The Impact of the Mobility Model on Delay Tolerant Networking Performance Analysis

Muhammad Abdulla and Robert Simon
Department of Computer Science
George Mason University
Fairfax, VA 22030
Email: {mabdulla, simon}@cs.gmu.edu

***Abstract*— Delay tolerant networks (DTNs) are a class of networks that experience frequent and long-duration partitions due to sparse distribution of nodes. The topological impairments experienced within a DTN pose unique challenges for designing effective DTN routing protocols. For an important class of DTNs nodes depend on their mobility to carry message to the destination. It is therefore essential to understand the impact of commonly-used mobility models on performance analysis of DTN routing schemes. This paper shows how the underlying statistical properties of a frequently used mobility model can be used to enhance our understanding of DTN simulation results. Using these results we also analyze several DTN routing techniques, including direct transmission, Spray and Wait, and a novel core-assisted routing schemes.**

I. INTRODUCTION

Delay tolerant networks (DTNs) are a class of emerging systems that experience frequent and long lasting partitions [3]. In DTNs an end-to-end path between the source and the destination may only exist for brief and unpredictable periods of time. With the increased use of wireless mobile devices, many new network applications fall into this category, such as wildlife tracking, military networks, and disaster recovery and emergency response systems.

Although routing in general has been studied extensively for Mobile Ad Hoc networks (MANETs), routing in DTNs is a challenging problem. Since an end-to-end path may not exist, traditional proactive and reactive routing schemes for MANETs fail to work. Proactive routing schemes, where nodes try to keep up to date routing information for other nodes, may fail to converge while producing high number of periodic update packets. In reactive routing schemes, where routing information is obtained on demand, nodes may fail to find a path to the destination. However, this does not mean that the packets cannot be delivered to the destination. Due to node mobility, different links come up and down over time, enabling nodes to achieve *eventual delivery* through a *store-and-forward* approach, which uses buffers to hold the message until the next link comes up in the end-to-end path due to node mobility. A necessary condition for this approach to work is the existