of the Markov chain, and get:

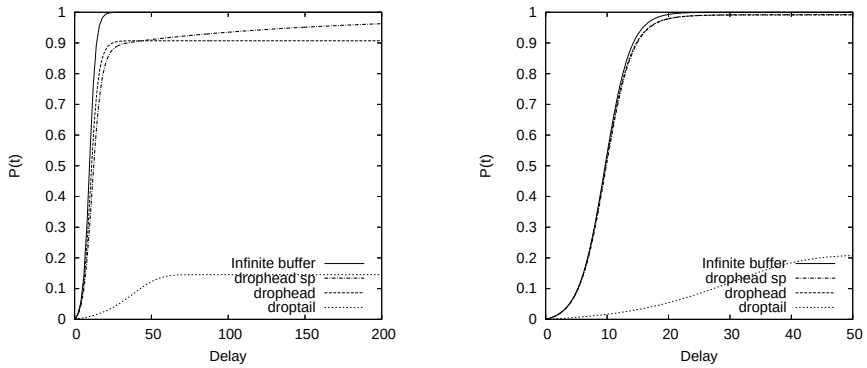
$$P_f = p_B = \frac{\frac{\lambda}{\beta}(\frac{\lambda}{\lambda+\beta})^{B-1}}{1+\frac{\lambda}{\beta}-(\frac{\lambda}{\lambda+\beta})^{B-1}}.$$

Given P_f, a *relay* packet is accepted with probability $1-P_f$, leading to an effective infection rate of $\beta(1-P_f)$. The following ODE can be used to study the delivery delay; here $I_j^s(t)$ denotes the probability that the source node's copy of the packet is the j-th newest source packet in the buffer.

$$\begin{aligned}
S'(t) &= -\beta(1-P_f)S\sum_{1\le i\le B}(I_i^s+I_i)\\
I_1'(t) &= \beta(1-P_f)S\sum_{1\le i\le B}(I_i^s+I_i)-(\overline{G}_{dhs}+1)\lambda I_1\\
I_j'(t) &= (\overline{G}_{dhs}+1)\lambda(I_{j-1}-I_j),\quad 2\le j\le B\\
I_1^{s\prime}(t) &= -\lambda I_1^s\\
I_j^{s\prime}(t) &= \lambda(I_{j-1}^s-I_j^s),\quad 2\le j\le B\\
D'(t) &= (\overline{G}_{dhs}+1)\lambda I_B+\lambda I_B^s\\
P'(t) &= \beta\sum_{1\le i\le B}(I_i^s+I_i)(1-P)
\end{aligned}$$

The initial conditions are given by: $S(0)=N-1$, $I_j(0)=0$, for $j=1,...,K$, $I_1^s(0)=1$, $I_i^s(0)=0$, for $i=2,...B$, $D(0)=0$, $P(0)=0$.

Similar to the *drophead* scheme, we find $\overline{G}_{dhs}$ by solving a fixed point problem using the extended ODE model (taking into consideration the recovery process).



(a) $B = 5$ (b) $B = 10$

Figure 2.14. $P(t)$ under $B = 5, 10$

2.6.4 Comparisons of Different Schemes

We have simulated these buffer management schemes, using the same setting as before ($N = 100, \lambda = 0.01$, $\beta = 0.00435$), with different buffer sizes $B = 5, 10, 20$. We compare the results obtained from these simulations to the prediction from our ODE models. Table 2.6.4 tabulates the packet loss probability, i.e. the probability that all copies of a packet are dropped before the destination receives one. Figures 2.14 plot the delay distributions predicted for $B = 5, 10$, in the range $[0, 200]$ and $[0, 50]$ of the X axis respectively so that the difference between schemes can be seen.

We observe that the models provide reasonably accurate loss probability predictions that reflect the relative performance of the three schemes. The shape of the distribution probability function for delivery delay is also well-captured by the model. We observe that naive *droptail* performs poorly. *Drophead* provides fast infection, as relay packets are always accepted; however, significant packet losses are incurred for $B \leq 10$. With *drophead_sp*, although the infection spreads slower than under *drophead*, more packets are delivered. If the packet rate is so high that the buffer can only hold its own source packets, *drophead_sp* degenerates to direct source-destination transmission. Note that with infinite buffers, the average buffer occupancy for this setting is over 200 (Figure 2.2(b)). Our results here suggest that similar performance

can be achieved by *drophead* and *drophead_sp* with a much smaller buffer size, equal to only 20 packets.

2.7 Related Work

In the mathematical epidemiology field, there exists a vast literature about mathematical models on the spreading of infectious diseases, including both stochastic and deterministic models [11, 27]. These mathematical techniques have been applied to various computer networking problems that exhibit a strong analogy to epidemic spreading of disease. For example, [70, 106, 88, 120] modeled the spread of computer viruses and worms in computer networks by adapting epidemiological models. Moreover, a number of network applications and protocols have adopted epidemic-style communication for data dissemination and resource discovery, and therefore epidemic models are natural ways to study their performance. These include epidemic algorithms [30] for maintaining consistency of replicated database, gossip or rumor-based protocol [65], broadcast communication [71, 34] and peer-to-peer data sharing [91] in mobile ad hoc network, and more recently, epidemic routing [110, 43, 117] in Delay Tolerant Networks. Epidemic routing differs from the other above mentioned broadcast based protocols in that it supports *unicast applications*, using epidemic style flooding to decrease the delivery delay.

Buffer size	simulation/model	droptail	drophead	drophead_sp
5	simulation	0.9696	0.2234	0.0536
	model	0.8544	0.0928	0.0079
10	simulation	0.9471	0.0315	0.0
	model	0.7891	0.0088	0.0
20	simulation	0.899	0.0016	0.0
	model	0.7011	0.0	0.0

Table 2.4. Loss probability under buffer constraints

Based on earlier results in [38], we have used a homogeneous mixing model employing a single parameter (derived from mobility parameters) to capture the contact rate between mobile nodes. [71, 88] considered similar network settings as our work, i.e., mobile ad hoc network. [71] made a similar homogeneous mixing assumption, and obtained the contact rate through finding best-fitting formula from results of many simulation runs. [88] considered a network with higher node density and slower nodal mobility than our paper, and extended Kephart-Whilte model [69] to model the virus spreading, characterizing the fraction of nodes with varying connectivities under given mobility models.

Another important difference between our work and the above mentioned work lies in the fact that we are interested in performance metrics that are unique to the unicast application in DTN. We have seen that there exists a tradeoffs between the delivery delay and resource consumptions in terms of the number of transmissions made for a packet and buffer occupancy. Using ODE models, we have studied the performance of various epidemic style routing, and explored the tradeoffs they achieve.

The work most closely related to ours is the work by Haas and Small [43], where an ODE model is applied to study delay under epidemic routing, and Markov chain models are used to study the storage requirement under different recovery schemes. While both our work and [43] study the delay, storage requirement, and transmission numbers of epidemic routing, our work goes beyond this single scheme to study schemes such as 2-hop forwarding, probabilistic forwarding, limited-time forwarding, global timeout scheme, and epidemic routing in the buffer-constrained scenario. In addition, our analysis leads to new closed-form expressions and asymptotic results, when the number of nodes increases, for a number of schemes. Furthermore, we study epidemic routing under buffer-constrained scenario using ODE models coupled with Markov models to compare different buffer management strategies. We also note that the approach in [43] is a hybrid approach and requires obtaining some model param-

eters, such as the number of nodes infected at the time of delivery, from simulations. We derive all metrics as part of the model itself. Last, because our focus is on the use of ODE models, we provide insight into when they do or do not work and why, and show how moment closure techniques could be employed to improve the model predictions.

Another closely related work is [38], where based on the result of Poisson meeting process, Groenevelt *et al.* modeled 2-hop forwarding and epidemic routing using Markov chain models. Average delay and the number of copies generated at the time of delivery for these two schemes was derived from the models. Using ODE models, we have more easily derived similar results. [44] later extended this work to consider a variant of 2-hop scheme with exponential timers at each node, and with a limit on the maximum number of copies. Through Markovian analysis, the author derived closed-form formulas and numerical solutions for delivery delay, number of copies transmitted for these two schemes respectively. Given the difficulty in deriving asymptotic formulas from Markovian analysis, ODE models were employed to derive asymptotic formulas for moments of delivery delay, and copies made at delivery time.

2.8 Summary

In this chapter, we proposed and investigated a unified framework based on ODEs to study the performance of various forwarding and recovery schemes. We derived ODE models as limit processes of Markovian models under a natural scaling as the number of nodes increases, and employed the ODE models to obtain a rich set of closed-form formulas regarding the packet-delivery delay, number of copies sent, and buffer occupancy under various schemes. We validated the models through simulations, and observed a good match between the model prediction and simulation results. We used the ODE models to explore performance tradeoffs achieved by various schemes, and obtained insights into the different schemes. We further considered

the buffer-constrained case, and showed that with an appropriate buffer management scheme, a much smaller buffer can be used with negligible effect on delivery performance.

CHAPTER 3

ON THE BENEFIT OF NETWORK CODING FOR UNICAST APPLICATION IN DTN

3.1 Introduction

Network coding is a new field in information theory started by the seminal work by Ahlswede *et al.* [8] in 2000. Network coding refers to the new concept where the network nodes (e.g., routers, switchs) combine/mix previously received packets before forwarding them, rather than simply forwarding data received. [8] showed that network coding allows for higher network throughput for a single multicast flow case. Since then, network coding has found many intersting applications (see the short primer [36] which gave a nice review on previous works on network coding).

Random Linear Coding (RLC) is a form of network coding that was first proposed by Ho *et al.* [46]. Basically, under RLC, each network node forwards random linear combinations of the data it has received. Previous works have applied RLC to networking scenarios including P2P content distribution [37], multicast applications [25], gossip protocols [29] and distributed storage [28, 7].

In this chapter, we investigate the benefit of applying Random Linear Coding (RLC) to unicast applications in DTNs with opportunistic contacts and resource constraints. To our knowledge, the only previous work applying network coding to a DTN setting is [112] by Widmer and Boudec. There, the authors consider *broadcast* data delivery using RLC; our focus here is on using RLC for unicast delivery.

For unicast applications, there are different possible ways to combine packets: each node can combine all the packets in its buffer, or only the packets destined

to the same destination, or only the packets belonging to the same flow (i.e., same source-destination pair). We first consider these three possibilities in the simple case where a single block of K packets propagate in the network.

The performance metric of interest is the delay until the last packet in a block is delivered, but we will also comment on the average packet delay for a block. We show that for the single block case, when bandwidth is constrained, applying RLC over packets destined to the same node achieves the minimum delay with high probability. We find that this benefit increases further when buffer space within DTN nodes is limited. We also demonstrate that the "price" to be paid for the improved delay performance is a larger number of epidemically-spread copies of data in the network. However, when a token-based scheme is used to limit the number of transmissions made, the RLC based scheme yields a smaller average delivery delay under similar transmission overhead as non-coded schemes.

We then consider the scenario where there are multple source/destination pairs with blocks of K packets arriving according to a Poisson bulk arrival process at each source. We find that the RLC scheme achieves slightly smaller average block delay than non-coded schemes when only bandwidth is constrained, but shows more significant benefits when both bandwidth and buffers are constrained.

The remainder of this chapter is structured as follows. We introduce the network model, the forwarding and recovery schemes, and the simulation setting in Section 3.2. Section 3.3 studies the benefit of the RLC scheme over non-coded schemes for the single generation case. Section 3.4 extends the study to multiple generation case. Section 3.5 reviews related work. Finally, Section 3.6 summarizes this chapter and discusses future works.

3.2 Network Model, Forwarding and Recovery Schemes

In this section, we first introduce the network model, and then describe the forwarding and recovery schemes we study in this work, and finally describe the simulation setting. Table 3.1 summarizes the notations used throughout this chapter.

3.2.1 Network Model

We consider unicast communications (i.e. each message is destined for a single node) in a network consisting of N nodes moving according to a mobility model (discussed shortly) within a closed region. Each node has a limited transmission range, such that the network is sparse and disconnected.

We employ the *temporal network* model proposed by Kempe *et al.* in [68] to represent the dynamic network topology formed by the mobile nodes. Basically, a temporal network is an directed graph $G = (V, E)$ in which each edge e is annotated with a time label $\lambda(e)$ specifying the time at which its two endpoints "communicated". We extend this model such that each link also has a capacity attribute and assmume that one packet can be exchanged over each link. We construct the temporal network as follows (Figure 3.1): there are N vertices, each corresponding to a mobile node. For each contact between a pair of nodes that can exchange b packets in each direction, b directed edges are added in each direction between the corresponding vertices. Edges are labeled with the times that the contacts occur. A *time-respecting path* in the network is a path in the network where the successive edges have increasing timestamps. For example, there are three time-respecting paths from node 1 to node 4, i.e., two paths that goes through node 2 and one path that goes via node 3. There is no time-respecting path from node 4 to node 1. A set of paths are *independent* if they do not share edges. In this example, the two paths from node 1 to 4 going through node 2 are not independent, as they share the edge $(2, 4)$. Pathes $(1- > 2- > 4)$ and $(1- > 3- > 4)$ are independent.

notation	meaning
N	number of nodes in the network
β	pair-wise meeting rate between a node pair
K	block (generation) size
λ	block arrival rate to each flow
l	packet size in bits
b	number of packets can be exchanged in one direction during a meeting
B	number of relay packets a node can store
q	size of finite field GF_q, where $q = p^n$, p is a prime and n is a positive integer.
d	dimension of the packet in the finite field GF_q
A	encoding matrix
r	rank of the encoding matrix
D_{block}	the time to deliver the last packet of a block
L	the number of per-packet tokens

Table 3.1. Tables of notations used in Chapter 3

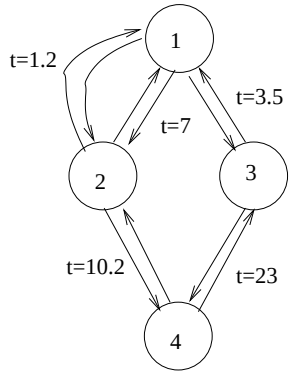


Figure 3.1. Random graph representing the contacts between nodes

3.2.2 Forwarding, Recovery Schemes

When two nodes come within transmission range of each other (i.e., they meet), they first each figure out if the other has some useful information and, if any, they try to exchange it. We detail this process with reference to the two mechanisms we are going to compare: non-coded packet-forwarding and RLC-forwarding.

Non-coded forwarding: When two nodes meet, each of them uniformly randomly selects one or more packets, depending on the available bandwidth, among the packets that the other node does not have, and forwards them to the other

node. We refer to this scheme as the **random** selection scheme. We also consider a **RR_random** scheme in which the packet's source node chooses a packet to forward in round-robin manner, while intermediate nodes perform random selection. Intuitively, the round-robin scheduling at the source node help speed up the propagation of initial copies of each packet.

As each packet is duplicated by the nodes in the network, when it is first delivered to the destination, there are multiple copies of the packet in the network. A recovery scheme can be used to delete these obsolete copies from the network to free up storage space and avoid useless transmission [43]. We will focus on VACCINE recovery scheme. Under VACCINE, when a packet is first delivered, an *antipacket* is generated and propagated through the network (in the same way as data packets) to delete buffered copies of this packet.

Random Linear Coding based forwarding: RLC is applied to a finite set of K packets, $m_i, i = 1, 2, ..., K$, called a *generation*. Under the RLC based scheme, each packet is viewed as a d dimensional vector over a finite field [80], GF_q of size q. More specifically, a packet of l bit length is viewed as an $d = \lceil l/log_2(q) \rceil$ dimensional vector over GF_q, i.e., we have $m_i \in GF_q^d, i = 1, 2, ...K$.

A linear combination of the K packets is:

$$x = \sum_{i=1}^{K} \alpha_i m_i, \ \alpha_i \in GF_q,$$

where the coefficients $\alpha = (\alpha_1, ..., \alpha_K)$ are referred to as *encoding vector*, and addition and multiplication are over GF_q, and the generated linear combination, x, are referred to as the *encoded data*. Each original packet, m_i, can be viewed as a special combination with coefficients $\alpha_i = 1$, and $\alpha_j = 0, \forall j \neq i$.

Under the RLC scheme, linear commbinations of the packets, together with the coefficients, are stored and forwarded by network nodes. If a node carries r linearly independent encoded data, $X = (x_1, ..., x_r)$ (together the corresponding encoding

vectors), we say that the rank of the node is r, and refer to the $(K \times r)$ matrix made up by the encoding vectors as the node's *encoding matrix*, $\mathbf{A}$. Initially, the source node(s) carries the original packets, $M = (m_1, ..., m_K)$. When two nodes, say u and v meet, they first send their encoding matrices to each other, and then perform the following operations which we describe using node u as example. Node u, based on the matrix of node v, checks if it has useful information for node v[1]. If so, node u generates a random linear combination of the currently stored combinations, say $x_1, ..., x_r$, $x_{new} = \sum_{j=1}^{r} \beta_j x_j$, where the coefficients $\beta_1, ...\beta_r$ are chosen uniformly randomly from the field GF_q. Obviously, x_{new} is also a linear combination of the original K packets. This new combination, along with the coefficients *with respect to the original packets*, is forwarded to node v. Given that node u has useful information for node v, this randomly generated combination is useful to node v (i.e., can increase the rank of node v) with probability greater or equaled to $1-1/q$, according to Lemma 2.1 in [29]. We further note that as node u has knowledge about the encoding matrix of node v, node u can iteratively generate random linear commbinations from its stored combinations, until a combination useful to node v is generated. Such processing pays computation overhead to attain savings in transmission bandwidth. We do not consider such processing in this thesis.

Essentially, a node with rank r has stored r linear equations with the K source packets as the unknown variables, i.e., $AM^t = X^t$, where M^t represents the original K packets. When a node (e.g., the destination) reaches rank K (i.e., full rank), it can decode the original K packets through matrix inversion, as $AM^t = X^t$ leads to $M^t = A^{-1}X^t$. We then can use the Gaussian elimination algorithm to solve for the original packets, M^t. [2].

[1] In fact, if a node has at least one combination that cannot be linearly expressed by the combinations stored in another node, it has useful (i.e., *innovative*) information for the latter node.

[2] A packet can be decoded before the matrix reaches full rank, as long as the encoding matrix contains a simple encoding coefficient.

Similar to non-coded scheme, when a generation is delivered to the destination, the destination generates an antipacket for the generation, which is subsequently propagated maximally in the network to delete buffered combinations of the generation (i.e., VACCINE recovery). To simplify analysis and simulation, we assume that the storage and transmission of antipacket are not subject to bandwidth and buffer constraints.

3.2.3 Performance Metrics and RLC Scheme Overhead

We define *block delivery delay*, D_{block}, as the time from the arrival of the block in the network to the delivery of the whole block to the destination. We compare this time deliver a block of packets when the packets are forwarded without any coding or when RLC is applied to the block of packets. Depending on the specific application, other metrics could be more meaningful, like the average time to deliver a packet in the block, or the average time to deliver a packet in order. Note that D_{block} is the metric more favorable to RLC in the comparison. Other performance metrics of interest are the average number of packet copies or combinations made within the network, as this is a measure of resources consumed (bandwidth, transmission power, buffering) within the DTN.

Compared to non-coded scheme, the RLC based scheme incurs additional overhead in computation, storage and transmission. First, when nodes come in contact, they check if they have useful information for each other before they generate and forward linear combinations. This, together with the decoding performed at the destination, introduces computational overhead. One can also consider a scheme where such checking is not performed, which save computation overhead by paying the price of some useless transmission. Secondly, as the encoding vectors are stored and transmitted along with coded packets, the RLC scheme pays extra storage and transmission overhead. Actually, the ratio between the size of the encoding vector (i.e.

coefficients) to the data packet is $K/(\lceil l/log_2(q) \rceil) \approx Klog_2(q)/l$; this overhead can be ignored when $l >> log_2(q)$.

3.2.4 Simulation Setting

Throughout this chapter, we mainly rely on simulation to quantify performance gains of the RLC scheme. We extend the simulator used in Chapter 2 to implement buffer management and transmission scheduling schems and the RLC based scheme. We report simulation results based on a pair-wise Poisson meeting process between two nodes, rather than an actual mobility model such as random waypoint/direction mobility model. This simplification speeds up the simulation, and as [39] has shown, under the random waypoint/direction models, the inter-meeting time between a pair of nodes follows a Poisson process when node velocity is relatively high compared to the region size, and the transmission range is relatively small. We have also performed simulations using the actual mobility models and observe similar performance as the poission meeting simulation. For the results presented in this paper, we simulate a network of $N = 101$ nodes with a pair-wise meeting rate of $\beta = 0.0049$. We use a finite field of size $q = 701 = 701^1$, as field arithmetic operations of this field are simpler to implement than then those of the commonly used field $GF(2^8)$ or $GF(2^{16})$.

Note that we have outlined in Section 3.2 the RLC scheme in its most basic form. In reality, one could improve performance in terms of average delivery delay using various optimizations. For example, the node (including destination) can decode packet before the matrix reaches full rank, and forward the decoded packets to destination directly (other than generate random linear combinations). These approaches allow improvement in the average packet delivery delay for the RLC scheme.

3.3 Single Generation Case

Having described the network setting and the non-coded and RLC schemes, in this section, we investigate the benefit of the RLC scheme under simple setting where there is a single generation of packets in the network. In particular, we assume that K packets arrive at the same time in the network.We examine the following three scenarios:

- **SS_SD (Single Source/Single Destination):** the in K packets from a source are to be delivered to a single destination;
- **MS_SD (Multiple Source/Single Destination):** the K packets from different sources are to be delivered to the same destination;
- **MS_MD (Multiple Source/Multiple Destination):** the K packets (each from a different source) are to be delivered to different destinations.

3.3.1 Coding Benefit under Bandwidth Constraints

We first consider the bandwidth constrained case and assume when two nodes meet, they can send a maximum of b packets in each direction. Mobile nodes are assumed to have sufficient buffer space to store all packets in this section.

3.3.1.1 Analysis of RLC benefit

Recall that we use temporal network to represent the dynamic network formed by the nodes. It's easy to see that for the SS_SD case, where the source node at time $t = 0$ has K packets to send, the minimal time to deliver these K packets is the time when there are K independent paths from the source to the destination. Similarly, for the MS_SD case, the minimal time to deliver the K packets from the K source nodes to the single destination is the earliest time that there are K independent paths from the K source nodes to the destination.

Notice that in DTN routing schemes, the mobile nodes have no knowledge, or only delayed knowledge about packets transmission decision made by other nodes. As a result, a node along the K paths might choose to forward information that some other path is forwaring or has forwarded. Under the RLC scheme, rather than choosing from the K packets, nodes randomly and independently combine packets to generate "equally important" encode-packets. As the number of independent coded packets is much greater than K, the probability that some path forward data that is useless to the destination is much smaller than for a non-coded scheme. As [29] pointed out, such benefit of RLC scheme is well captured by the Coupon Collector Problem [35].

Next, we demonstrate that the RLC scheme achieves the minimum delay to deliver a block of packets with high probability.

Proposition 3.3.1 *If there is a single block of K packets in the network, for the SS_SD and MS_SD case, RLC achieves the minimum D_{block} with high probability. For the SS_SD case, the probability can be bounded as follows:*

$$p_{achieve\_min\_delay} \leq (1 - 1/q^K)(1 - 1/q^{K-1})(1 - 1/q^{K-2})...(1 - 1/q). \tag{3.1}$$

Proof: For the SS_SD case, in order to achieve the minimal block delivery delay, the K combinations generated by the source node to send along the K paths must be linearly independent. Under the basic RLC scheme, the probability that K combinations generated by the source is linearly independent is given by $(1 - 1/q^K)(1 - 1/q^{K-1})(1 - 1/q^{K-2})...(1 - 1/q)$[3], which yields the upper bound above. ■

We note that to achieve the minimal block delivery delay, it's necessary that *each node* along the K paths chooses to forward a combination that is independent from combinations forwarded by nodes along the other paths. Quantifying this probability

[3]This is obtained by dividing the number of ways of generating K linearly independent combinations of the K packets, $(q^K - 1)(q^K - q)(q^K - q^2)...(q^K - q^{K-1})$, over the total number of different ways of generating K combinations, $(q^K)^K$.

requires considerations for the exponentially large number of possible scenarios for the K paths.

In the remainder of this subsection, we first illustrate the benefit of the RLC scheme compared to non-coded scheme using a simple example. We then present two examples to demonstrate that the probability for the RLC scheme to achieve minimum block delivery delay is dependent on the specific meeting scenarios.

Consider the 4-node network as shown in Figure 3.1. Assume that at time $t = 0$, node 1 generates two packets m_1 and m_2 destined to node 4. By time $t = 23$, there are two edge-disjoint paths from node 0 to node 4, therefore the minimum delay to deliver the two packets is 23.

Under the RLC scheme, source node 1 forwards random linear combinations c_1 and c_2 to node 2, and c_3 to node 3 at the contacts at times $t = 1.2, 7, 3.5$ respectively. With proability $1 - 1/q$, c_1 and c_2 are independent. For the case where c_1 and c_2 are independent, node 2 stores both combinations. When node 2 meets node 4, it generates a random linear combination c_{12} of c_1 and c_2 and forwards it. If c_{12} and c_3 are independent, node 4 can decode the two original packets after node 3 delivers c_3 at time $t = 23$. Note that c_3 can be linearly expressed by c_1 and c_2, and with probability $1 - 1/q$, c_{12} is independent from c_3. For the case where c_1 and c_2 are linearly dependent, node 2 stores c_1, and forwards it to node 4 at $t = 10.2$. If c_3 and c_1 are independent (with probability $1 - 1/q$), node 4 reaches full rank at $t = 23$ and the two packets are delivered at the minimum delay. Summing up both cases, we conclude that for this particular contact scenario, RLC achieves minimum block delivery delay with probability $1 - 1/q$. In this particular example, the benefit of the RLC scheme over non-coded scheme is reflected in the forwarding decision made by node 2. Having no knowledge about the contents at other nodes (node 3), node 2 cannot choose to forward information that's not available at node 3. The RLC

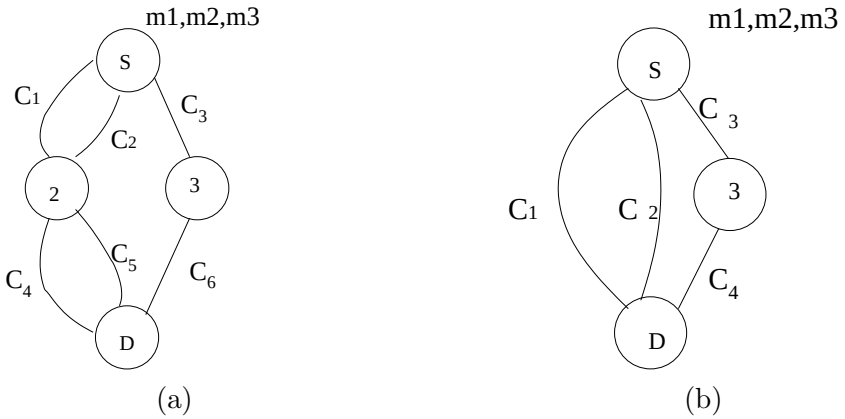


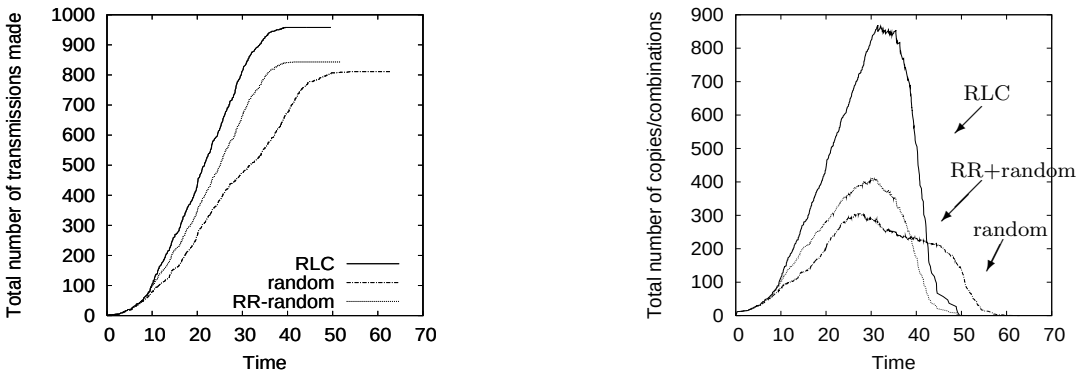
Figure 3.2. Two meeting scenarios for a 4-node network

scheme decreases the probability that a useless infomation is forwarded through its increased randomness.

On the other hand, under non-coded scheme, node 1 forwards m_1 and m_2 to node 2 at time $t = 1.2, 7$ respectively, and one of the packets (say m_1) to node 3 at time $t = 3.5$. When nodes 2 and 4 meet at $t = 10.2$, node 2 randomly selects a packet and delivers to node 4 (as it has no global knowledge of past and future contacts for other nodes). With probability 0.5, packet m_2 is selected to forward to node 4, and thus when node 3 meets node 4 at $t = 23$, it has no useful information for node 4. Hence, the non-coded scheme achieves the minimum delay with probability 0.5.

The probability for the RLC scheme to achieve the minimum block delivery delay depends on the particular meeting scenarios. For example, Figure 3.2 depicts two different meeting scenarios for a four-node network. We have, for the meeting scenario shown in Figure 3.2(a),

$$\begin{aligned} p_{achieve\_min\_delay} &= prob\{c_1,\ c_2 \text{ and } c_3 \text{ are independent}\} \\ &\quad \times prob\{c_4 \text{ and } c_5 \text{ are independent}\} \\ &\quad \times prob\{c_6 \text{ is non trivial}\} \\ &= (1-1/q^3)(1-1/q^2)(1-1/q)(1-1/q^2)(1-1/q)(1-1/q) \end{aligned}$$



(a) RLC scheme achieves faster propagation (b) storage comsumption

Figure 3.3. RLC scheme versus non-coded schemes

$$= (1-1/q^3)(1-1/q^2)^2(1-1/q)^3$$

Whereas, for the meeting scenario shown in Figure 3.2(b), we have

$$\begin{aligned} p_{achieve\_min\_delay} &= prob\{c_1,c_2 \text{ and } c_3 \text{ are independent}\} \times prob\{c_4 \text{ is non trivial}\} \\ &= (1-1/q^3)(1-1/q^2)(1-1/q)(1-1/q) \\ &= (1-1/q^3)(1-1/q^2)(1-1/q)^2. \end{aligned}$$

3.3.1.2 Characteristics of the RLC Scheme

We now highlight several characteristics of the RLC scheme compared to the non-coded schemes using simulation.

First, we observe that the RLC scheme allows faster propagation of the information in the network, but incurs more transmissions being made in the network. For example, for a particular run for the SS_SD case with $N = 101, K = 10$ case, Figure 3.3(a) and (b) respectively depict the cumulative number of transmissions made, and the total number of packet copies (for the non-coded schemes) or combinations (for the RLC scheme) in the entire network as a function of time. We observe that there are two factors causing more transmissions made under the RLC scheme. first,

the RLC scheme allows faster propagation of information, as the random combination performed at each node allows two nodes that meet each other to have useful information to exchange more often. Secondly, under the RLC scheme, the recovery process starts only when the whole generation is delivered, much later than under non-coded approach, where the recovery process for individual packet starts immediately when the packet is delivered.

The second point to make concerns the performance metric. Throughout this chapter, we mainly study the average block delivery delay as the performance metric; there are alternative metrics such as mean packet delay, in-order delay. For multiple simulation runs of the above setting, Figure 3.4 plots the empirical CDF for different delay metrics achieved by RLC and RR-random scheme. It shows that although RLC is able to decreases the block delivery delay, it sacrifices performance metrics such as mean packet delay and in-order packet delay.

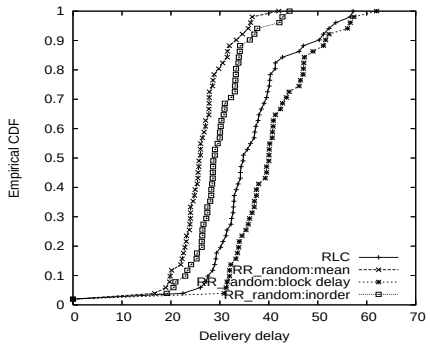
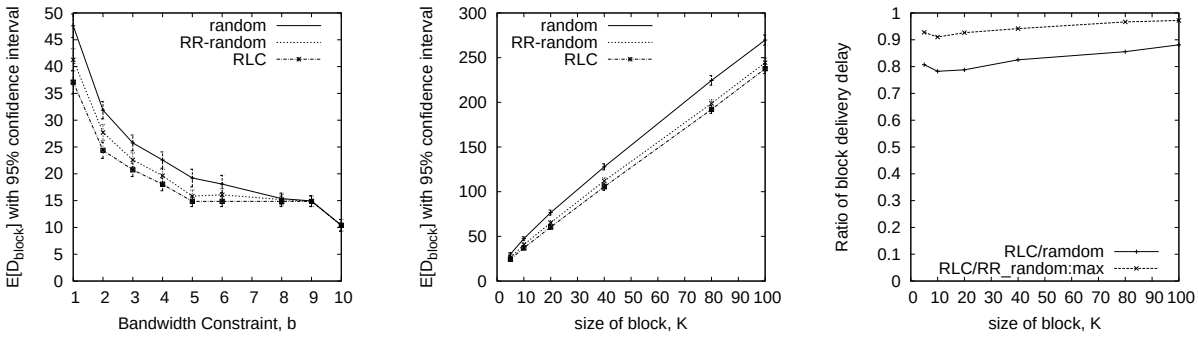


Figure 3.4. Comparison of different performance metrics under SS_SD with $N = 101, K = 10$

3.3.1.3 Performance Gain of the RLC scheme

We now quantify the performance gain of the RLC scheme through simulation. We note that due to the random nature of the contacts and the large size of the network in which we are interested, a quantitative analysis of delivery delay is difficult.



(a) $E[D_{block}]$ under varying bandwidth

(b) $E[D_{block}]$ under different block size

(c) Benefit of RLC under different block size

Figure 3.5. Benefit of the RLC scheme under SS_SD

We first explore the relative benefit of the RLC scheme with respect to the non-coded schemes under varying bandwidth constraints. Figure 3.5(a) plots the $E[D_{block}]$ and its 95% confidence interval for the SS_SD case with $K = 10$ under varying bandwidth constraints. (The average block delivery delay, $E[D_{block}]$ reported throughout Section 3.3 is the average value from 50 different simulation runs). The figure shows that that the RLC scheme achieves a lower $E[D_{block}]$ than both random and RR_random schemes. All schemes perform the same under $b = 10$ case where the $K = 10$ packets are propagated independently without competing for bandwdith; whereas as bandwidth decreases, the relative benefit of the RLC scheme increases.

We next study the sensitivity of the performance gain to the block size K. Figure 3.5(b) plots the average D_{block} for the SS_SD case with varying block size K and a bandwidth constraint of $b = 1$ (i.e., on every contact, only one packet can be sent in each direction. For the remainder of this chapter, this is the default bandwidth constraint used in our simulation results), and Figure 3.5(c) plots the relative benefit of the RLC scheme over non-coded schemes. We observe that as the block size increases, the relative benefit of the RLC scheme over non-coded schemes decreases. This is because for non-coded schemes, with a larger block size, there are a larger

number of packets to randomly choose from, and therefore the probability of two paths choosing to forward the same packets is smaller.

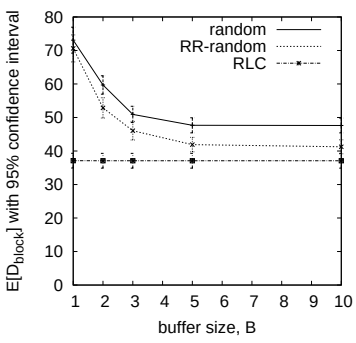
Our results for the MS_SD and MS_MD case are not shown here. We note that the benefit achieved by the RLC scheme for the MS_SD case is smaller than for the SS_SD case. Basically, under the MS_SD case, the K packets start to propagate from the K different source nodes, and the effect of relay nodes choosing the wrong packets to forward becomes less significant. For the MS_MD case, the RLC scheme performs worse than the non-coded scheme since the RLC scheme forces every destination node to receive K independent combinations to decode the one single packet destined to it.

3.3.2 Coding Benefit under Bandwidth and Buffer Constraints

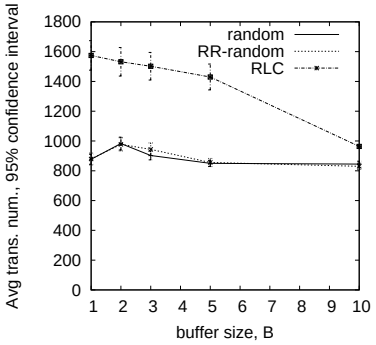
In the previous section, we have assumed that nodes have unlimited buffer capacity. In this section, we conisder the case where each node can store at most B ($B < K$) *relay* packets or combinations, but has sufficient buffer space to store all source packets and packets destined to it. As the nodes do not have enough memory to hold the whole block of packets, an algorithm is needed to determine what to keep or drop when the buffer is full.

We consider the following buffer management schemes. For the RLC scheme, when a node receives a combination and its buffer is full, it randomly combines the new combination with an existing combination in the buffer and stores the result. For the non-coded scheme, a drophead scheme [117] is used. Under the drophead scheme, when a new relay packet arrives to a node, and the node's buffer is full, the node drops the relay packet that has resided in the buffer the longest.

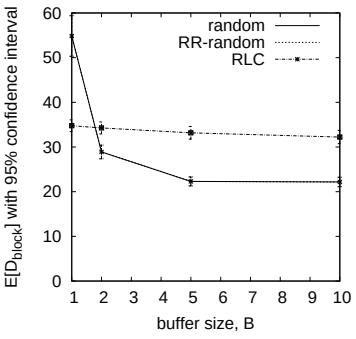
Figure 3.6(a) plots the block delivery delay under the RLC scheme and the non-coded schemes for the case SS_SD (with $K = 10$) under varying nodal buffer sizes. We find that as nodal buffer sizes decrease, the performance of the RLC scheme degener-



(a) $E[D_{block}]$ for SS_SD under varying buffer



(b) Number of transmissions made for SS_SD



(c) $E[D_{block}]$ for MS_MD under varying buffer size

Figure 3.6. Bandwidth and buffer constrained case

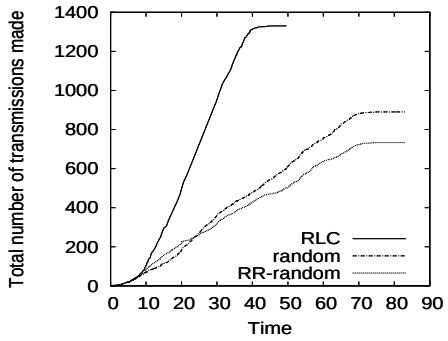


Figure 3.7. RLC scheme makes use of more transmission opportunities, B=1

ates only slightly; while the performance of the non-coded schemes degrade quickly. We examine the simulation trace closer to better understand the performance gain of the RLC scheme. For a particular run, Figure 3.7 plots the cumulative number of transmissions made as a function of time for different schemes. We see that the RLC scheme is able to make use of more transmission opportunities, and therefore propagates information much faster than the non-coded schemes. Further examination of the simulation traces reveals that under the RLC scheme, different information propagates evenly throughout the network. As different packets are mixed randomly by nodes, when a node drops a combination, an equal amount of information is lost for each packet. For the non-coded schemes, however, different packets in the block propagate at very uneven speeds: some packets spread quickly to a large number of

nodes, while other packets spread much more slowly. The uneveness of propagation of the non-coded schemes can be explained by the adopted random selection scheme: the more copies a packet has in the network, the more likely the packet is copied to some other node with the effect of kicking out copies of other packets. As a result of such uneveness, it takes much longer to deliver the "slowest" packet using non-coded schemes than using the RLC scheme.

Again, we note that the improvement in delay performance of the RLC scheme is achieved at the cost of more transmissions being made as shown in Figure 3.6(b). Notice that when there is no buffer constraint, at most K linear combinations of a generation (of size K) are sent to each node. This is however, not the case when there are buffer constraints. When a relay node cannot store all combinations of a generation, it can be repeatedly sent different combinations of a generation without increasing its rank.

For the MS_SD case, we observe similar performance gains achieved by the RLC scheme (not shown here). For the MS_MD case with $K = 10$, where coding is applied to packets sent by different sources to different destinations, we observe that the RLC scheme out-performs the non-coded schemes when the buffer is very constrained ($K = 10, b = 1$ for this setting) as shown in Figure 3.6(c).

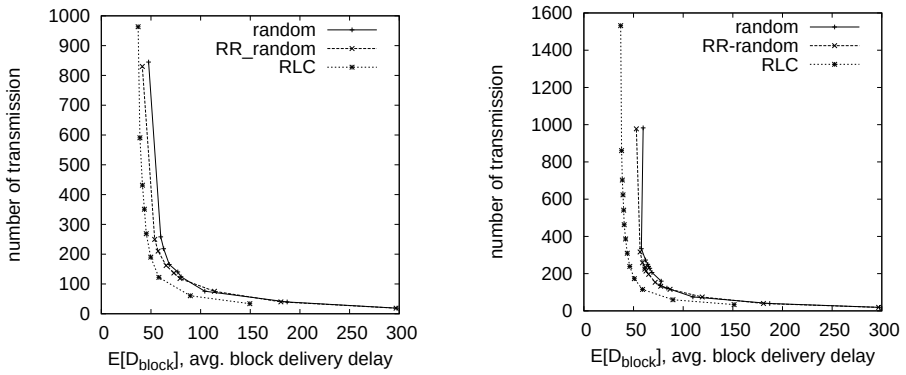
3.3.3 Controlling Transmission Power Consumption

So far, we have seen that the RLC scheme delivers a block of data, or collects multiple packets from different sources faster than the non-coded schemes, at the "cost" of having more copies of packets present in the network, consuming more buffer space, transmission power and bandwidth (to send these copies). Can the RLC scheme achieve a smaller average block delivery delay than the non-coded schemes *(i)* under the same transmission power consumption, *(2)* under the same transmission power consumption and buffer constraint ? We answer these questions in this section.

To limit the number of copies made of a packet, we use a token-based scheme, extending the *binary spray and wait* scheme proposed independently in [104, 101]. We refer to the maximum number of copies made for a packet as *the number of per-packet tokens*. Each node carrying a copy of a packet is assigned a token number that denotes the number of copies the node can make for the packet.

The spray and wait protocol [104] with the number of per-packet tokens, L, consists of two phases: a spray phase to spread $L-1$ copies of the packet, and a wait phase (if the destination node has not been reached) where each of L carriers (including the source) performs direct transmission to deliver the packet to the destination. There can be different ways to spread the initial $L-1$ copies, one of them is binary spray and wait. Under binary spray and wait with number of per-packet tokens L, every new packet generated at the source is assigned $L-1$ tokens. When the source node meets another node, the packet is copied to the other node and half of the tokens are assigned to the new copy, while the source node keeps the remaining half of the tokens. A relay node carrying a copy in turn does the same. When a packet copy has only a single token remaining, it can only be forwarded to the destination. [104] has shown that under an independently and identically distributed mobility model, binary spray and wait achieves minimum expected delay among all spray and wait routing schemes.

We note that this scheme can be improved by allowing two nodes carrying copies of the same packet to average their token numbers when they meet, as the two nodes have equal opportunities to meet susceptible nodes or destination node (and to propagate and deliver the packet). Furthermore, to apply the notion of tokens to the RLC scheme, we associate a token number with each generation, which limits the total number of combinations that can be exchanged for the generation in the network. The generation token number equals the product of the number of packets in the generation and the per-packet token number. When a node sends a random combi-



(a) Transmission power vs delay trade-off

(b) Transmission power vs delay trade-off under buffer constraint, B=2

Figure 3.8. Transmission number vs block delivery delay trade-off

nation to another node, its token number is decreased by one. After two nodes finish exchanging combinations of a generation, they average their token numbers: the sum of the two nodes' token number is reallocated to the two nodes in proportion to their ranks. Even if the two nodes meet each other have no information to exchange, they average their token numbers. The rationale behind averaging tokens in proportion to ranks is that the potential of a node to spread the generation is linear with the rank of the node, i.e., the "amount" of information the node carries for the generation.

We run simulations for the SS_SD case with $K = 10$ and the number of per-packet tokens varying between 5 and 90, and ∞. Figure 3.8 plots ((a) without buffer constraint, (b) with buffer constraint of $B = 2$), the number of transmissions versus delay tradeoff achieved under different per-packet token limits. The results show that even with similar transmission numbers, the RLC scheme is still able to out perform non-coded schemes. This is because the random mixing performed by the RLC scheme allows faster and more even propagation of independent information through the network. The results for limited relay buffer case further establish the usefulness of the RLC scheme in decreasing block delivery delay.

3.4 Multiple Generation Case

In the previous section, we examined the behavior of the RLC scheme in a single generation setting. We found that it provides faster delivery of a block of packets under bandwidth constraint, and that the delay performance degrades very slowly as nodal buffer becomes more and more constrained, at the cost of more transmissions. Although limiting token numbers leads to a larger average delay under both RLC and non-coded schemes, the RLC scheme achieves better transmission power versus delay trade-off than non-coded schemes.

The natural next question to ask is whether the benefit of RLC continues when one moves from a single generation case to the more realistic case where there are multiple continuous flows in the network. We address this question in this section by considering a scenario where there are multiple asynchronous continuous unicast flows in the network. In what follows, we first introduce the traffic process and scheduling schemes. We then present the results for the following two scenarios: when only bandwidth is constrained, and when both bandwidth and buffers are constrained. Finally, we discuss the feasible throughput of network under the non-coded schemes and the RLC scheme.

3.4.1 Traffic Process and Scheduling Schemes

We assume there are N flows in the network, with each node being the source of one flow and the destination of another flow. Each source independently generates a block of $K = 10$ packets according to a Poisson process with rate λ. Thus the total packet arrival rate to the network is $NK\lambda$. We only consider applying the RLC scheme to packets belonging to the same block, i.e. each block forms a generation, as this case has been shown to result in the largest benefit under the single generation setting.

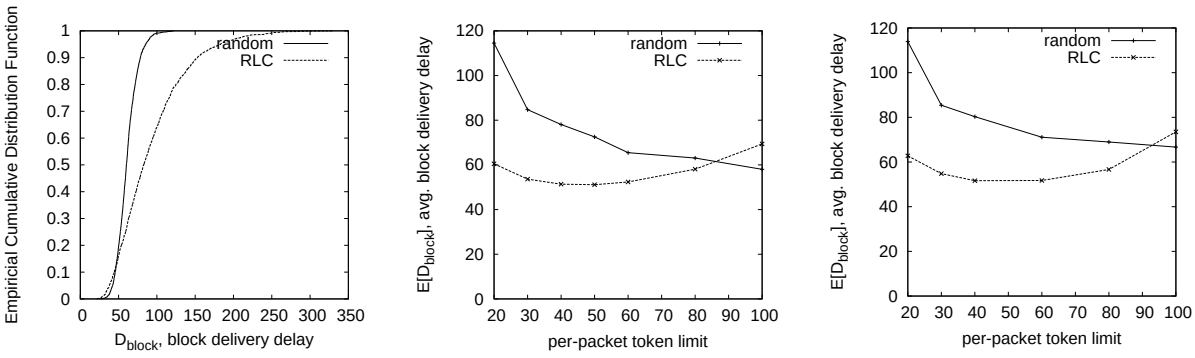
Our focus is on understanding the benefit of using RLC, not on designing an optimal scheme. Hence we adopt simple randomized scheduling. For non-coded schemes, when a node meets another node, it randomly selects a packet from the set of packets that it carries and the other node does not have, and forwards it. For the RLC scheme, the node first randomly chooses a generation from the set of generations that it carries which contain useful information for the other node, and then generates a random linear combination for this chosen generation to forward. For both cases, priorities are given to the packets/generations destined to the other node; furthermore, among such packets/generations, those originated from the sender itself are served first.

3.4.2 Coding Benefit under Bandwidth Constraint

We have seen that for one single generation, under bandwidth constraints, the RLC scheme achieves a smaller average delay than the non-coded schemes, because the RLC scheme can take advantage of more contact opportunities. We now examine the multiple generation case.

We perform simulations under varying block arrival rate with bandwidth constraint $b = 1$. We observe that the RLC scheme only shows a benefit when the traffic rate is low; and performs worse than the non-coded scheme when the traffic rate is high, as shown in Figure 3.9(a), which plots the empirical cumulative distribution function (CDF) of D_{block} under $\lambda = 0.00045$.

The reasons that the RLC scheme experiences worse performance than the non-coded scheme under relatively high traffic rates are two-fold. First, when the arrival rate λ is high, there is a large number of different packets in the network under the non-coded schemes; and it is more likely that two nodes have some useful information to exchange when they meet. As a result, the relative benefit of the RLC scheme through its increased randomization is smaller. Secondly, as we have shown in Figure 3.3(a),



(a) Empirical CDF of D_{block} under $\lambda = 0.00045$

(b) $E[D_{block}]$ under different token limit

(c) $E[D_{block}]$ under different token limit, $B = 5$

Figure 3.9. Block delivery delay under multiple generation case

the RLC scheme generates more transmissions for each generation than the non-coded schemes; this means that when the block arrival rate is high and there are many simultaneous generations in the network, the contention for bandwidth is severer under the RLC scheme. An optimal scheduler should favor generations that have fewer combinations spread throughout the network, but the currently implemented random scheduling scheme does not consider this optimization.

The trade-off between the average number of transmissions and average block delivery delay shown in Figure 3.8 suggests a way to deal with this resource contention problem. Figure 3.8 shows that the RLC scheme can achieve similar block delivery delays as the non-coded schemes with a significantly smaller number of transmissions (left part of the curve), so we expect a significant benefit by appropriately limiting the number of copies made of a generation. Figure 3.9(b) confirms that this is the case. Figure 3.9(b) plots the $E[D_{block}]$ achieved for the RLC and random schemes for a block arrival rate $\lambda = 0.00045$, when the per-packet token limit is varied between 20 and 100. In particular there is an optimal token limit value for the RLC scheme between 40 and 50 tokens. If the token limit is too large, the system suffers severe contention that degrades performance; if too small, some useful meetings cannot be exploited because all the tokens have been consumed. For the non-coded scheme

under this arrival rate, contention is not significant and the reduction in the token limit incurs larger delays. We do observe that under a higher block arrival rate, the non-coded scheme also benefits from limiting the number of copies made for a packet.

How to set the per packet token limit based on bandwidth constraint and block arrvial rate is an open problem. We can estimate an upper bound of the number of transmissions that can be made for each packet as the ratio between the total bandwidth available in the networks, $N(N-1)\beta$, and the total arrival rate, $NK\lambda$. For the specific setting considered here, this value is equal to 100.

3.4.3 Coding Benefit under Bandwidth and Buffer Constraints

We have seen in Section 3.3.2 that for a single generation case, the RLC scheme is especially useful when buffer is constrained. We now consider whether this is still the case when there are multiple generations in the network.

As for the single generation case, we assume that each node has limited buffer space for storing relay packets, but unlimited buffer space for storing its own source packets or packets destined to it. Since the source node always stores a packet until it is known to be delivered, there is no packet loss. Under the RLC scheme, when a node receives a combination and its buffer is full, it first selects one generation from its buffer to compress. This is done by randomly choosing one generation from the set of generations in its buffer that have the highest rank. If the newly received combination is for the chosen generation, the combination is combined with an existing combination within that generation; otherwise, the node compresses the matrix of the chosen generation by one [4] to make room for the new combination, and insert the new combination into the generation it belongs to. For the non-coded schemes, the drophead scheme is used.

[4]This is done by randomly combining two encoded packets into one.

When both bandwidth and buffer are constrained, limiting the number of transmissions made for each generation becomes even more important for the RLC scheme. As Figure 3.6(b) in Section 3.3.2 has shown, under a single generation case, the RLC scheme generates much more transmissions than the non-coded scheme. Therefore, when there are multiple generations in the network, resource contention is even greater than when buffer space is not constrained. We expect that a token scheme will allow bandwidth and buffer space to be allocated more evenly among different generations. We simulate the case of block arrival rate of $\lambda = 0.00045$, and every node only store $B = 5$ relay packets (combinations) under various token limits. As Figure 3.9(c) shows, the RLC scheme achieves a lower average block delivery delay than the non coding scheme, reducing the average block delivery delay by about 22.5%.

3.4.4 Feasible Throughput

In the previous two sections, we compared the average block delivery delay achieved by the RLC scheme and non-coded schemes under certain block arrival rates. An intersting question is whether network coding, can increase the throughput, i.e., the maximum per-flow block arrival rate that can be supported by the network.

When nodal buffer is not constrained, a DTN can be viewed as a traditional static network, where the link bandwidth represents the long term bandwidth available between the nodes (i.e., taking into account the meeting frequency). For the communication links in wireless networks, data transmission links along both directions share the same spectrum, therefore it's more natural to view the network as undirected network [79]. As conjectured by Li and Li [78], the benefit of network coding for multiple unicast sessions in direcional networks is likely to be non-existent.

The question of whether network coding scheme can increase throughput when nodal buffer is constrained remains to be answered.

3.5 Related Work

In this section, we first review briefly previous works that studied the benefit of network coding for wireless networks. Next, we compare in more details our work with previous work studying network coding benefit for broadcast applications in DTNs and wireless ad hoc networks. Then we review previous works that performed analytic studies of the RLC scheme for similar settings. Finally, we compare network coding approach with the source-coding approach.

Several previous works have investigated the benefit of network coding for wireless network. For multicast applications, Lun *et al.* [86] and Wu *et al.* [114] studied the problem of minimum-energy multicast, and showed that allowing network coding greatly simplifies the problem (from an NP-complete problem to a linear optimization problem solvable in polynomial time). For broadcast applications, Widmer *et al.* [112, 113] proposed RLC based scheme for energy efficient broadcast in mobile and static networks. For unicast applications, Wu *et al.* [115] and Katti *et al.* [66] studied the benefit of network coding in taking advantage of the shared nature of the wireless medium. Such benefit is applicable to relatively dense network, but not applicable to the sparse mobile network we are considering.

We now compare in more detail our work with those by Widmer *et al.* [112, 113], in which a RLC-based scheme was proposed for broadcast applications in DTNs and wireless ad hoc network, and was shown to achieve higher packet delivery rates than non-coded schemes under the same forwarding overhead. Our work differs from these two works as we consider unicast applications rather than broadcast applications. Even though we consider epidemic style routing where a flooding protocol is used for unicast delivery, it's different from broadcast delivery in that we consider token scheme for limiting the total number of copies made and a recovery scheme for deleting obsolete copies once the first copy is delivered. Moreover, we are interested in the overhead in terms of the number of copies made for each packet. Although [112]

considered generation management (i.e., how to decide which packets form a generation) and information aging (i.e., how to delete or compress information), they only reported simulation results for the single generation case. We have considered both single generation and multiple generation cases, and demonstrate that the coding based scheme is especially robust under relay buffer constraint.

Throughout this chapter, we assume that two nodes coming into contact will exchange information about what they carry, and only transmit data that is useful to the other node. This signaling protocol is equivalent to the *intelligent beacons* considered in [112]. We have also performed simulation using a signaling protocol equivalent to *normal beacons* in [112], where nodes do not exchange buffer content information before the actual data transmission. We find that under such signaling, the relative benefit of RLC is much more significant than the intelligent beacon signaling. This result is also in line with findings in [29] showing with intelligent signaling, the benefit of RLC scheme over non-coded scheme to be less significant.

Our main focus in this chapter is to investigate the benefit of applying network coding to unicast applications in DTNs, therefore we rely on simulations so that we can quantify the benefits accurately. Recent work by Lin *et al.* [81] proposed an ODE model for analyzing delivery delay under an RLC-based scheme and replication (epidemic routing) scheme, for the case of a single block of K packets propagating in resource constrained network. The model is proposed based on the assumptions that *(i)* two nodes with ranks between 1 and $K-1$, i.e., carry some, but not all information about a generation, always have useful information for each other, *(ii)* for all such nodes, equal fractions of them are of rank $1, ..., B-1$, *(ii)* under replication based schemes, the K packets are equally likely to reside each nodes. We comment that the results therein confirm our findings in this chapter, for example, the benefit of a RLC scheme under buffer constraints; and we have considered more complicated scenarios than them. A priority scheme is also proposed in the paper, which strictly

transmits packets of different priorities in sequence. We observe that such a scheme is not optimal in making use of contact opportunity, and therefore can be improved by priority transmission scheduing at each nodes.

As pointed out in Section 3.3.1, the benefit of RLC observed in our setting is similar in spirit to that of rumor mongering as studied [29]. They studied the problem of simultaneously disseminating multiple message in a large network, under the "random phone call" communication model where in each time step, each node communicates with another node randomly uniformly chosen from all the nodes. Through rigorous stochastic analysis, asympotic bounds for the delay under coding and non-coded schemes were derived. As both the communication model and the schemes considered (no signaling) therein differ from ours, applying similar analysis to our setting is non-trival.

Two previous works have investigated the application of erasure coding to DTNs, where the source node uses an erasure coding algorithm such as Reed-Solomon codes [93] and Low-Density Parity-Check (LDPC) based coding (e.g., Gallager codes, Tornado codes [85], or LT codes [84]) to encode a message into a large number of code blocks, such that if a fraction of $1/r$ or more of the code blocks is received, the message can be decoded. For DTNs where there are prior knowledge about paths and their loss behavior, Jain *et al.* [56] addressed the problem of allocating the source-erasure-coded blocks to the multiple paths between source and destination each with different loss behavior, in order to maximize the message delivery probability. Wang *et al.* [111] considered DTNs with unpredictable node mobility, and proposed to source-erasure-code message with a fixed overhead, and then send the large number of coded blocks over a large number of relays that then try to deliver them to the destination. Such erasure coding scheme allows for the usage of a large number of relays, in order to decrease the variance of the delivery delay, while maintaining a small fixed redundancy. Chen *et al.* [24] later proposed a hybrid scheme that combines the

erasure-coding based scheme [111] with a scheme that aggregressively forwards coded blocks, to achieve both good worst-case performance and small delay performance. Compared to the erasure coding scheme, RLC based schemes have different benefits and characteristics. An intersting open problem to consider is how to combine these two type of codings to attain the different benefits simultaneously.

3.6 Summary and Future Work

We study the benefits of applying RLC to unicast applications in mobile DTNs in this chapter. When there is a single generation in the network, we found that RLC achieves minimum block delay with high probability for a block of data destined to the same destination. Larger gains are achieved by the RLC scheme when buffer space is also constrained. Although the RLC scheme generate more transmissions, by using a token limit scheme, it can achieve better transmission power/delay tradeoff than non-coded schemes. When there are multiple generations in the network, under appropriately chosen token limit, the RLC scheme achieve a slight gain over non-coded schemes under only bandwidth constraint, and significant gains when nodal buffer is also constrained. Essentially, for epidemic style routing (i.e., replication based scheme) to work effectively in resource constrained DTNs, the most challenging problem is how to schedule packet transmissions and manage node buffers. RLC based schemes, where each node randomly combine multiple packets together to transmit to downstream node, and randomly evict a combination on buffer full, has higher degree of randomness compared to a randomized scheme. As a result, under RLC schemes, the probability that a node forwards/keeps a piece of information useful for the eventual delivery is greater than the case where random selection is done on per-packet base.

In the future, we plan to study several practical issues in applying RLC. First of all, we will analyze the computational complexity, and storage and transmission (signaling

and data transmission) overhead of RLC scheme. As the finite field size, q, affects the probability of achieving minimum delay, the complexity of encoding/decoding, and the storage/transmission overhead, choosing q is an important practical issue. Another practical issue concerns generation management. We have assumed that packets arrives to the source node in batches, which could arise from applications that generate large messages that are then fragmented to smaller packets to take advantage of more transmission opportunities. For applications that generate small messages, it is not reasonable to fragmented the message to even smaller packets, because the relative per (coded) packet overhead will be too large. For such scenarios, RLC can be applied to a group of packets whose generation times are close to each other. We expect the benefit of RLC schemes will be smaller.

We are also interested in performing analytic studies of the performance of RLC-based schemes and non-coded schemes to obtain closed-form (asymptotic) results. We expect an analysis for an ODE model such as done in [16] might be promising.

CHAPTER 4

MOBILITY TRACES MODELING AND IMPLICATIONS ON DTN ROUTING

4.1 Introduction

The many advantages offered by mobile communications have pushed wireless networks beyond supporting laptops in buildings to more challenging environments. Many of the wireless mobile ad hoc networks for groups of vehicles, pedestrians, or tracked wildlife experience intermittent node connectivity and disconnection of nodes or groups of nodes due to limitations of power, mobility, node density, and equipment failure. Network architecture and protocol designs that route data despite intermittent connectivity among nodes are generally referred to as Disruption-Tolerant Networks (DTNs). Such networks have been deployed in the context of buses [18, 12, 98], pedestrians [21], animal tracking [61], and underwater sensor networks [32].

Unlike other network regimes — such as tethered networks or multi-hop, unpartitioned MANETs — routing performance in DTNs is primarily affected by the frequency and duration of opportunities for data transfer between nodes. Therefore, when studying the performance of routing protocols and applications in DTNs, it is important to have models that accurately characterize these transfer opportunities.

There is a rich body of work on the measurement, characterization, and modeling of mobility traces taken from contemporaneously connected wireless LANs [109, 54, 72] and mobile ad hoc networks [58, 10]. Several recent studies have characterized traces collected from actual mobile networks with intermittent connectivity [21, 18] or adapted from traces collected from wireless LAN, and evaluate the impact of

the measured mobility on DTN applications [107, 49]. These works characterized only certain aspects of traces, e.g., the *aggregate* inter-contact time or the contact graph, without considering which aspects of the underlying mobility patterns are most important in determining DTN performance and therefore need to be captured and modeled accurately.

In this chapter, we develop a *generative* model of the inter-contact time of DTN nodes based on traces collected from our operational vehicular DTN, UMass DieselNet [18]. The model is generative in that it can be used to generate synthetic traces of node inter-contact times that can then be used to drive simulations. As we will see, however, these models are of interest in their own right, as models at the appropriate level of granularity can reveal structure that is hidden at the aggregate level and that can influence DTN performance. Indeed, a focus of our research is to understand the right level of modeling granularity so that traces generated by the model can then be used in simulation to accurately predict DTN performance. We show that while the all-bus-pairs aggregated inter-contact times show no clear pattern, inter-contact times at the bus-route level show periodic structure that can be modeled as mixtures of normal distributions (whose parameters can be inferred from empirical traces using an EM algorithm). Using a trace-driven simulation of epidemic routing, we show that this finer-grained route-level model of inter-contact times predicts performance much more accurately than the coarser-grained aggregated all-bus-pairs model.

The remainder of this chapter is structured as follows. Section 4.2 describes our testbed and trace data. In Section 4.3, we describe the performance metrics that we use to evaluate our generative model. In Section 4.4, we evaluate the aggregate model used in previous work and show that it does not perform well by our metrics. In Section 4.5 we propose a route-level model that generates synthetic traces that better match the routing performance of the original trace. We review related works in Section 4.6, and summarize this chapter and discuss future work in Section 4.7.

4.2 UMass DieselNet Traces

In this section, we first describe the UMass DieselNet [18] testbed, explain the traces collected and present background information about bus dispatching. We then describe our preprocessing of the traces, including merging two directional processes into one symmetric process, merging contacts that occurs close in time, and excluding falty devices from the traces.

4.2.1 Testbed and Trace Collection

UMass DieselNet consists of 40 buses serving the area surrounding the University of Massachusetts, Amherst campus. Each bus is equipped with a Linux computer, an 802.11b Access Point (AP), a second 802.11b interface, and a GPS device. The AP on each bus beacons its SSID once every 100 ms. The second radio continuously searches for SSID broadcasts. On discovering a remote bus's AP, the discovering bus obtains an IP address from the remote bus. Then, a TCP connection is opened, initiating a *contact event*, and data is continuously transmitted to the remote bus until the TCP connection is broken when the buses move out of range. Once the socket reports an error or closure, the contact event is marked as ended and logged. For each contact, the receiver logs the ID of the sender, the time, duration, and the number of bytes received. These *bus-to-bus transfer records* are transmitted to a central repository whenever a bus is able to associate with a fixed 802.11 access point that is attached to the Internet (e.g., offered by a cafe or in the bus garage). We refer to the records of the times and locations that each bus connects to fixed APs as *(bus-to-AP) check-in records*.

It is helpful to understand how buses are scheduled and dispatched since these are the primary determinants of bus mobility. The bus system serves approximately ten *routes*. Some routes have more buses running at the same time than others. In this work, we focus on the three most popular routes, the campus SHUTTLE that

tours the campus in a butterfly shape route (see Section 4.5.2 for details) and the SN_SA and NA_BR routes that travel between our campus and nearby towns within 150 square miles. During weekdays, beginning at approximately 7 A.M. and ending at approximately 7 P.M., there are multiple *shifts* serving each of these three routes. The shifts within a route are spaced so that there is a 15-minute spacing between shifts. Each of the other seven routes is served by only one or two shifts at a time, or they may be served only every other day, and in general we have fewer data points characterizing their operation.

For dispatching and driver assignment purposes, shifts are divided into morning (AM), midday (MID), afternoon (PM), and evening (EVE) *sub-shifts*. In the morning, drivers choose buses at random to run the AM sub-shifts. At the end of the AM sub-shift, the bus is often handed over to another driver (often at a bus stop) to operate the next sub-shift; but in some cases, the bus returns to the bus garage, and it is then possibly assigned to another shift on that route or to another route.

Our results are based on the study of 55 days of traces collected during the spring 2006 semester, from Jan 30 to May 28 with weekends, spring break, and holidays removed since during these times the buses run on reduced schedules. We focus on the events logged between 7 A.M. and 7 P.M. for each day, when buses are running regularly. We also use bus dispatching records, which record the mapping from buses to routes and shifts for each day. Both the traces and dispatching records are available for download at http://traces.cs.umass.edu.

4.2.2 Mobility Traces Preprocessing

As discussed earlier, when two buses are in transmission range, each one connects to the other's AP to transmit data to the other bus using a separate TCP connection, i.e., the recorded contacts are directional. The contacts in both directions are over the same 802.11b channel; as a result, during one physical meeting of two buses, there

can be multiple directional contacts (in both directions) as they gain or lose access to the shared channel. We note that this is due to the way the current system is built; we can imagine systems where symmetric contact is established when two buses meet or where two different channels are used for the contacts in each direction. As we wish to focus on the mobility rather than the specific operations of the bus nodes (including MAC layer), we assume all contacts are symmetric, i.e., data can flow in either direction.

Figure 4.1. A contact process between bus A and bus B. Here X_2, X_3 are fully observed inter-contact time, X_1 is a start-censored observation, and X_4 is a end-censored observation of inter-contact time

Figure 4.1 illustrates the contact process between two buses, A and B, during a day. In the figure, we use black boxes to represent contacts and spaces in-between to represent the interval between contacts. We refer to the duration of time between two subsequent contacts as the *inter-contact time*.

In our traces, there are many very short inter-contact times. This occurs, for example, when two buses that travel closely in the same direction repeatedly come in and out of range of each other as their spacing changes with the road traffic. We merge such events as little else can occur between the contacts. Specifically, for each pair of buses, we combine any two subsequent contacts that occur within 60 seconds of each other. The merged contact has a starting time equal to the earlier contact's starting time and an ending time equal to the later contact's ending time.

We observed that some buses operating on routes during a day were not observed in the traces. It may be the case that the bus did not physically meet other buses, but it may also be the case that the bus failed to set up TCP connections when in range of other buses. There are several reasons for the latter. When the buses

	transfer & check-in	transfer & no check-in	no transfer & check-in	no transfer & no check-in
Counts	1055	42	34	228
%	77.91	3.1	2.5	16.84

Table 4.1. Number and fraction of four different cases of daily bus status. There are 1,354 records in total.

are moving at high speeds, there is not enough time for two passing buses to form an 802.11 association and initiate a TCP transfer; we have data on bus speeds that confirms this problem. Hardware failures are not uncommon on the testbed and occasionally mechanics disable the system when servicing the bus and neglect to enable it afterwards. As we don't know whether devices were functioning correctly when the traces were collected, for any day, if we observed a bus in the bus-to-bus transfers or bus-to-AP check-in records, then we assume the device on the bus worked properly for that whole day; otherwise, the device was assumed to be faulty and the bus was removed from the trace for the day.

Table 4.1 shows, among all the buses running on routes during the whole trace, the numbers and fractions of instances that a bus *(i)* had transfer records and check-in records, *(ii)* had transfer records but no check-in records, *(iii)* had no transfer records but had check-in records, and *(iv)* had no transfer records or check-in records during a day. We observe that the correlation of "having a bus-to-bus contact" and "having check-in records" is high.

4.3 Performance Characteristic under the Trace

The goal of modeling the bus mobility trace is to correctly predict DTN routing performance. In the past, many routing schemes have been proposed for DTNs, but we focus here on basic epidemic routing. When there are no resource constraints in the network, epidemic routing provides the best-case delivery delay performance. In

this section, we first introduce epidemic routing and the performance metrics that we are interested in, and then describe the trace-driven simulation we develop for evaluating the performance of epidemic routing under given mobility traces.

4.3.1 Epidemic routing and performance metrics

Epidemic routing [110] adopts a "store-carry-forward" paradigm: a node receiving a packet buffers and carries that packet as it moves, passing the packet on to new nodes that it encounters. Analogous to the spread of infectious diseases, each time a packet-carrying node encounters a new node that does not have a copy of that packet, the carrier is said to *infect* this new node by passing on a packet copy; newly infected nodes, in turn, behave similarly. The destination receives the packet when it first meets an infected node, and initiates a recovery process that delete packets copies at infected nodes by the propagation of acknowledgment information in the network. Many recovery schemes have been previously proposed and studied [117, 18]; we will adopt the VACCINE recovery scheme in which acknowledgment information is propagated maximally in the same manner as data packet.

As we have seen from Chapter 2, for DTN routings, there exists a trade-off between packet delivery delay and the number of copies made for each packet, where delivery delay is an important performance metric for application and the number of copies made is directly related to transmission overhead. In this chapter, in addition to these two important performance metrics, we also study the *number of hops of the minimal delay paths discovered by epidemic routing*. This hop count metric is useful in setting the maximum number of hops in a K-hop scheme [39].

4.3.2 Trace-driven Simulation of Epidemic Routing

As our primary focus is on the impact of mobility on DTN routing, we assume there is no resource contention in the network in terms of bandwidth or buffers and that when two buses come into contact, they can instantaneously exchange an ar-

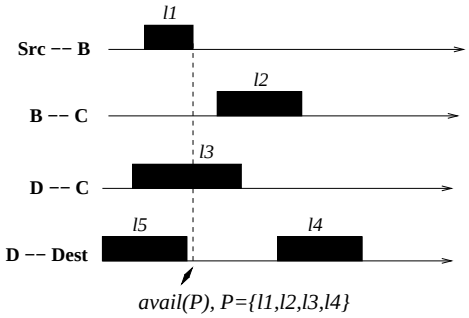


Figure 4.2. Example of a path $P = \{l_1, l_2, l_3, l_4\}$ from src to dest. Note that $\{l_1, l_2, l_3, l_5\}$ is not a time-respective path.

bitrary number of packets. We next describe the trace-driven simulation that we use evaluate the performance of epidemic routing under a given mobility trace under these assumptions.

A meeting trace can be represented as $G =< V, L >$, where V is the set of nodes and L is the set of edges. Each edge $l \in L$, represents a contact between two nodes $v_1, v_2 \in V$, and is labeled with the time interval that the contact happens, $[s(l), e(l)]$, where $s(l)$ is the starting time of the contact, and $e(l)$ is the ending time of the contact.

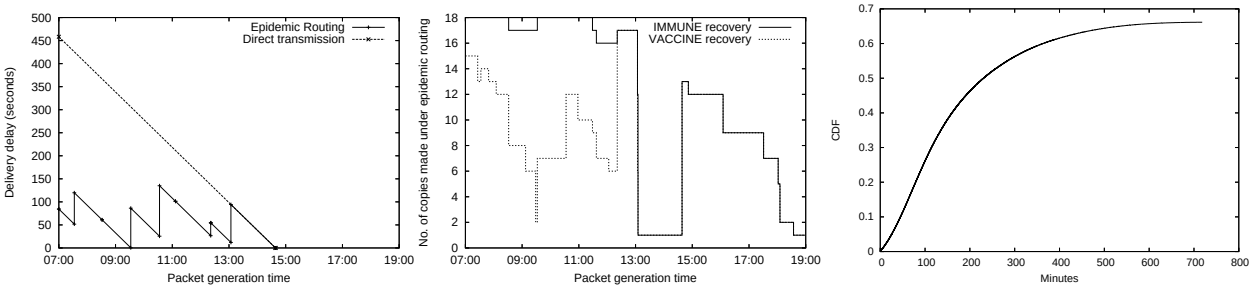
Under instantaneous transmission assumption, as Figure 4.2 demonstrates, in order for a packet generated at node *src* at time t to reach the destination node *dest*, a time-respective path in the network, $P = (l_1, ..., l_k)$, is required such that $e(l_1) \geq t$. A path is called time-respective path if the edges along the path have increasing time labels, i.e., $s(l_i) < e(l_j)$, for any $i < j$. Each path is available until a certain time, i.e., $avail(P) = min\{e(l_i), i = 1, ..., k\}$. A packet generated at time t traversing along path P will experience a delay given by $max\{0, s(l_i) - t, i = 1, ..., k\}$. This means that as t increases, the delay on a path decreases linearly with time, until it becomes 0 after which the delay remains 0 until $t = avail(P)$. We assume that packets generated at $t = avail(P)$ can be delivered through path P, but packets generated *right after* this time, denoted as $avail(P)+$, cannot be delivered through path P.

We wish to evaluate the delivery delay, the number of copies made, and the hop count of the path for packets generated *at any point of time for given source-destination pair* under epidemic routing scheme. We next consider the plots that depicts such information, e.g., the packet-delivery-delay versus packet-generation-time plot and the number-of-copies-made versus packet-generation-time plot. The following observations allow us to simulate the propagation of a finite number of packets to obtain these information.

First, one can show that, for a class of routing schemes including epidemic routing, the delivery-delay versus packet-generation-time plot of a node pair is piecewise linear[1]. As Figure 4.3(a) shows, the delay versus generatime-time plot is made up of multiple line segments, connected with vertical lines at time instances when a previous path becomes invalid, or when a new path is used from that time on. Similarly, the number-of-copies-made versus generation-time plot and the hop-count versus generation-time plot are also piecewise linear, and more specifically, step functions. Secondly, we observe that the time instances when contacts start (or end) are the time instances when new paths become available (or existing paths become invalid).

Based on the above piecewise linear property, we generate packets at time instances when contacts start and end, and then using the metrics (delay, copies made, hop count) obtained for these packets to obtain these metrics for packets generated at any time. More specifically, for each source-destination pair, a packet was generated respectively at the simulation start time, the starting time $s(l)$, the ending time $e(l)$,

[1]Actually, this class of routing schemes include all routing scheme where the forwarding/routing decision at a node does not change in between the two subsequent node-to-node contacts in the network.



(a) Delivery delay from bus 3029 to 3038 on April 4, 2006

(b) Copies made from bus 3029 to 3038 on April 4, 2006

(c) CDF of delay for all bus pair during the whole trace

Figure 4.3. Performance of epidemic routing under the trace

and *right after the ending time* $e(l)+$, of each contact.[2]. We then perform trace-driven simulations of epidemic routing for these trace packets.

As an example, Figure 4.3(a) plots the delivery delay under epidemic routing and direct source-to-destination transmission for packets sent from bus 3029 destined to bus 3038 at any time between 7 A.M. and 7 P.M. on April 4, 2006. We observe a significant difference between the delays achieved by epidemic routing and by direct transmission (i.e., where only the source can deliver a packet directly to a destination). As the two buses have only one contact on this day, the delay under direct transmission is very large. Epidemic routing, however, is able to make use of other buses to relay packets, achieving an average delay of 67.5 minutes. Figure 4.3(b) plots the number of copies made for a packet generated on the same day for this unicast pair under IMMUNE (where the acknowledgment is not propagated in the network; only destination node can "cure" infected nodes) and VACCINE recovery.

Making use of the piece-wise linear property of the above delay-(copies, and hop count) -versus-generation-time figures, we can evaluate the cumulative distribution function for these performance metrics *assuming packets arrive uniformly randomly*

[2]The packets generated right after $e(l)$, i.e., $e(l)+$, have a generation time of $e(l)$, but is marked as *trail* packets such that it cannot be sent during contact l.

to each bus pair, at times uniformly randomly distributed between 7 A.M. to 7 P.M. for all the 55 days in the trace. For example, Figure 4.3(c) plots the cumulative distribution function of packet delivery delay for all bus pairs over the whole trace. Our goal is to build a generative model that accurately captures DTN routing performance in terms of the above cumulative distribution functions of delivery delay, copies made and hop counts under epidemic routing.

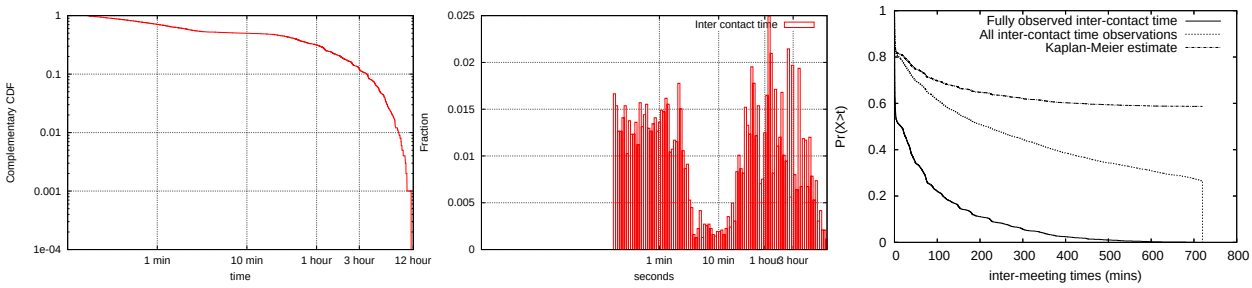
4.4 An aggregate model for bus DTN

The contact processes between node pairs, and in particular, the inter-contact times between node pairs, determine DTN routing performance. In this section, we characterize and model the bus mobility traces by studying the all-bus-pairs-all-day aggregated inter-contact time. Such approach has been taken by Chaintreau *et al.* [21]. The underlying assumptions made by such approach are *(i)* the contact processes of node pairs are renewal processes, *(ii)* there is no correlation between the contact processes of different node pairs.

In the remainder of this section, we first define the different inter-contact time observations in the trace. We then present the aggregate inter-contact time statistics. Finally, we evaluate a generative model based on the aggregate statistics.

4.4.1 Censored observations of inter-contact times

Recall that in Section 4.2.2, we have defined the inter-contact time as the time between two subsequent contacts. For any mobility trace, however, we have different inter-contact time observations. First, there are *fully observed inter-contact time*, measured as the duration of time from the end of a contact to the beginning of the subsequent contact, such as X_2, X_3 in Figure 4.1. There are also some incomplete observations of inter-contact times. Suppose that we measure the system from 7 A.M. to 7 P.M., for each bus pair, the duration from 7 A.M. to their first observed contact,



(a) CCDF of fully observed inter-contact times (b) Histogram of fully observed inter-contact times (c) Empirical CCDFs and KM estimate

Figure 4.4. Aggregate inter-contact times

such as X_1 in Figure 4.1, is a censored observation. We refer to such an observation as a *start-censored* observation, as we don't know when the inter-contact time starts. Similarly, the duration of time from the last contact between a bus pair to 7 P.M. is also a censored observation, which we refer to as an *end-censored* observation. For the case when two buses have no contacts during this measurement period, we have a *no-meeting* observation with duration given by 12 hr for the bus pair. For such an observation, we do not know the starting or the ending time of the inter-contact time.

To our knowledge, previous studies of mobility traces studied the inter-contact time solely based on fully observed inter-contact times and simply ignored censored observations, with Chen *et al.* [23] as an exception. Chaintreau *et al.* [21] recognized the effect of finite measurement duration, but did not consider its effect in their characterization of inter-contact time. As longer inter-contact times are more likely to be censored, ignoring censored observations leads to an under-estimation of the inter-contact time distribution, especially when the duration of the measurement is short.

4.4.2 Aggregate inter-contact time statistics

To study aggregate inter-contact time statistics, we first analyze, for each day, the contact process for each bus pair and obtain censored and fully observed inter-contact

times. We then aggregate all the fully observed inter-contact times and censored observations together.

Figures 4.4(a) and (b) plot the empirical complementary cumulative distribution function (ECCDF) and histogram of the aggregated fully observed inter-contact times, respectively. We observe that the fully observed inter-contact time distribution has two modes, and that there are many short inter-contact times.

The above figures do not suggest an obvious model for the aggregate inter-contact time distribution. Hence we adopt the standard Kaplan-Meier estimator (KM estimator) [63] to estimate the CCDF of the aggregate inter-contact time (also called a survival function), $S(t) := Pr(X > t)$, based on all observations. Suppose there are n distinct fully observed inter-contact times in the sample as follows: $T_1 < T_2 < ... < T_n$, and let $n_i, 1 \leq i \leq n$ be the number of inter-contact times, including both fully observed and censored observations, that are greater than or equal to T_i, and let $d_i, 1 \leq i \leq n$ be the number of inter-contact times of length T_i, then the KM estimator for $S(t)$ is:

$$\hat{S}(t) = \prod_{T_i<t} \frac{n_i - d_i}{n_i}. \tag{4.1}$$

Eq.(4.1) is the non-parametric maximum likelihood estimate of $S(t)$.

Figure 4.4(c) compares the survival function for inter-contact time (i.e., $Pr(X > t)$) estimated by the KM estimator, the ECCDF of fully observed inter-contact times, and the ECCDF of all observations. The results show a very large difference between the CCDF of fully observed inter-contact time and $\hat{S}(t)$. This comparison demonstrates quantitatively the importance of carefully accounting for censored observations when modeling inter-contact times.

4.4.3 Generative model based on aggregated statistics

In order to accurately model a DTN mobility trace, is it sufficient to model the all-node-pairs aggregate inter-contact time? To answer this question,we compare the routing performance of the original traces to the synthetic tracesgenerated based on the inter-contact time statistics.

To generate a synthetic trace that is comparable to the original trace, we generate traces for the same number of days. For each day, we generate the same number of active buses as in the original trace. The contact process between each bus pair for each day is generated as follows: we draw the time until their first contact (since 7 A.M.) from the observed samples of all the start-censored and no-meeting observations. The subsequent inter-contact times are drawn based on the KM estimate of the conditional distribution of inter-contact time given two buses have contacts on the day, calculated using fully observed inter-contact times and end-censored observations. The contact durations are drawn uniformly and randomly from the aggregate contact duration samples.

We first compare the number of contacts per day in the synthetic trace with that in the original trace. Figure 4.5 compares the scatter plots of the number of contacts versus the number of active nodes for all the days in the original trace and in the generated trace. It shows that the aggregate model generates a similar total number of contacts per day as the original trace.

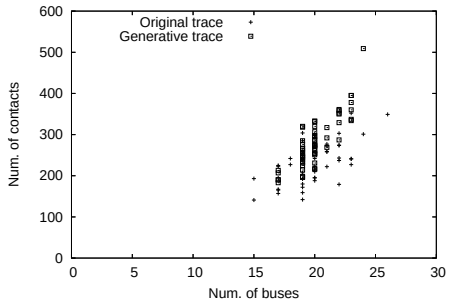
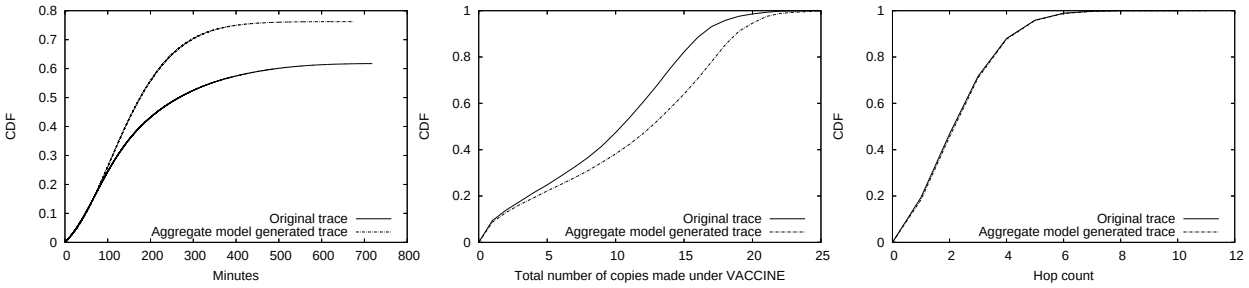


Figure 4.5. Comparison of no. of active nodes and contacts in aggregated model generated trace and original trace



(a) Delivery delay CDF (b) No. of copies made CDF (c) Hop count CDF

Figure 4.6. Comparison of epidemic routing performance under aggregate model generated trace and original trace

We then compare epidemic routing performances under the two traces. Figure 4.6 compares the all-bus-pairs-aggregated CDFs for delivery delay, total copies made in the network, and hop count under the original trace and the generated trace. The results show that many more packets are delivered and fewer copies are made for packets based on the generated trace than on the real trace, although the two traces have a similar number of contacts. (The CDF of the epidemic path hop count, however, is very close.) The reason is that under the aggregate model, contacts are equally distributed to all bus pairs, leading to more balanced connectivities for all buses, which in turn results in more packets being delivered. In fact, we observe that, under original traces, the delivery delays of different bus pairs can differ quite significantly, whereas the generated trace incurs similar performance for different bus pairs. This suggests the need for a finer-grained model to accurately predict DTN routing performance.

4.5 Modeling Route-level Aggregate Inter-Contact Time

Our study of the aggregate model in the previous section suggests the need for a finer-grain model in order to capture the heterogeneity among different buses. The next question is then: what granularity shall we use to model the mobility trace ?

One approach is to build the finest-grained model possible by characterizing the contacts between individual bus-pairs. This is problematic for two reasons. First, within a day, there are usually just a few contacts between a bus pair; there are simply not enough samples to accurately characterize the pair's contact behavior. Second, each bus is randomly dispatched to a route each day and may change routes during a day, so a bus pair exhibits different meeting behaviors on different days and even during different times of the day. Therefore, one cannot simply aggregate traces from multiple days. For the above reasons, we focus on the contact process between two buses running on certain shift pairs, i.e., shift-pair contact process, rather than the contact process between two physical buses. In the following subsection, we describe the process to construct a shift-pair contact process from the original trace, and we present the route-level aggregate statistics.

4.5.1 Route-Level Inter-Contact Time Statistics

Recall that for each route in the bus system, there are multiple simultaneous shifts continuously running back and forth on the route. We construct a shift-pair contact process from bus-pair contact processes, making use of the bus dispatching records. Figure 4.7 illustrates this process. Suppose that we want to generate the contact process between Shift01 and Shift02 (both belong to the SN_SA route). From bus dispatching record, we find that Shift01 (with duration $[t0, t1]$) is served by bus A, while Shift02 is served first by bus B during $[t2, t3]$ and then by bus C during $[t3, t4]$ (as shown by the two middle axes in the diagram). The overlapping time of the two shifts is $[t_s, t_e] = [max(t_0, t_2), min(t_1, t_3)]$. We then insert those contacts between bus A and bus B (shown by top axis) that occur when the buses are running on Shift01 and Shift02 respectively into the (Shift01, Shift02) contact process (shown by the bottom axis), and similarly for bus A and bus C.

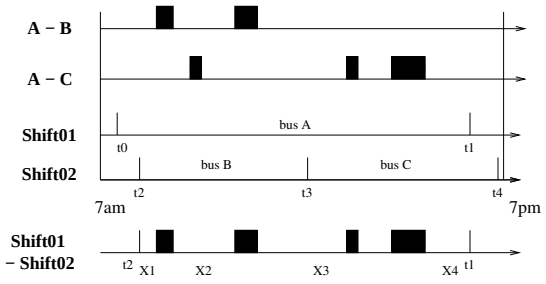


Figure 4.7. Obtaining (Shift01, Shift02) contact process from original traces using dispatching records. Bus A runs Shift01 during $[t_0, t_1]$, bus B runs Shift02 during $[t_2, t_3]$, bus C runs on Shift02 during $[t_3, t_4]$. For the (Shift01, Shift02) contact process, X_1 (X_4) is a start-censored (end-censored) inter-contact time observation, X_2, X_3 are fully observed inter-contact times.

In this particular example, our observation of the shift-pair contact process starts at t_s and ends at t_e (i.e., the duration of time that both shifts are actively running). Under our classification of different observations, we have X_1 (X_4) as a start-censored (end-censored) observation for the shift pair, and X_2,X_3 as fully observed inter-contact times. If we observed no contacts between two shifts, we introduce a no-meeting observation of length $t_e - t_s$.

As we expect different shifts within the same route to exhibit similar contact processes, we aggregate shift-pair inter-contact time observations that belong to the same route pair together to study route-level inter-contact times. For example, Figure 4.8 plots the histograms of the different observations of the inter-contact time for route pair (SN_SA,SN_SA). Let's first consider the censored observations. We observe the same number of start-censored and end-censored inter-contact times as expected. There are many instances when a pair of buses running on this route have no contacts. The histogram of the fully observed inter-contact times (Figure 4.8(a)) exhibits interesting periodic behavior and a trend of decreasing probability for longer inter-contact times. There are a large number of small inter-contact times (the first peak in Figure 4.8(a)). Recall that we have discussed the possible causes for very small inter-contact times in Section 4.2.2, and we have merged contacts that are less

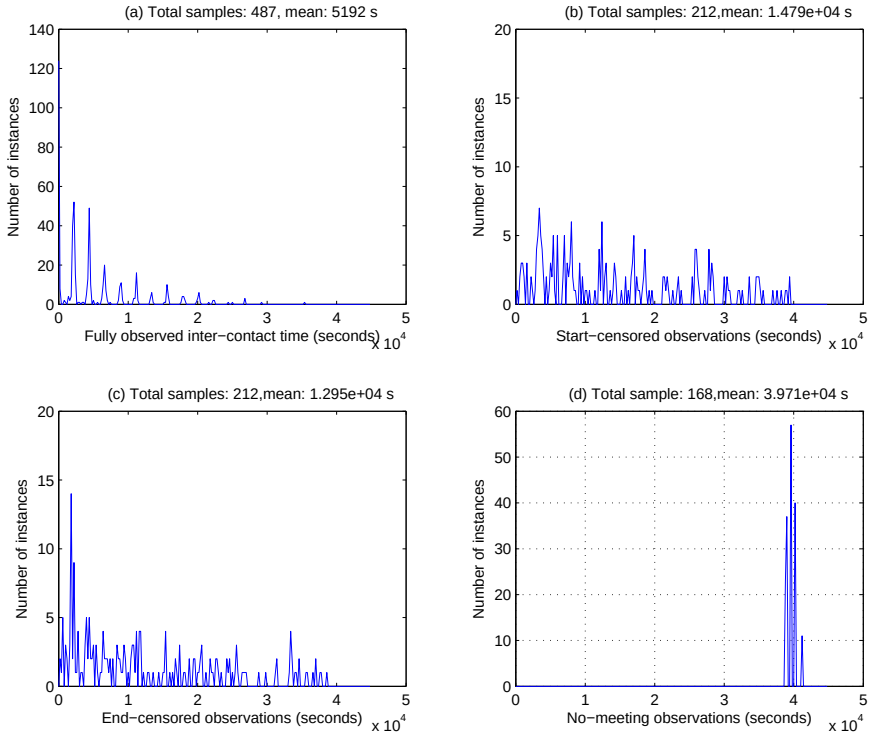


Figure 4.8. Observations of inter-contact times for SN_SA and SN_SA route pair

than 60 seconds apart. This figure suggests that there are still many instances of small inter-contact times even after this processing.

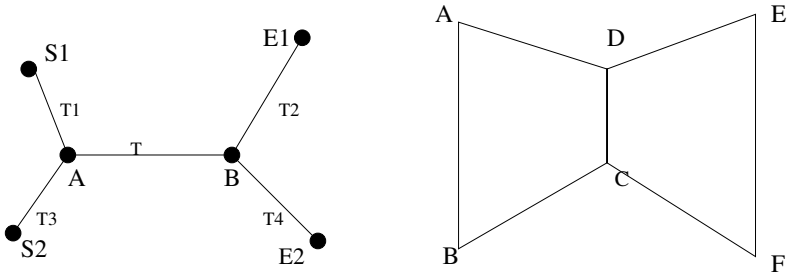
The histograms of fully observed inter-contact times for other route-pairs show similar periodic behavior. This suggests that there is interesting structure in the inter-contact times. To better understand the cause of such characteristics, we investigate the deterministic meeting behavior of buses in the next section.

4.5.2 Understanding Deterministic Meeting Behavior

In this section, we analyze the meeting patterns of two buses running on certain routes based on the assumption that buses operate according to planned schedules and run at constant speed. We define the *inter-meeting time* as the duration of time between when two buses are in transmission range; notice that this is different from

inter-contact time, which is defined as the duration of time between two subsequent contacts.

We first classify bus routes in our network as either a *linear* or *butterfly-shaped* route. On a linear route, shown in Figure 4.9(a), bus goes back and forth between two endpoints of the route. On a butterfly-shape route, shown in Figure 4.9(b), a bus either travels along direction $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow C \rightarrow D \rightarrow A$ or in the reverse direction.



(a) Two linear routes: S1−>A−>B−E1 and S2−>A−>B−E2. **(b) Butterfly Route**

Figure 4.9. Linear route and butterfly-shaped route

For two buses running on a same linear route, let the *round trip time* of the route be the time it takes for a bus to travel from an endpoint to another endpoint and then coming back to the starting endpoint, then they always meet every half round trip time, regardless of the spacing between them.

For a *butterfly-shape route*, let T_l be the travel time for the left loop (ABCDA), and T_r be the travel time for the right loop (CDEFC). The round trip time for the route is given by $T_l + T_r$. It's easy to show that two buses running in opposite directions either follow the inter-meeting time sequence $\{T_l/2, T_l/2, T_r/2, T_r/2, T_l/2, T_l/2, ...\}$ or meet periodically with period $(T_l + T_r)/2$. The latter case occurs when the two buses are spaced so that they do not meet in the joint segment $C - D$. Two buses running in the same direction meet in the $C - D$ segment if their spacing is exactly T_l or T_r. For the butterfly-shape route in our network, i.e., SHUTTLE, we observe that $T_l \approx T_r = T$, therefore a pair of SHUTTLE buses travel in opposite directions have

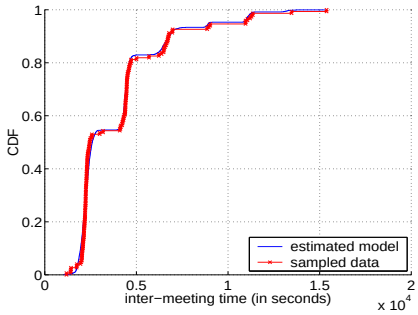
the following inter-meeting time sequences: $\{T, T, ...\}$ or $\{2T, 2T, ...\}$. In addition, SHUTTLE buses running in the same direction very rarely meet, as the buses are scheduled to avoid such meetings.

For two buses running on different routes, we divide the bus routes into smaller segments as needed, and we keep track the time that the buses enter or leave these route segments. If during some time interval, the two buses travel on a same segment in opposite directions, then they will meet each other in the middle of this time interval. For example, for the two linear routes that have overlapping segments (e.g., SN_SA and NA_BR) as shown in Figure 4.9(a), we consider the the following segments $S_1A, S_2A, AB, BE_1, BE_2$, and let T_1,T_2,T_3,T_4,T be the travel time for each segment. The deterministic inter-meeting times takes up to 5 different values; the inter-meeting time sequence varies depending on the time-phasing of the two buses.

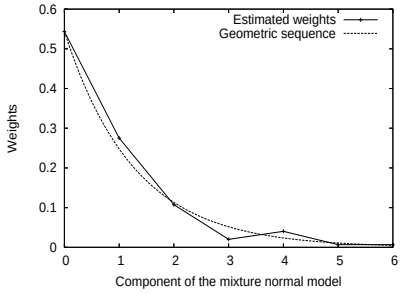
4.5.3 Mean-Restricted Mixture Normal Model

In the previous section, we considered the deterministic meeting sequences between bus pairs, ignoring random influence such as varying traffic and bus-operation conditions. We found that two buses running on a specific route pair had a fixed meeting sequence that is made up a number of inter-meeting times $T_{bi}, i = 1, 2, 3, ..., k$.

In reality, due to varying traffic conditions, bus speeds and other considerations, the inter-meeting time of buses is not constant, but rather a random variable that we can model as a normal distribution with mean T_{bi} and a certain variance. Furthermore, when two buses are in transmission range of each other, they are not always able to associate and transfer data, due to high bus speed, or because one of the buses is already in contact with a fixed access point. As a result, a data transfer can occur at the l-th physical meeting since the last contact ($l = 1, 2, 3, ...$). This means that an inter-contact time is made up of l inter-meeting times. As each inter-meeting time can be modeled as a normal random variable with mean given by $T_{bi}, i = 1, 2, 3, ..., k$,



(a) CDFs of data and estimated model



(b) Estimated weights and geometric seq.

Figure 4.10. Model fitting result for mean-restricted mixture normal models for SN_SA and SN_SA data

the inter-contact times can be modeled as the sum of l such normal random variables where $l = 1, 2, 3, ...$

For the case where there is a single inter-meeting time between a bus pair running on the route-pair, e.g., (SN_SA, SN_SA) and (NA_BR, NA_BR), or when the inter-meeting times are multiples of a single base value, e.g., (SHUTTLE,SHUTTLE) route pair, we propose the following mixture normal model for the inter-contact times:

$$f_{SM}(x) = \sum_{i=1}^{G} w_i f_N(x|i\mu, \sigma^2), \tag{4.2}$$

where f_N represents the PDF of the normal distribution parameterized by mean $i\mu$ and variance σ^2, μ corresponds to the base inter-meeting time T_b, σ^2 is the common variance for all normal components, the weights w_i depend on the specific inter-meeting time sequence for the route-pair, and we have $\sum_{i=1}^{G} w_i = 1$.

We derive an Expectation Maximization algorithm [14] to estimate the model parameters from fully observed inter-contact times[3]. The detail of the EM algorithm is given in Appendix C. As this model, and the model in the next section, focus

[3]Censored observations are not considered here, as we propose a model that is more appropriate for taking into account censored data in Section 4.5.4.

on the periodicity of inter-contact times, we have excluded the short inter-contact time observations when applying the model. We applied the model to study the (SN_SA,SN_SA) data set, and compared the empirical CDF of fully observed inter-contact time (with short inter-contact time removed) with that of the estimated model in Figure 4.10(a). We find that they match very well.

The above model has incorporated the periodicity by setting the means of the normals to be multiples of a single base value. From the original data (e.g., Figure 4.8), we also observe a geometric trend in the heights of the different normal components. In Figure 4.10(b), we plot the estimated weights, and we find that they match quite closely with the curve of the geometric sequence $p^{i-1}(1-p)$, with $p = 1-w_0$. Actually, if we assume there is a fixed probability that two buses fail to set up a contact when they meet, then we have $w_i = p^{i-1}(1-p)$, where i is the number of meetings until a successful contact.

As for the case when there are multiple inter-meeting times between a bus pair running on a route pair, such as SN_SA and NA_BR route pair, one could consider a mixture of normals with the means set to different linear combinations of the basic inter-meeting times. As we don't have enough data samples for such route pairs in our network, i.e., (SN_SA, NA_BR), (SN_SA, SHUTTLE) and (NA_BR, SHUTTLE), we leave the modeling of them for future work.

4.5.4 Mean-Weight-Restricted Mixture Normal Model

The model proposed in the previous section has incorporated our knowledge about the deterministic meeting sequences of the route pair, but still involves parameters that have no clear physical interpretations, i.e., the weights and the number of components. Furthermore, it's not clear how to take into account censored observations when estimating model parameters. Nevertheless, our analysis of the weights as estimated by the model parameter estimation algorithm has suggested the following

models that explicitly model the probability of failing to set up contact when buses are in range.

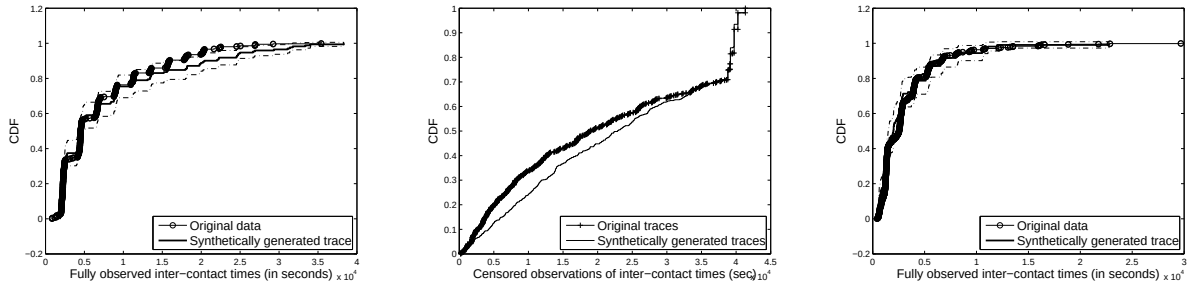
One-Base-Mean Model. For bus pairs with a single inter-meeting time, one can use the following model to characterize their inter-contact times:

$$f_{GEO\_1P\_1BM}(x) = \sum_{i=1}^{\infty} p^{i-1}(1-p) f_N(x|i\mu, \sigma^2), \tag{4.3}$$

where p is the probability that two nodes in transmission range fail to establish a contact, μ corresponds to the base inter-meeting time, and a single variance σ^2 is used for all normals as buses tend to keep to their schedules, so the variance does not add up. As there is a single base inter-meeting time μ, we refer it as one-base-mean model.

In Section 4.4.2, we have demonstrated that one needs to consider both fully observed inter-contact times and censored observations in order to correctly characterize the inter-contact time. Recall that we have used Kaplan-Meier estimator when we do not assume a model for inter-contact times. We now discuss how to account for censored observations when estimating parameters for the above model.

To consider censored data in the model parameter estimation, we first need to understand how the censored observations relate to the inter-contact times. Let's denote the PDF (Probability Distribution Function) and CDF of the inter-contact time as $f_X(x)$ and $F_X(x)$ respectively, and denote the mean of the inter-contact time as $E[X]$. We assume that the time we start to observe a shift-pair, i.e., the time that two buses enter the routes (t_s in Figure 4.7), is a random incidence into an inter-contact time interval. As a result, the time until we see the next contact, i.e., the start-censored observation is the residual lifetime [73] following a PDF given by $f_Y(x) = \frac{1-F_X(x)}{EX}$. If we observe no meeting, i.e., a no-meeting censored observation of length $t_e - t_s$, this means that we observe a residual life time that is longer than $t_e - t_s$ (the probability of which is given by $1 - F_Y(t_e - t_s)$, where $F_Y(x)$ is the CDF



(a) Fully observed inter-contact times for (SN_SA,SN_SA)

(b) Censored observations for (SN_SA, SN_SA)

(c) Fully observed inter-contact time (SHUTTLE,SHUTTLE)

Figure 4.11. Model fitting results for mean-weight-restricted mixture normal model

for $f_Y(x)$.). We further assume that all inter-contact times are equally likely to be cut off in the end. Thus an end-censored observation of value y means that a random inter-contact time has value larger than y, the probability of which is then $1 - F_X(y)$. Based on the above analysis of the censored data, we derive an EM algorithm to estimate the parameters p, μ, σ in model (4.3) from empirical data (details are given in Appendix D).

It turns out that the above model doesn't provide a good fit to the (SN_SA,SN_SA) data set. A careful examination of the traces and the bus schedule reveals that some shift pairs have fewer contacts than other shift pairs. This is mainly due to the fact that different shift pairs meet at different points within the route segment, some meet at high speeds, others meet at more congested downtown areas at low speeds. When buses traveling at high speed come into transmission range, there is shorter duration of time for them to set up connection and transfer data, which means a higher failure probability in setting up a contact. Based on these observations, we extend the above model to account for such factors:

$$f_{GEO\_MP\_1BM}(x) = \sum_{i=1}^{C} w_i \sum_{l=1}^{\infty} p_i^{l-1}(1-p_i) f_N(x|l\mu, \sigma^2), \tag{4.4}$$

where C is the number of components, and w_i specifies the fraction of bus pairs that have failure probability given by p_i.

Similar to model (4.3), we derive an EM algorithm to estimate the parameters for the above model, using all observations. We then use model (4.4) with $C = 2$ to model (SN_SA, SN_SA) data set, and then generate synthetic traces based on the estimated model. Figure 4.11(a) and (b) respectively compare the CCDF of the model-generated fully observed inter-contact time and censored observations with those in the original traces. We observe that for the fully observed inter-contact times, the original data fall within the 95% confidence interval of the model. As to the censored observation, the match is less good. We believe that this is due to the fact that there are other failure conditions that haven't been taken into account in the model, such as bus hardware failure or hardware being turned off for certain duration.

Two-Base-Mean Model. Recall from our deterministic analysis of the SHUTTLE route that *(i)* a bus pair running on SHUTTLE in the opposite direction either meet every half round trip time T or every round trip time $2T$; *(ii)* a bus pair running in the same direction very rarely meet with each other. Based on this knowledge, we propose the following model for the inter-contact time for a pair of SHUTTLE buses running in the opposite directions:

$$f_{GEO\_2BM}(x) = \sum_{i=1}^{2} w_i \sum_{l=1}^{\infty} p_i^{l-1}(1-p_i) f_N(x|li\mu, \sigma^2). \tag{4.5}$$

where p is the probability that two buses in transmission range fail to establish a contact, $\mu, 2\mu$ correspond to the base inter-meeting times T and $2T$, and a single variance σ^2. We refer this model as two-base-mean model as there are two base inter-meeting times, μ and 2μ.

Similar to the One-Base-Mean model, we develop an EM algorithm to estimate the parameters for the above model from the fully observed data and censored data.

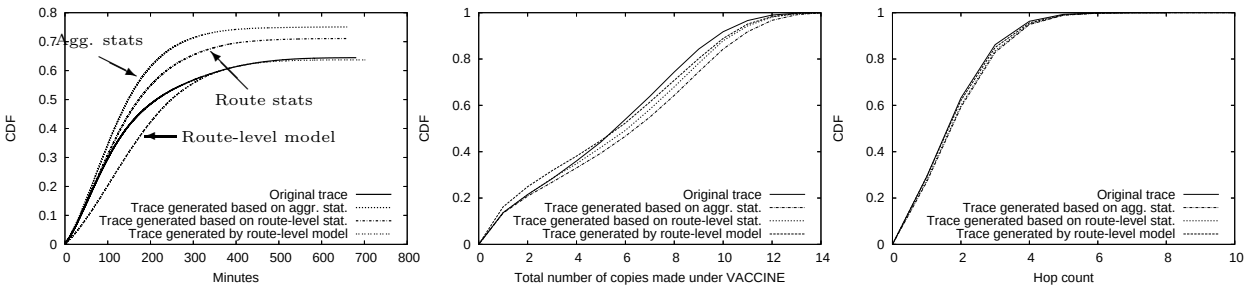
We apply the model to (SHUTTLE, SHUTTLE) dataset, and the results are plotted in Figure 4.11(c), which shows a good fit of our model to the empirical data.

4.5.5 Model Comparison

In this section, we compare three models, i.e., the model based on all-shift-pair aggregate statistics, the model based on route-level statistics, and the route-level model described in the last section, with a focus on their accuracy in capturing epidemic routing performance of the original trace.

We first process the original traces to include only buses running on the three routes that we have been focusing on, and we analyze this thinned trace to obtain aggregated inter-contact time statistics and route-level aggregated statistics. We then generate three synthetic traces. The first synthetic trace is generated based on all-shift-pair aggregated statistics, using the procedure described in Section 4.4.3; the second synthetic trace is generated based on the route-level statistics in a similar way; the third synthetic trace is generated using the route-level inter-contact models that we developed based on the route-level statistics in the last section, combined with route-level statistics for route-pairs that we don't have a model for. Last, we simulate epidemic routing respectively over the thinned original trace and the three synthetic traces. Figure 4.12 compares routing performance in terms of delivery delay, copies made, and the hop count of minimal delay paths under the four traces. We observe that under the four traces, the difference in delivery delay is the largest, followed by copies made. All traces have similar hop count CDF.

Similar to what we have observed in Figure 4.6 in Section 4.4.3, the trace generated by aggregate model exhibits significantly different performance compared to the original trace (i.e., the trace of inter-contact times empirically observed in the operational bus network). The trace based on route-level statistics, which is able



(a) CDF for delivery delay (b) CDF for no. of copies made (c) CDF of hop count

Figure 4.12. Comparison of epidemic routing performance under original trace and synthetic traces generated by three different models, respectively

to capture the heterogeneity among different bus routes, exhibits epidemic routing performance closer to the original trace.

Now let's focus on the the route-level model generated trace. We find that under this trace, all performance metrics are closer to the original trace than those of the previous two traces. In particular, under this trace, the average delivery delay is 15.8% larger than that under the original trace; and the packet delivery ratio is 0.75% less than that under the original trace. We think that the larger delivery delay and slightly smaller delivery ratio are due to the fact that our route-level model does not capture those short inter-contact times, and therefore it generates fewer contacts than the original ones. Nevertheless, as the model captures longer inter-contact times accurately, it's able to predict the longer time range delivery performance well.

As the route-level models are developed based on route-level statistics, it's somehow surprising that the prediction performance of the former is better than the latter. Our explanation is that in the route-level model we developed, we have treated the SHUTTLE bus pairs traveling in the same direction, and those traveling in the opposite direction differently, whereas in route-level statistics, we treated them together in SHUTTLE-SHUTTLE route pair. A related comment is that we expect that the granularity between shift- and route-level might be able to capture different meeting

behavior exists between different shift-pairs within a route, and therefore will achieve a good balance between model complexity and prediction performance.

4.6 Related Work

Many previous works have proposed mobility models [47, 109, 54, 72, 94, 48] for Wireless LANs or Vehicular Ad-Hoc networks, with some of them based on real mobility traces [109, 72]. Our work differs significantly from the above because we model the contact process between node pairs and in particular, the inter-contact time distribution between node pairs, as contact opportunity frequency and duration are the main determinants of DTN routing performance. Moreover, our traces are generally longer and more fine-grained than those used in previous studies.

Due to the lack of large scale DTN deployment, traces collected from campus WLANs such as UCSD [5] and Dartmouth College [45] of access point (AP) association records for laptops and PDAs were sometimes adapted to support DTN research. This is based on the assumption that mobile devices associating with the same AP would also be in transmission range of each other. Recently, several projects have collected traces from real DTNs, including the University of Toronto trace [107], iMote traces by Haggle project [95, 96]. The Reality Mining project at MIT [3, 33] has also made available node-to-node contact traces of mobile phones carried by students and faculty.

A couple of works have analyzed mobility traces and studied the impact on routing schemes. For example, Hsu and Helmy [49] also used the Dartmouth traces [45] to study encounter-based broadcasting. They studied the trace induced *encounter relationship graph* (where a pair of nodes is connected with an edge if they ever meet each other), found it exhibits a small world property, and showed that encounter-based forwarding is robust to selfish node behaviors. For both works, the traces used are WLAN traces, rather than a real trace collected from a DTN, and the approximation

with the same access point is used to infer contacts between a pair of devices. Another example is the works by Su *et al.* [108, 107] that studied DTN traces collected from a network of 20 students carrying PDAs with bluetooth radio. They studied direct contact and multi-hops paths between node pairs, and used trace driven simulation of epidemic routing and link-state routing to characterize the trade-off between delay and replication.

We next review several recent works that characterize DTN mobility traces through studying the inter-contact times between node pairs, similar to our works.

First, Chaintreau *et al.* [21] characterized the all-node-pairs aggregate inter-contact times from the UCSD trace [5], Dartmouth trace [45], Toronto trace [107] and three iMote traces, all of pedestrians carrying wireless devices. They observed an approximately power-law distribution of inter-contact times, with a power law index less than 1. Based on this observation, the authors proposed a simplified stationary *i.i.d.* contact model with power-law distributed inter-contact times, and they analytically studied the performance of different forwarding algorithms under such a model. The question of whether an aggregate model is sufficient for predicting DTN routing performance was not addressed. Their study did not account for censored observations in the characterization of aggregate inter-contact times. Furthermore, the claim that inter-meeting time follows power-law distribution needs more careful examination.

In their analysis of the UCSD and Dartmouth trace [5, 45], Chen *et al.* [23] took into account censored observations in their characterization of the aggregate inter-contact times, and proposed a censorship removal algorithm.

Karaginnis *et al.* [64] analyzed human mobility traces including the UCSD trace [5], the MIT Reality Mining trace [3, 33], the iMote traces [95, 96], and personal vehicular GPS dataset [74]. They found that for the aggregate inter-contact times of all traces, there is a characteristic time, in the order of half of day, beyond which the distribution decays exponentially; up to this value, the distribution follows a

power law. This is in contrast to previous hypothesis of power-law distribution by Chaintreau *et al.*, and suggests the prediction therein on routing schemes performance may be overly pessimistic. They also demonstrated that simple synthetic models can feature the above dichotomy, showing that simple synthetic models might be good for capturing human beings's mobility trace. Through further analysis of the trace, the author demonstrated that the dichotomy can be explained by the underlying returning time to favorite home of the mobile nodes. In addition to the above analysis, the author further explored the spatial and temporal heterogeneity within the trace. We comment that they have not considered censored observations, and therefore the conclusions about exponential decay can be an under-estimate of the actual inter-contact times.

A recent work by Conan *et al.* [26] studied the Dartmouth traces, iMote traces and the Reality Mining traces. Other than focusing on aggregate inter-contact time distribution, they studied pair-wise inter-contact times and found that log-normal distribution fits the largest fraction of data. They also provided a potential explanation to the approximate power-law distribution observed in the aggregate inter-contact times: when pair-wise inter-contact times follow exponential distribution with different rate, one can gain power-law phenomenon in the aggregate statistics. Finally, for a DTN with exponential pair-wise inter-contact times (with different rate for each pair), an opportunistic single-copy routing scheme (that minimize expected delay) is proposed and evaluated against other schemes.

4.7 Summary

In this work, we have studied mobility traces taken from UMass DieselNet, with the goal of building a generative model that can capture aspects of mobility (specifically inter-contact times) at the right level of granularity. The model is generative in that it can be used to generate synthetic traces to drive a trace-driven simulation. Al-

though this model is derived from mobility traces collected from a specific bus-based network, we expect such a model is applicable to other transport based networks that follow certain periodic schedules. Further work is needed to validate the model using mobility traces collected from different networks once they become available.

As the first careful study of a fielded system, the model is of interest in its own right, as they revealed structure that was hidden at the aggregate level — structure that can influence DTN performance. Indeed, using a trace-driven simulation of epidemic routing, we showed that this finer grained route-level model of inter-contact times predicts performance much more accurately than the coarser-grained aggregated all-bus-pairs model. This suggests that one must take care in choosing the right level of model granularity when modeling mobility-related measures such as inter-contact times, in DTN networks. Determining the appropriate granularity of models is both a difficult and a deep problem. At one extreme we can use a movement model, such as Brownian motion with parameters that are chosen to correspond to the parameters of the aggregate inter-contact time distribution obtained from a trace. At another extreme we can devise a model that accounts for the physics of the underlying system such as described above for our bus-based DTN network.

Our ongoing work includes the understanding and modeling of short inter-contact times. Our future research will focus on identifying the level of abstraction needed to produce good models, where goodness refers to how well generated traces statistically match collected mobility traces and how well models predicting the behavior of different information dissemination algorithms. We will also focus on developing techniques for teasing out the physical structure from a trace (such as the underlying periodic behavior in the inter-contact times) in the absence of domain knowledge.

CHAPTER 5

CONCLUSION

5.1 Summary of the Thesis

In this thesis, we have studied a class of Disruption Tolerant Networks with opportunistic contacts caused by node mobility and sparse density. We focused on three fundamental problems about unicast routing in such networks.

First, in Chapter 2, we proposed a unified Ordinary Differential Equations (ODEs) based framework to study the performance of epidemic style routing in DTNs. For DTNs with pair-wise exponential meeting process, we derived ODEs models from Markov chain models that characterize the forwarding and recovery process. The ODEs models allow us to derive a rich set of closed-form result for performance metrics of interest, i.e., packet delivery delay, number of copies sent, and buffer requirements. When numerical solution is used, the complexity does not increase with network size. We validated the models through simulations and explored the tradeoffs achieved by different schemes using the modeling result. We further extended the models with Markovian or fluid queue models, in order to consider the buffer-constrained case. We showed that with appropriate buffer management schemes, a much smaller buffer can be used with negligible effect on delivery performance.

Next, in Chapter 3, we studied the benefits of Random Linear Coding (RLC), a type of randomized network coding, for unicast applications in resource-constrained DTNs. For the case where there is a single generation in the network, we found that RLC applied to a block of data destined to the same destination achieves minimum block delay with high probability. A larger gain is achieved by the RLC scheme

when buffer space is constrained. Generally, the RLC scheme allows faster and more even propagation of information, due to its increased randomness. Although the RLC scheme generates more transmissions, by using a token limit scheme, the RLC scheme can achieve better transmission power versus block delivery delay tradeoff than the non-coding approach. When there are multiple generations in the network, under an appropriately chosen token limit, the RLC scheme achieves a slight gain over non-coding scheme under only bandwidth constraints, and a significant gain when nodal buffers are also constrained.

Last, in Chapter 4, we studied mobility traces taken from UMass DieselNet, and built a generative model that characterizes the inter-contact times at the route level. The model is generative in that it can be used to generate synthetic traces to drive a trace-driven simulation. As the first careful study of a fielded system, the model is of interest in its own right, as it revealed structure that was hidden at the aggregate level — structure that can influence DTN performance. Indeed, using a trace-driven simulation of epidemic routing, we showed that this finer grained route-level model of inter-contact times predicts performance much more accurately than the coarser-grained aggregated all-bus-pairs model, the approach that most previous works have taken. Our works suggests that one must take care in choosing the right level of model granularity when modeling mobility-related measures such as inter-contact times, in DTN networks.

5.2 Future Work

We have discussed some ongoing and future works in the previous three chapters. In this section, we outline additional new research directions that we would like to investigate in the future.

The problem of DTN routing under resource constraints is an inherently very hard problem, due to the severe resource constraints, local knowledge, uncertainty about

the future contacts, and more. However, there are also many possible ways to facilitate DTN routing. For example, mobile or fixed nodes can be strategically deployed in the network. For some applications such as transportation networks, it might be possible to optimally plan the trajectories and schedules of mobile nodes under certain application-imposed constraints, to achieve optimal network connectivity.

It's also imaginable that for some applications, the mobile node might be instrumented with two radios of different characteristics [61]. For example, [62] has shown this approach to be beneficial for power management purpose. It will be interesting to explore whether and how multiple radios can improve routing performance, for example, by dedicating radios with different transmission ranges for data communication and signaling respectively.

Throughout the thesis and in most previous works, unicast routing is considered. We imagine for some applications, this might not be the most preferable communication paradigm, and one need to explore other communication paradigms, including broadcast, publication and subscription networks. A related question is the design of prioritized transmission schemes to handle network traffic with different quality of service requirements.

Broadly speaking, the DTNs we studied in this thesis fall into the category of complex networks [15] that can arise from areas ranging from biological and chemical systems, neural networks, social interacting species, to the Internet and World Wide Web. Many previous works have characterized the structure of complex networks, and studied the different processes such as epidemic spreading on such networks. It is worthwhile to investigate results from this field and consider their relevance for DTNs.

APPENDIX A

DERIVATION OF ODES FROM MARKOV CHAIN THROUGH MOMENT CLOSURE TECHNIQUES

In this section, we show how the ODE model can be derived from Markov Chain model by ignoring variability and how variability can be taken into account using differential equations involving higher moments.

We consider the generic epidemic routing under IMMUNE recovery with a pair-wise infection rate of γ, and per-node recovery rate of β. Under the basic epidemic routing, we have $\gamma = \beta$; for probabilistic forwarding, we have $\gamma = p\beta$. A bivariate Markov chain as illustrated in Figure A.1 can be used to model the infection and IMMUNE recovery process, with state $(S(t), I(t))$ denotes a state where there are $S(t)$ susceptible nodes, and $I(t)$ infected nodes at time t, given that $S(0) = N-1, I(0) = 1$.

Define the state probabilities: $P_{s,i}(t) = Pr\{S(t) = s, I(t) = i | S(0) = N-1, I(0) = 1\}$. The Kolmogorov forward equation for the process is :

$$\begin{aligned} \frac{dP_{s,i}(t)}{dt} &= -P_{s,i}(t)(\beta i + \gamma si) + P_{s,i+1}(t)\beta(i+1) \\ & \quad P_{s+1,i-1}(t)\gamma(s+1)(i-1) \end{aligned}$$

Let $M(\theta_1, \theta_2, t) := \mathrm{E}[e^{\theta_1 s + \theta_2 i}]$ be the *moment generating function*. Multiplying the above equation with $e^{\theta_1 s + \theta_2 i}$, and summing over all possible s, i, we get:

$$\frac{\partial M}{\partial t} = \beta(e^{-\theta 2} - 1)\frac{\partial M}{\partial \theta_2} + \gamma(e^{\theta_2 - \theta_1} - 1)\frac{\partial^2 M}{\partial \theta_1 \partial \theta_2}, \tag{A.1}$$

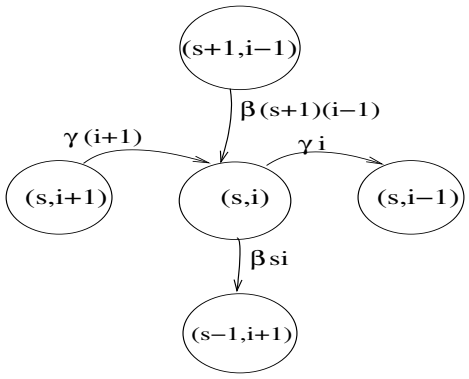


Figure A.1. Markov Chain for generic epidemic routing with infection rate γ, recovery rate β

We define the *cumulant generating function*, $K(\theta_1, \theta_2, t) := \log M(\theta_1, \theta_2, t)$, and observe that the following equations hold:

$$\begin{aligned}
\frac{\partial K}{\partial t} &= \frac{1}{M}\frac{\partial M}{\partial t} \\
\frac{\partial K}{\partial \theta_1} &= \frac{1}{M}\frac{\partial M}{\partial \theta_1} \\
\frac{\partial^2 K}{\partial \theta_1 \partial \theta_2} &= -\frac{\partial K}{\partial \theta_1}\frac{\partial K}{\partial \theta_2} + \frac{1}{M}\frac{\partial^2 M}{\partial \theta_1 \partial \theta_2}
\end{aligned}$$

Substitute these equations into Equation (A.1), we get:

$$\frac{\partial K}{\partial t} = \beta(e_2^{\theta} - 1)\frac{\partial K}{\partial \theta_2} + \gamma(e^{\theta_2 - \theta_1} - 1)\left(\frac{\partial^2 K}{\partial \theta_1 \partial \theta_2} + \frac{\partial K}{\partial \theta_1}\frac{\partial K}{\partial \theta_2}\right) \tag{A.2}$$

By taking partial derivatives of θ_1 and θ_2 respectively on Equation (A.2) and setting $\theta_1 = \theta_2 = 0$, we can get the following ODE system.

$$\begin{aligned}
\frac{d\bar{S}}{dt} &= -\gamma(\bar{I}\bar{S} + C_{IS}) \\
\frac{d\bar{I}}{dt} &= -\beta\bar{I} + \gamma(\bar{I}\bar{S} + C_{IS})
\end{aligned}$$

where $\bar{S}(t) = \mathrm{E}[S(t)], \bar{I}(t) = \mathrm{E}[I(t)], C_{IS}(t) = \mathrm{Cov}(S(t), I(t))$.

If we ignore covariance of $I(t)$ and $S(t)$, and set $C_{IS} = 0$, we get:

$$\begin{aligned} \frac{d\bar{S}}{dt} &= -\gamma \bar{I}\bar{S} \\ \frac{d\bar{I}}{dt} &= -\beta \bar{I} + \gamma \bar{I}\bar{S} \end{aligned}$$

This is exactly the ODEs we have derived as limiting process of Markov Chain model.

If we continue this process, we could derive ODEs for second-order moments by taking second order partial derivatives of θ_1 and θ_2 respectively on Equation (A.2) and setting $\theta_1 = \theta_2 = 0$:

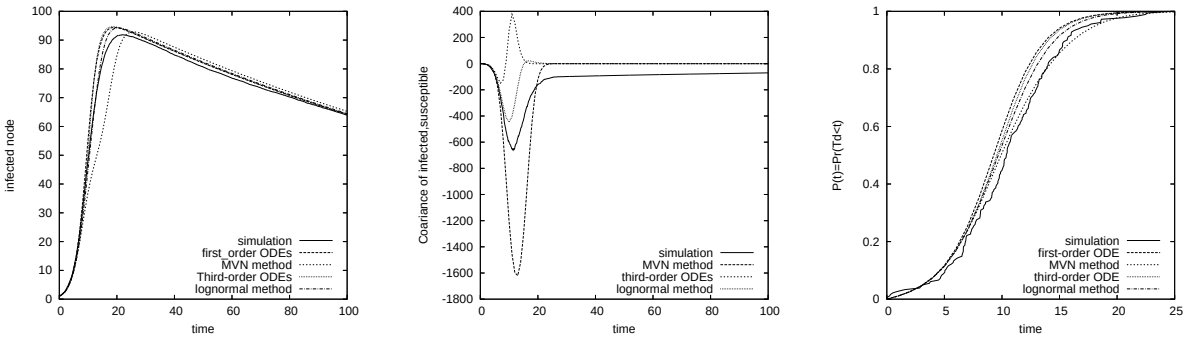
$$\begin{aligned} \frac{dV_S}{dt} &= \gamma(\bar{I}\bar{S} + C_{IS}) - 2\gamma(T_{SSI} + V_S\bar{I} + \bar{S}C_{IS}) \\ \frac{dV_I}{dt} &= \beta\bar{I} - 2\beta V_I + \gamma(C_{IS} + \bar{I}\bar{S}) + 2\gamma(T_{SII} + C_{IS}\bar{I} + \bar{S}V_I) \\ \frac{dC_{IS}}{dt} &= -\beta C_{IS} - \gamma(C_{IS} + \bar{I}\bar{S}) - \gamma T_{SII} - \gamma C_{IS}\bar{I} - \gamma\bar{S}V_I + \gamma T_{SSI} + \gamma V_S\bar{I} + \gamma\bar{S}C_{IS} \end{aligned}$$

where $V_s(t) = \mathrm{Var}(S(t)), V_I(t) = \mathrm{Var}(I(t))$, and T_{SII}, T_{SSI} are the third central moments: $T_{SII} = \mathrm{E}[(S - ES)(I - EI)^2], T_{SSI} = \mathrm{E}[(S - ES)^2(I - EI)]$.

One could keep on this procedure to derive ODEs for the third and higher moments, but eventually a moment closure technique is needed to truncate the equations at certain order. We experiment with three different methods [67, 97, 83].

- MVN (Multi-Variate Normal) method: setting third central moments to zero. This is equivalent to assuming a multi-variate normal distribution of the state variables $(S(t), I(t))$.

- Lognormal method: if we assume a lognormal distribution for the state variables, then the third moments can be expressed in terms of the lower moments



(a) $I(t)$ (b) $COV(I(t), S(t))$ (c) $P(t)$

Figure A.2. Comparison of different moment equations for the case $p = 1.0$

- Third-order moment: truncate the equations by setting fourth-order moments to zero.

In order to compare the performance of these different methods, we simulate probabilistic forwarding, varying p in the range between 0.001 to 1.0, with $N = 100$, and compare the model predictions with the simulation results.

For the basic epidemic routing, i.e., $p = 1.0, \gamma = \beta$ case, Figure A.2 plots the average infected node number, the covariance of infected node number of susceptible node, and the CDF of delay, comparing simulation results with the prediction of different moment equations. We observe that third-order ODEs gives similar result as first-order ODEs, with slight improved match with simulation results. Like first and third order ODEs, lognormal equations under estimates the covariance, and therefore over predicts the infection spreading process, and under predicts the delivery delay. On the other hand, MVN method over estimates the covariance, and under estimates the spread of the infection (as Figure A.2.(a) shows). For this case, MVN method performs best in prediction of delivery delay as shown also in the Figure 2.1 in Section 2.2.

However, MVN method has a drawback. For P in the range $[0.01, 0.3]$, the MVN ODEs have no stable equilibrium, i.e., the solution diverges. [83] observed this draw-

back of MVN method (under a different model), and attributed it to the large variability under the scenario considered.

APPENDIX B

DERIVATION OF DELAY ASYMPTOTIC RESULTS

Here we are going to derive the different bounds and asymptotic values we presented in the paper. For each of the following forwarding schemes, closed-form expressions can be derived for the number of infected nodes $I(t)$ and the cumulative distribution of delay $P(t) = Pr(T_d < t) = 1 - Q(t)$. The expected delay can be evaluated as $\mathrm{E}[T_d] = \int_0^\infty Q(t)dt$, so we are going to show how this integral can be approximated for the different schemes.

- 2-hop forwarding (Section 2.3.1.1)

  The expected delay is equal to:

$$\mathrm{E}[T_d] = \frac{1}{\beta}\int_0^\infty e^{-t}e^{(N-1)(1-t-e^{-t})}dt$$

  $e^{(N-1)(1-t-e^{-t})}$ has a single maximum for $t = 0$, hence according to the saddle point approximation when $N \to \infty$ we can consider:

$$e^{-t}e^{(N-1)(1-t-e^{-t})} \approx e^{-0}e^{-(N-1)t^2/2}$$

  hence

$$\mathrm{E}[T_d] \approx \frac{1}{\beta}\int_0^\infty e^{-(N-1)t^2/2}dt = \frac{1}{\beta}\sqrt{\frac{\pi}{2}}\frac{1}{\sqrt{N-1}}$$

- Probabilistic routing (Section 2.3.1.3). In this case

$$Q(t) = \left(\frac{N}{e^{N\beta pt} + N - 1} \right)^{\frac{1}{p}}.$$

This expression can be easily bounded:

$$\frac{N}{e^{N\beta t} + N - 1} \leq Q(t) \leq \frac{N}{e^{N\beta pt} + N - 1}.$$

Note that these bounds correspond to the comparison of the probabilistic forwarding with epidemic routing with inter-meeting rates of β and βp respectively: probabilistic forwarding is slower than the first one, but faster than the second one.

If we integrate the previous inequality, we get:

$$\frac{\ln(N)}{\beta(N-1)} \leq \mathrm{E}[T_d] \leq \frac{\ln(N)}{\beta p(N-1)}$$

- Limited-time scheme with reinfection (Section 2.3.1.3) In this case:

$$Q(t) = \frac{(a_2 - a_1)e^{-a_1\beta t}}{(a_2 - 1) + (1 - a1)e^{(a2-a1)\beta t}},$$

where a_2 and a_1 are respectively the positive and the negative solution of the equation $\beta I(N - I) - \mu(I - 1) = 0$ (to be solved for I), obtained by imposing $\frac{dI}{dt} = 0$.

We consider three different asymptotic values: for $N \to \infty$, for $\mu \to \infty$ and for $N = \frac{\mu}{\beta} \to \infty$.

As regards the first bound, we proceeded in the following way: we considered a function $Q_{a,N}(t) > 0$ which approximates $Q_N(t)$ (we have stressed the dependence from N), and for which we can closely evaluate $\int_0^\infty Q_{a,N}(t)dt$. This is an asymptotic value for the expected delay if:

$$\lim_{N\to\infty} \frac{\int_0^\infty Q_N(t)dt - \int_0^\infty Q_{a,N}(t)dt}{\int_0^\infty Q_{a,N}(t)dt} \to 0$$

In order to prove it, we proved that $\frac{Q_N(t)-Q_{a,N}(t)}{Q_{a,N}(t)}$ converges uniformly to zero in $\mathbf{R}^+$ as N diverges:

$$\frac{Q_N(t) - Q_{a,N}(t)}{Q_{a,N}(t)} \xrightarrow[N,\infty]{u} 0.$$

In fact in this case $\forall \epsilon > 0$, $\exists\, n_\epsilon \in \mathbf{N}$ such that $\forall t \in \mathbf{R}^+$ and $\forall N > n_\epsilon$

$$\frac{|Q_N(t) - Q_{a,N}(t)|}{|Q_{a,N}(t)|} < \epsilon$$

hence:

$$\frac{|\int_0^\infty Q_N(t) - Q_{a,N}(t)dt|}{|\int_0^\infty Q_{a,N}(t)|} \leq \epsilon.$$

The asymptotical behavior of a_2 and a_1 as $N \to \infty$ ($\lim_{N\to\infty} a_2 = +\infty$, $\lim_{N\to\infty} a_1 = 0$) suggests to consider:

$$Q_{a,N}(t) = \frac{a_2 - a_1}{(a_2 - 1) + (1 - a_1)e^{a_2\beta t}}$$

which can be easily integrated.

$$\left|\frac{Q_N(t) - Q_{a,N}(t)}{Q_{a,N}(t)}\right| = \frac{(1 - e^{a_1\beta t})}{e^{a_1\beta t} + \frac{(1-a_1)}{a_2-1}e^{a_2\beta t}} \leq \frac{(1 - e^{a_1\beta t})}{\frac{(1-a_1)}{a_2-1}e^{a_2\beta t}}$$

We can easily evaluate the maximum of the right expression, and we get:

$$\left|\frac{Q(t)_N - Q_{a,N}(t)}{Q_{a,N}(t)}\right| \leq \frac{-a_1(a_2 - 1)}{(1 - a_1)(a_2 - a_1)} \left(\frac{a_2}{a_2 - a_1}\right)^{-\frac{a_2}{a_1}}$$

The maximum converges to 0 when N diverges, hence the convergence is uniform.

The asymptotic value is:

$$\int_0^\infty Q_{a,N}(t)dt = 1/\beta \frac{a_2 - a_1}{(a_2 - 1)a_2} \ln\left(\frac{a2 - a1}{1 - a_1}\right)$$

which behaves asymptotically as:

$$\frac{1}{\beta}\frac{\ln(N - \frac{\mu}{\beta})}{N - \frac{\mu}{\beta}}$$

In the same way we have found the second bound as $\mu \to \infty$. In this case $\lim_{\mu\to\infty} a_2 = 1$, $\lim_{\mu\to\infty} a_1 = -\infty$, and we consider

$$Q_{a,\mu}(t) = e^{-a_2\beta t}.$$

$$\left|\frac{Q_N(t) - Q_{a,N}(t)}{Q_{a,N}(t)}\right| = \frac{1}{\frac{a_2 - a_1}{(a_2-1)(1-e^{-(a2-a1)\beta t})} - 1}$$

The supremum is achieved for $t \to \infty$ and is equal to:

$$\frac{a_2 - 1}{1 - a_1}$$

which converges to 0 as μ diverges.

The asymptotic value is:

$$\int_0^\infty Q_{a,\mu}(t)dt = \frac{1}{\beta a_2} \underset{\mu\to\infty}{\sim} \frac{\mu - N\beta}{\beta\mu}$$

Finally, as regards the third bound, a closed-form expression can be found for $\mathrm{E}[T_d]$, considering $N = \mu/\beta$:

$$\mathrm{E}[T_d] = \frac{2\,\mathrm{arccot}\sqrt{\frac{\sqrt{N-1}+1}{\sqrt{N-1}-1}}}{\beta\sqrt{N-2}},$$

and

$$\mathrm{E}[T_d] \underset{N\to\infty}{\sim} \frac{\pi}{2\beta\sqrt{N-2}}.$$

APPENDIX C

EM ALGORITHM FOR MEAN-RESTRICTED MIXTURE NORMAL MODEL

In this section, we outline the EM algorithm for the mean-restricted mixture normal model Eq.(4.2) proposed in Section 4.5.3.

Suppose that we have N fully observed inter-contact times: $x_i, i = 1, 2, ..., N$. The following EM algorithm is used to iteratively estimate μ, σ^2, and $w_l, \forall l \in 1, 2, ..., G$ from these fully observed inter-contact times:

$$w_{t+1,l} = (1/N) \sum_{i=1}^{N} p(l|x_i, \Theta_t) \tag{C.1}$$

$$\mu_{t+1} = \frac{\sum_{l=1}^{G} l \sum_{i=1}^{N} x_i p(l|x_i, \Theta_t)}{\sum_{l=1}^{G} l^2 \sum_{i=1}^{N} p(l|x_i, \Theta_t)} \tag{C.2}$$

$$(\sigma^2)_{t+1} = \frac{\sum_{l=1}^{G} \sum_{i=1}^{N} (x_i - l\mu_{t+1})^2 p(l|x_i, \Theta_t)}{\sum_{l=1}^{G} \sum_{i=1}^{n} p(l|x_i, \Theta_t)} \tag{C.3}$$

where $t = 1, 2, ...$ is the iterative step, $\Theta_t = \{\mu_t, (\sigma^2)_t, w_{l,t}, l = 1, 2, ..., G\}$ is the current estimate of the parameters, and $p(l|x_i, \Theta_t)$ is the probability that the random sample x_i comes from component l given the model parameter vector Θ_t. By Bayes' Rule, we have:

$$p(l|x_i, \Theta_t) = \frac{p(l, x_i|\Theta_t)}{p(x_i|\Theta_t)} = \frac{p_t^{l-1}(1 - p_t) f_N(x_i, lu_t, \sigma_t^2)}{\Sigma_{j=1}^{\infty} p_t^{j-1}(1 - p_t) f_N(x_i, j\mu_t, \sigma_t^2)},$$

here the function $f_N(x, \mu, \sigma^2)$ represents the PDF of normal distribution with mean μ and variance σ^2.

APPENDIX D

EM ALGORITHM FOR ONE-BASE-MEAN MODEL

In this section, we first outline the derivation of Expectation-Maximization algorithm for the One-Base-Mean model (Eq.(4.3) in Section 4.5.4), and then give the EM algorithm for model given by Eq.(4.4) and Two-Base-Mean model.

One-Base-Mean Model. We start by outlineing the derivation of EM algorithm for the following model:

$$f_{GEO\_1P\_1BM}(x) = \sum_{i=1}^{\infty} p^{i-1}(1-p) f_N(x|i\mu, \sigma^2).$$

Our goal is to derive the maximum likelihood estimate for the model parameters $\Theta = (p, \mu, sigma^2)$, based all observations of the inter-contact times. Assume that we have N fully observed inter-contact time, x_i, $(i = 1, ..., N)$, N_s start-censored inter-contact time observations, S_i,$(i = 1, ..., N_s)$, N_e end-censored inter-contact time observations, E_i,$(i = 1, ..., N_e)$, and N_n no-meeting observations, N_i,$(i = 1, ..., N_n)$. We assume that the fully inter-contact time x_i's are independent and identically distributed (i.i.d.) with PDF given by above model, and the distributions of the censored observations relate to the above model as to our discussion in Section 4.5.4. We denote the whole data set as:

$$X = (x_1, ..., x_N, S_1, ..., S_{N_s}, E_1, ..., E_{N_e}, N_1, ..., N_{N_n}).$$

Due to the complex form of the model, we resort to EM algorithm to obtain the maximum likelihood estimate.

First, we introduce hidden variables for each observations (i.e., samples) of the inter-contact time, representing the number of physical meetings within that inter-contact time. We denote the whole set of hidden variables as Y.

Next, in the Expectation-step, we derive $p(l|x_i, \Theta_t)$ ($p(l|S_i, \Theta_t)$, $p(l|E_i, \Theta_t)$, $p(l|N_i, \Theta_t)$), i.e., the distribution of the hidden variable (the number of physical meetings within the inter-contact time), given the fully observed data (started-censored data, end-censored data, no-meeting observations) and the current estimates of model parameters, $\Theta_t = (p_t, \mu_t, \sigma_t^2)$.

(1) **Fully observed inter-contact times**. By Bayes' Rule, we have:

$$\begin{aligned} p(l|x_i, \Theta_t) &= \frac{p(l, x_i|\Theta_t)}{p(x_i|\Theta_t)} \\ &= \frac{p_t^{l-1}(1-p_t) f_N(x_i, lu_t, \sigma_t^2)}{\Sigma_{j=1}^{\infty} p_t^{j-1}(1-p_t) f_N(x_i, j\mu_t, \sigma_t^2)} \end{aligned}$$

for $i = 1, ..., N$, $l = 1, 2, ...$

(2) **Start-censored observations**. As discussed in Section 4.5.4, we assume such observation is the residual lifetime, and its PDF is given by $f_Y(x) = \frac{1-F_X(x)}{EX}$, where $F_X(x), EX$ is the CDF and mean of the inter-contact time respectively. We have:

$$g_l(S_i) := p(S_i|l, \Theta_t) = \int_{S_i}^{\infty} f_N(x, l\mu_t, \sigma_t^2) dx / l\mu_t. \tag{D.1}$$

The conditional distribution for the number of physical meetings given the start-censored observation is given by:

$$\begin{aligned} p(l|S_i, \Theta_t) &= \frac{p(l, S_i|\Theta_t)}{p(S_i|\Theta_t)} \\ &= \frac{\int_{S_i}^{\infty} p_t^{l-1}(1-p_t) f_N(x, l\mu_t, \sigma_t^2) dx / l\mu_t}{\sum_{j=1}^{\infty} p_t^{j-1}(1-p_t) \int_{S_i}^{\infty} f_N(x, j\mu_t, \sigma_t^2) dx / j\mu_t} \end{aligned}$$

for $S_i, i = 1, ..., N_s, l = 1, 2, ...$

(3) **No-meeting observations**. Assume the CDF corresponding to the PDF $g_l(x)$, i.e., Eq.(D.1), is $G_l(x)$, i.e., $G_l(x) = \int_{-\infty}^{x} g_l(y)dy$. We have

$$p(l|N_i, \Theta_t) = \frac{p(l, N_i|\Theta_t)}{p(N_i|\Theta_t)} = \frac{p_t^{l-1}(1-p_t)(1-G_l(N_i))}{\sum_{j=1}^{\infty} p_t^{j-1}(1-p_t)(1-G_j(N_i))},$$

for $i = 1, ..., N_n, l = 1, 2, ...$

(4) **End-censored observations**. We have

$$\begin{aligned} p(l|E_i, \Theta_t) &= \frac{p(l, E_i|\Theta_t)}{p(E_i|\Theta_t)} \\ &= \frac{\int_{E_i}^{\infty} p_t^{l-1}(1-p_t) f_N(x, l\mu_t, \sigma_t^2)dx}{\sum_{j=1}^{\infty} p_t^{j-1}(1-p_t) \int_{E_i}^{\infty} f_N(x, j\mu_t, \sigma_t^2)dx} \end{aligned}$$

for $i = 1, ..., Ne, l = 1, 2, ...$

Last, in the Maximization-step, we derive the expectation of log complete data likelihood function, conditioned on the observations and the current estimates of model parameters, as follows:

$$Q(\Theta, \Theta_t) := E[log(P(X, Y|\Theta)|X, \Theta_t],$$

and obtain new estimates for the model parameters by setting:

$$\Theta_{t+1} = \text{argmax}_{\Theta} Q(\Theta, \Theta_t).$$

Omitting the derivations (which is similar to that in [14]), we get the following updating rule:

$$p_{t+1} = num_p/denom_p$$

$$\begin{aligned}
\mu_{t+1} &= \frac{\sum_{i=1}^{N}\sum_{l=1}^{\infty} x_i l p(l|x_i,\Theta_t)}{\sum_{i=1}^{N}\sum_{l=1}^{\infty} l^2 p(l|x_i,\Theta_t)} \\
(\sigma^2)_{t+1} &= \frac{\sum_{i=1}^{N}\sum_{l=1}^{\infty}(x_i - l\mu_{t+1})^2 p(l|x_i,\Theta^t)}{\sum_{i=1}^{N}\sum_{l=1}^{\infty} p(l|x_i,\Theta_t)}
\end{aligned}$$

where $t = 1, 2, ..$ is the iterative step, and

$$\begin{aligned}
num_p &= \sum_{i=1}^{N}\sum_{l=1}^{\infty}(l-1)p(l|x_i,\Theta_t) + \sum_{i=1}^{N_s}\sum_{l=1}^{\infty}(l-1)p(l|S_i,\Theta_t) \\
&\quad + \sum_{i=1}^{N_e}\sum_{l=1}^{\infty}(l-1)p(l|E_i,\Theta_t) + \sum_{i=1}^{N_n}\sum_{l=1}^{\infty}(l-1)p(l|N_i,\Theta_t) \\
denom_p &= \sum_{i=1}^{N}\sum_{l=1}^{\infty} lp(l|x_i,\Theta_t) + \sum_{i=1}^{Ns}\sum_{l=1}^{\infty} lp(l|S_i,\Theta_t) \\
&\quad + \sum_{i=1}^{Ne}\sum_{l=1}^{\infty} lp(l|E_i,\Theta_t) + \sum_{i=1}^{Nn}\sum_{l=1}^{\infty} lp(l|N_i,\Theta_t)
\end{aligned}$$

Note that we assume the censored observations only affect the estimate of p, and ignored them while updating μ, σ^2.

One-Base-Mean-Multi-FailureProb Model. Here we outline the EM algorithm for the model Eq.(4.4) as follows:

$$f_{GEO\_MP\_1BM}(x) = \sum_{i=1}^{C} w_i \sum_{l=1}^{\infty} p_i^{l-1}(1-p_i) f_N(x|l\mu,\sigma^2).$$

To derive EM algorithm for this model, we consider the following two hidden variables for each data samples. One hidden variable, denoted as $c, 1 \le c \le C$, designates the failure probability (p_c) governed the observed inter-contact time; another hidden variable, denoted as $l, 1 \le l \le \infty$, is the number of physical inter-meeting times within the observed sample.

We have the following updating rules, here $\Theta = (w_1, ..., w_C, p_1, ..., p_C, \mu, \sigma^2)$.

$$(w_j)_{t+1} = \frac{1}{N}\sum_{i=1}^{N} p(c=j|x_i,\Theta_t) + \frac{1}{N_s}\sum_{i=1}^{N_s} p(c=j|S_i,\Theta_t)$$

$$
\begin{aligned}
& + \frac{1}{N_e} \sum_{i=1}^{N_s} p(c = j | E_i, \Theta_t) + \frac{1}{N_n} \sum_{i=1}^{N_s} p(c = j | N_i, \Theta_t) \\
\mu_{t+1} &= \frac{\sum_{i=1}^{N} \sum_{l=1}^{\infty} x_i l p(l | x_i, \Theta_t)}{\sum_{i=1}^{N} \sum_{l=1}^{\infty} l^2 p(l | x_i, \Theta_t)} \\
(\sigma^2)_{t+1} &= \frac{\sum_{i=1}^{N} \sum_{l=1}^{\infty} (x_i - l\mu_{t+1})^2 p(l | x_i, \Theta^t)}{\sum_{i=1}^{N} \sum_{l=1}^{\infty} p(l | x_i, \Theta_t)} \\
(p_j)_{t+1} &= num_{p_j} / denom_{p_j}
\end{aligned}
$$

where $j = 1, ..., C$. The updating rules for $p_j, j = 1, ..., C$, μ and σ^2 are the same as those of the previous model, with the only difference is the evaluation of the conditional distribution of hidden variables given the data and the current estimates of model parameters. By Bay's rule, we have:

$$
\begin{aligned}
p(c, l | x_i, \Theta_t) &= \frac{p(x_i, c, l | \Theta_t)}{p(x_i | \Theta_t)} \\
&= \frac{(w_c)_t (p_c)_t^{l-1} (1 - (p_c)_t) f_N(x_i, l\mu_t, \sigma_t^2)}{\sum_{j=1}^{2} \sum_{k=1}^{\infty} (w_j)_t (p_j)_t^{k-1} (1 - (p_j)_t) f_N(x_i | jk\mu_t, \sigma_t^2)},
\end{aligned}
$$

and

$$
p(c | x_i, \Theta_t) = \sum_{l=1}^{\infty} p(c, l | x_i, \Theta_t),
$$

$$
p(l | x_i, \Theta_t) = \sum_{c=1}^{C} p(c, l | x_i, \Theta_t).
$$

We have:

$$
\begin{aligned}
num_{p_j} &= \sum_{i=1}^{N} \sum_{l=1}^{\infty} (l-1) p(c = j, l | x_i, \Theta_t) + \sum_{i=1}^{N_s} \sum_{l=1}^{\infty} (l-1) p(c = j, l | S_i, \Theta_t) \\
&\quad + \sum_{i=1}^{N_e} \sum_{l=1}^{\infty} (l-1) p(c = j, l | E_i, \Theta_t) + \sum_{i=1}^{N_n} \sum_{l=1}^{\infty} (l-1) p(c = j, l | N_i, \Theta_t) \\
denom_{p_j} &= \sum_{i=1}^{N} \sum_{l=1}^{\infty} l p(c = j, l | x_i, \Theta_t) + \sum_{i=1}^{Ns} \sum_{l=1}^{\infty} l p(c = j, l | S_i, \Theta_t)
\end{aligned}
$$

$$+\sum_{i=1}^{Ne}\sum_{l=1}^{\infty} lp(c=j,l|E_i,\Theta_t)+\sum_{i=1}^{Nn}\sum_{l=1}^{\infty} lp(c=j,l|N_i,\Theta_t)$$

Similar to the One-Base-Mean model, we derive the conditional distribution of the hidden variables given the censored observations.

Two-Base-Mean Model. We now briefly describe the EM algorithm used for the following model:

$$f_{GEO\_2BM}(x)=\sum_{i=1}^{2} w_i \sum_{l=1}^{\infty} p_i^{l-1}(1-p_i) f_N(x|li\mu,\sigma^2).$$

Here we have $\Theta=(w_1,w_2,p_1,p_2,\mu,\sigma^2)$. The derivation of the EM algorithm is similar to the One-base-mean model, except that for this model, we introduce two random variables for each data sample (i.e., observation): one, c denotes the class of the inter-contact time, another, l, denotes the number of physical inter-meeting times within the observed inter-contact time. We evaluate the conditional distribution of the hidden variables as follows:

$$\begin{aligned} p(c,l|x_i,\Theta_t) &= \frac{p(x_i,c,l|\Theta_t)}{p(x_i|\Theta_t)} \\ &= \frac{(w_c)_t (p_c)_t^{l-1}(1-(p_c)_t) f_N(x_i,l\mu_t,\sigma_t^2)}{\sum_{j=1}^{2}\sum_{k=1}^{\infty}(w_j)_t (p_j)_t^{k-1}(1-(p_j)_t) f_N(x_i|jk\mu_t,\sigma_t^2)} \end{aligned}$$

Similarly, we evaluate the conditional distrbution of the hidden variables for the censored observations.

The updating rules are as follows:

$$\begin{aligned} (w_j)_t &= \sum_{l=1}^{\infty}(\sum_{i=1}^{N} p(j,l|x_i,\Theta_t)+\sum_{i=1}^{N_s} p(j,l|S_i,\Theta_t)+\sum_{i=i}^{N_e} p(j,l|E_i,\Theta_t)+\sum_{i=1}^{N_n} p(j,l|N_i,\Theta_t), \\ &\quad \text{for } j=1,...,C \\ (p_j)_t &= \frac{\sum_{l=1}^{\infty}\sum_{i=1}^{N}(l-1)p(j,l|x_i,\Theta_t)}{\sum_{1=1}^{\infty}\sum_{i=1}^{N} lp(j,l|x_i,\Theta_t)}, \text{ for } j=1,...,C \end{aligned}$$

$$\mu_t = \frac{\sum_{j=1}^{C}\sum_{l=1}^{\infty}\sum_{i=1}^{N} ljx_i p(j,l|x_i,\Theta_t)}{\sum_{j=1}^{C}\sum_{l=1}^{\infty}\sum_{i=1}^{N} l^2j^2 p(j,l|x_i,\Theta_t)}$$

$$\sigma_t^2 = \frac{\sum_{j=1}^{C}\sum_{l=1}^{\infty}\sum_{i=1}^{N} (x_i - cl\mu_t)^2 p(j,l|x_i,\Theta_t)}{\sum_{j=1}^{C}\sum_{l=1}^{\infty}\sum_{i=1}^{N} p(j,l|x_i,\Theta_t)}$$

BIBLIOGRAPHY

[1] Delay Tolerant Networking Research Group. http://www.dtnrg.org/.

[2] First mile solutions (i.e., daknet). http://firstmilesolutions.com/.

[3] Reality Mining Project, Human Dynamics Group at the MIT Media Lab. http://reality.media.mit.edu.

[4] Traffic Modeling and Resource Allocation in Call Centers. http://www.easyerlang.com/papers/Traffic_Modeling.htm.

[5] UCSD wireless topology discovery project. http://sysnet.ucsd.edu/wtd/.

[6] Wizzy digital courier. http://www.wizzy.org.za/.

[7] Acedanski, S., Deb, S., Medard, M., and Koetter, R. How Good is Random Linear Coding based Distributed Networked Storage. In *IEEE Workshop on Network Coding, Theory, and Applications (NETCOD)* (2005).

[8] Ahlswede, R., Cai, N., Li, S.-Y. R., and Yeung, R. W. Network information flow. *IEEE Trans. on Information Theory 46*, 1204–1216.

[9] Aldous, D., and Fill, J. Reversible Markov Chains and Random Walks on Graphs. (monograph in preparation), http://statwww.berkeley.edu/users/aldous/RWG/book.html.

[10] Bai, F., Sadagopan, N., Krishnamachari, B., and Helmy, A. Modeling Path Duration Distributions in MANETs and their Impact on Routing Performance. In *IEEE Journal on Selected Areas of Communications* (September 2004).

[11] Bailey, Norman T.J. *The Mathematical Theory of Infectious Diseases*. Charles Griffin & Company LTD, 1975.

[12] Balasubramanian, A., Levine, B. N., and Venkataramani, A. DTN Routing as a Resource Allocation Problem. In *ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM)* (2007).

[13] Bettstetter, C. Mobility modeling in wireless networks: categorization, smooth movement, and border effects. In *ACM SIGMOBILE Mobile Computing and Communications Review* (2001), vol. 5, Issue 3, July.

[14] Bilmes, J. A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Tech. Rep. ICSI-TR-97-021, University of Berkeley, 1997.

[15] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. Complex Networks: Structure and Dynamics. *Physics Reports 424* (2006), 175–308.

[16] Broder, A., and Mitzenmacher, M. Using Multiple Hash Functions to Improve IP Lookups. In *IEEE International Conference on Computer Communications (INFOCOM)* (2001).

[17] Burgess, J., Bissias, G., Corner, M., and Levine, B.N. Surviving Attacks on Disruption-Tolerant Networks without Authentication. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)* (2007).

[18] Burgess, J., Gallagher, B., Jensen, D., and Levine, B. N. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *IEEE International Conference on Computer Communications (INFOCOM)* (2006).

[19] Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., and Scott, K. Delay-Tolerant Networking: an Approach to Interplanetary Internet. In *IEEE Communications Magazine* (2003).

[20] Camp, T., Boleng, J., and Davies, V. A Survey of Mobility Models for Ad Hoc Network Research. In *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Network Research: Research, Trends and Applications* (2002).

[21] Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., and Scott, J. Impact of Human Mobility on the Design of Opportunitic Forwarding Algorithms. In *IEEE International Conference on Computer Communications (INFOCOM)* (2006).

[22] Chen, C., and Murphy, A.L. Enabling Disconnected Transitive Communication in Mobile Ad Hoc Networks. In *Proc. of the Workshop on Principles of Mobile Computing (POMC)* (2001).

[23] Chen, L.-J., Chen, Y.-C., Sun, T., Sreedevi, P., Chen, K.-T., Yu, C.-H., and Chu, H.-H. Finding Self-Similarities in Opportunistic People Networks. In *IEEE International Conference on Computer Communications (INFOCOM) Mini-Symposium* (2007).

[24] Chen, L-J, Yu, C-H, Sun, T, Chen, Y-C, and Chu, H-H. A Hybrid Routing Approach for Opportunistic Networks. In *ACM SIGCOMM Workshop on Challenged Networks (CHANTS)* (2006).

[25] Chou, P. A., Wu, Y., and Jain, K. Practical network coding. *Allerton Conference on Communication, Control, and Computing* (2003).

[26] Conan, V., Leguay, J., and Friedman, T. Heterogeneous Inter-contact Times: their Importance for DTN Routing, 2006. preprint.

[27] Daley, D.J., and Gani, J. *Epidemic Modelling*. Cambridge University Press, 1999.

[28] Deb, S., Choute, C., Medard, M., and Koetter, R. Data Harvesting: A Random Coding Approach to Rapid Dissemination and Efficient Storage of Data. Tech. rep., M.I.T. LIDS Technical Report, 2004.

[29] Deb, S., Medard, M., and Choute, C. Algebraic Gossip: A Network Coding Approach to Optimal Multiple Rumor Mongering. *IEEE/ACM Transactions on Networking, special issue on networking and information theory* (2006), 2486–2507.

[30] Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., and Terry, D. Epidemic Algorithms for Replicated Database Maintenance. In *ACM Symposium on Principles of Distributed Computing* (1987).

[31] Doria, A., Ud'en, M., and Pandey, D. P. Providing Connectivity to the Saami Normadic Community. In *International Conference on Open Collaborative Design for Sustainable Innovation (dyd02)* (Dec 2002).

[32] Dunbabin, M., Corke, P., Vailescu, I., and Rus, D. Data Muling over Underwater Wireless Sensor Networks using an Autonomous Underwater vehicle. In *International Conference on Robotics and Automation (ICRA)* (May 2006), IEEE.

[33] Eagle, N., and Pentland, A. Reality Mining: Sensing Complex Social Systems. In *Journal of Personal and Ubiquitous Computing* (2005).

[34] Fawal, A. E., Boudec, J.-Y. L., and Salamatian, K. Performance Analysis of Self-Limiting Epidemic Forwarding. Tech. rep.

[35] Feller, W. *An Introduction to Probability Theory and Its Applications*, vol. 1,2. J. Wiley and Sons, New York, 1964.

[36] Fragouli, C., Boudec, J.-Y. Le, and Widmer, J. Network Coding: An Instant Primer. *ACM SIGCOMM Computer Communication Review 36* (2006).

[37] Gkantsidis, C., and Rodriguez, P. Network Coding for large scale Content Distribution. *IEEE International Conference on Computer Communications (INFOCOM)* (2005).

[38] Groenevelt, R., Nain, P., and Koole, G. Message Delay in Manet. Tech. Rep. 5372, INRIA, November 2004.

[39] Groenevelt, R., Nain, P., and Koole, G. The Message Delay in Mobile Ad Hoc Networks. In *Performance* (October 2005).

[40] Grossglauser, M., and Tse, D. N. C. Mobility increases the capacity of ad-hoc wireless networks. In *IEEE International Conference on Computer Communications (INFOCOM)* (2001).

[41] Gupta, P., and Kumar, P. R. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory 46* (2000), 388–404.

[42] Gurin, R. Channel Occupancy Time Distribution in a Cellular Radio System. In *IEEE Transactions on Vehicular Technology* (1987), vol. Vol. vt-36, No. 3, pp. 89-99.

[43] Haas, Z. J., and Small, T. A New Networking Model for Biological Applications of Ad Hoc Sensor Networks. *IEEE/ACM Transactions on Networking* (February 2006).

[44] Hanbali1, A. A., Nain, P., and Altman, E. Performance Evaluation of Packet Relaying in Ad Hoc Networks. Tech. Rep. 5860, INRIA, 2006.

[45] Henderson, T., Kotz, D., and Abyzov, I. The Changing Usage of a Mature Campus-wide Wireless Network. In *Proc. MobiCom* (2004).

[46] Ho, T., Koetter, R., Medard, M., Karger, D.R., and Effros, M. The Benefits of Coding Over Routing in a Randomized Setting. In *IEEE International Symposium on Information Theory (ISIT)* (2003).

[47] Hsu, W.-J., and Helmy, A. IMPACT: Investigation of Mobile-user Patterns Across University Campus using WLAN Trace Analysis. Tech. rep., University of South California, 2005.

[48] Hsu, W.-J., Spyropoulos, T., Psounis, K., and Helmy, A. Modeling Time-variant User Mobility in Wireless Mobile Networks. In *IEEE International Conference on Computer Communications (INFOCOM)* (2007).

[49] Hsu, W.J., and Helmy, Ahmed. Encounter-based message broadcasting in ad hoc networks with intermittent connectivity. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), poster session* (2005).

[50] Hui, P., Chaintreau, A., Gass, R., Scott, J., Crowcroft, J., and Diot, C. Pocket Switched Networking: Challenges, Feasibility, and Implementation Issues. In *IFIP TC6 International Workshop on Autonomic Communication (WAC)* (2005).

[51] Hui, P., and Crowcroft, J. Bubble Rap: Forwarding in small world DTNs in ever decreasing circles. Tech. Rep. UCAM-CL-TR-684, University of Cambridge, Computer Laboratory, May 2007.

[52] Hui, P., Yoneki, E., Chan, S-Y, and Crowcroft, J. Distributed Community Detection in Delay Tolerant Networks. In *MobiArch (International Workshop on Mobility in the Evolving Internet Architecture)* (2007).

[53] Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A. K., Shih, E., Balakrishnan, H., and Madden, S. CarTel: A Distributed Mobile Sensor Computing System. In *ACM SenSys (Conference on Embedded Networked Sensor Systems)* (November 2006).

[54] Jain, R., Lelescu, D., and Balakrishnan, M. Model T: An Empirical Model for User Registration Patterns in a Compus Wireless LAN. In *MobiCom* (2005).

[55] Jain, Ravi, Shivaprasad, Anupama, He, Xiaoning, and Lelescu, Dan. Towards an empirical model of user mobility and registration patterns. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), poster* (2004).

[56] Jain, S., Demmer, M., Patra, R., and Fall, K. Using Redundancy to Cope with Failures in a Delay Tolerant Network. In *ACM SIGCOMM (Conference on Applications, Technologies, and Protocols for Computer Communication)* (2005).

[57] Jain, S., Fall, K., and Patra, R. Routing in a Delay Tolerant Network. In *ACM Conference on Applications, Technologies, and Protocols for Computer Communication (SIGCOMM)* (2004).

[58] Jetcheva, J., Hu, Y.-C., PalChaudhuri, S., Saha, A. Kumar, and Johnson, D. B. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In *Workshop on Mobile Computing Systems and Applications* (2003).

[59] Jindal, A., and Psounis, K. Fundamental Mobility Properties for Realistic Performance Analysis of Intermittently Connected Mobile Networks.

[60] Jones, E. P.C., and Ward, P. A.S. Routing Strategies for Delay-Tolerant Networks. Submitted to Computer Communication Review.

[61] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L.-S., and Rubenstein, D. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)* (2002).

[62] Jun, H., Ammar, M., Corner, M., and Zegura, E. Hierarchical Power Management in Disruption Tolerant Networks with Traffic-Aware Optimization. In *ACM SIGCOMM Workshop on Challenged Networks (CHANTS)* (2006).

[63] Kaplan, E.L., and Meier, P. Nonparametric Estimation from Incomplete Observations. *Journal of the American Statistical Association 53* (Jun 1958).

[64] Karagiannis, T., Boudec, J-Y. Le, and Vojnovic, M. Power Law and Exponential Decay of Inter Contact Times between Mobile Devices. In *ACM Conference on Mobile Computing and Networking (MOBICOM)* (2007).

[65] Karp, R., Schindelhauer, C., Shenker, S., and Vocking, B. Randomized Rumor Spreading. *Proc. Foundations of Computer Science* (2000).

[66] Katti, S., Katabi, D., Hu, W., and Hariharan, R. The Importance of Being Opportunistic: Practical Network Coding For Wireless Environments. In *Allerton Conference on Communication, Control, and Computing* (Sep 2005).

[67] Keeling, M. J. Multiplicative Moments and Measures of Persistence in Ecology. *Journal of Theoretical Biology 205* (2000), 269–281.

[68] Kempe, D., Kleinberg, J., and Kumar, A. Connectivity and Inference Problems for Temporal Networks. In *Journal of Computer and System Sciences, Volume 64,Special issue on STOC 2000* (2002).

[69] Kephart, J.O., and White, S. R. Directed-graph epidemiological models of computer viruses. In *IEEE Computer Society Symposium on Research in Security and Privacy* (1991).

[70] Kephart, J.O., and White, S.R. Measuring and Modeling Computer Virus Prevalence. In *IEEE Computer Society Symposium on research in security and privacy* (1993).

[71] Khelil, A., Becker, C., Tian, J., and Rothermel, K. An epidemic model for information diffusion in manets. In *International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)* (2002).

[72] Kim, M., Kotz, D., and Kim, S. Extracting a Mobility Model from Real User Traces. In *IEEE International Conference on Computer Communications (INFOCOM)* (2006).

[73] Kleinrock, Leonard. *Queueing Systems, Volume 1: Theory.* John Wiley & Sons, 1975.

[74] Krumm, J., and Horvitz, E. The Microsoft Multiperson Location Survey. Tech. Rep. MSR-TR-2005-13, Microsoft Research Technical Report, August 2005.

[75] Kurtz, T. G. Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes. *Journal of Applied Probabilities* (1970), 49–58.

[76] Leguay, J., Friedman, T., and Conan, V. Evaluating Mobility Pattern Space Routing for DTNs. In *IEEE International Conference on Computer Communications (INFOCOM)* (2006).

[77] Li, Q., and Rus, D. Communication in Disconnected Ad Hoc Networks Using Message Relay. *Journal of Parallel and Distributed Computing* (2005).

[78] Li, Z., and Li, B. Network Coding: The Case of Multiple Unicast Session. In *Allerton Conference on Communication, Control, and Computing* (2004).

[79] Li, Z., Li, B., Jiang, D., and Lau, L-C. On Achieving Optimal Throughput with Network Coding. In *IEEE International Conference on Computer Communications (INFOCOM)* (2005).

[80] Lidl, R, and Niederreiter, H. *Finite Fields, 2nd edition.* Cambridge, England: Cambridge University Press, 1997.

[81] Lin, Y., Liang, B., and Li, B. Performance Modeling of Network Coding in Epidemic Routing. In *Proc. MobiOpp* (2007).

[82] Lindgren, Anders, Doria, Avri, and Schelen, Olov. Probabilistic routing in intermittently connected networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), Poster* (2003).

[83] Lloyd, A. L. Estimating variablity in models for recurrent epidemics: assessing the use of moment closure techniques. *Theoretical Population Biology* (2004).

[84] Luby, M. LT codes. In *IEEE Symposium on Foundations of Computer Science (FOCS)* (2002), pp. 271–282.

[85] Luby, M., Mitzenmacher, M., Shokrollabi, A., and Spielman, D. Efficient Erasure Correcting Codes. *IEEE Transactions on Information Theory* (February 2001).

[86] Lun, D. S., Medard, M., Ho, T., and Koetter, R. Network Coding with a Cost Criterion. In *International Symposium on Information Theory and its Applications (ISITA)* (2004).

[87] Maffei, A., Fall, K., and Chayes, D. Ocean Instrument Internet. In *Proc. ASU Ocean Science Conference* (2006).

[88] Mickens, J. W., and Noble, B.D. Modeling Epidemic Spreading in Mobile Environment. In *ACM Workshop on Wireless Security (WiSE)* (2005).

[89] Musolesi, M., Hailes, S., and Mascolo, C. An Ad Hoc Mobility Model Founded on Social Network Theory. In *Proc. of WSWiM (International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems)* (2004).

[90] Neglia, G., and Zhang, X. Optimal Delay-Power Tradeoff in Sparse Delay Tolerant Networks: a preliminary study. In *ACM SIGCOMM Workshop on Challenged Networks (CHANTS)* (2006).

[91] Papadopouli, M., and Schulzrinne, H. Seven Degrees of Separation in Mobile Ad Hoc Networks. *IEEE Global Communications Conference (GLOBECOM)* (2000).

[92] Partan, J., Kurose, J., and Levine, B. N. A Survey of Practical Issues in Underwater Networks. In *ACM International Workshop on UnderWater Networks (WUWNet)* (2006).

[93] Reed, I.S., and Solomon, G. Polynomial Codes over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics 8* (1960), 300–304.

[94] Saha, A. K., and Johnson, D. B. Modeling Mobility for Vehicular Ad Hoc Networks. In *ACM international workshop on Vehicular ad hoc networks (VANET), poster session* (2004).

[95] Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A. CRAWDAD data set: cambridge/haggle (v. 2006-01-31), Jan. 2006.

[96] Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A. CRAWDAD trace: cambridge/haggle imote/infocom(v. 2006-01-31), Jan. 2006.

[97] Sell, I. N. An Extension of the Moment Closure Method. *Theoretical Population Biology 64* (2003), 233–239.

[98] Seth, A., Kroeker, D., Zaharia, M., Guo, S., and Keshav, S. Low-Cost Communication for Rural Internet Kiosks Using Mechanical Backhaul. In *Proc. ACM MobiCom* (2006).

[99] Sharma, G., and Mazumdar, R. R. Delay and Capacity Tradeoffs for Wireless Ad Hoc Networks with Random Mobility. preprint.

[100] Small, T., and Haas, Z. J. The Shared Wireless Infostation Model - a new ad hoc networking paradigm. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)* (2003).

[101] Small, T., and Haas, Z. J. Resource and performance tradeoffs in delay-tolerant wireless networks. In *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)* (2005).

[102] Spyropoulos, T., Psounis, K., and Raghavendra, C. Efficient Routing in Intermittently Connected Mobile Networks: The Multi-copy Case. In *ACM/IEEE Transactions on Networking* (2007).

[103] Spyropoulos, T., Psounis, K., and Raghavendra, C. Efficient Routing in Intermittently Connected Mobile Networks: The Single-copy Case. In *ACM/IEEE Transactions on Networking* (2007).

[104] Spyropoulos, T., Psounis, K., and Raghavendra, C. S. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)* (2005).

[105] Spyropoulos, T., Psounis, K., and Raghavendra, C. S. Performance Analysis of Mobility-assisted Routing. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing archive (MOBIHOC)* (2006).

[106] Staniford, S., Paxson, V., and Weaver, N. How to Own the Internet in Your Spare Time. *11th Usenix Security Symposium* (2002).

[107] Su, J., Chin, A., Popivanova, A., Goel, A., and de Lara, E. User Mobility for Opportunistic Ad-Hoc Networking. In *IEEE workshop on Mobile Computing Systems and Applications (WMSCA)* (2004).

[108] Su, J., Goel, A., and d. Lara, E. An Empirical Evaluation of the Student-Net Delay Tolerant Network. In *International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS)* (2006).

[109] Tuduce, C., and Gross, T. A Mobility Model Based on WLAN Traces and its Validation. In *IEEE International Conference on Computer Communications (INFOCOM)* (2005).

[110] Vahdat, A., and Becker, D. Epidemic Routing for Partially Connected Ad Hoc Networks. Tech. Rep. CS-200006, Duke University, April 2000.

[111] Wang, Y., Jain, S., Martonosi, M., and Fall, K. Erasure-coding based routing for opportunistic networks. *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)* (2005).

[112] Widmer, J., and Boudec, J.-Y. Le. Network Coding for Efficient Communication in Extreme Networks. *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)* (2005).

[113] Widmer, J., Fragouli, C., and Boudec, J.-Y. Le. Energy-efficient broadcasting in wireless ad-hoc networks. In *NETCOD (IEEE Workshop on Network Coding, Theory, and Applications)* (2005).

[114] Wu, Y., Chou, P. A., and Kung, S.-Y. Minimum-Energy Multicast in Mobile Ad hoc Networks using Network Coding. In *IEEE Information Theory Workshop* (2004).

[115] Y. Wu, P. A. Chou, S.-Y. Kung. Information Exchange in Wireless Networks with Network Coding and Physical-Layer Broadcast. Tech. rep., Microsoft Technical Report, MSR-TR-2004-78, August 2004.

[116] Zhang, X., Neglia, G., Kurose, J., and Towsley, D. On the Benefits of Random Linear Coding for Unicast Applications in Disruption Tolerant Networks. *NETCOD (IEEE Workshop on Network Coding, Theory, and Applications)* (2006).

[117] Zhang, X., Neglia, G., Kurose, J., and Towsley, D. Performance Modeling of Epidemic Routing. *Elsevier Computer Networks journal 51/10* (2007), 2859–2891.

[118] Zhao, W., Ammar, M., and Zegura, E. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)* (2006).

[119] Zhao, W., Chen, Y., Ammar, M., Corner, M. D., Levine, B. N., and Zegura, E. Capacity Enhancement using Throwboxes in DTNs. In *IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS)* (2006).

[120] Zou, C. C., Towsley, D., and Gong, W. On the Performance of Internet Worm Scanning Strategies. *Elsevier Journal of Performance Evaluation* (2006).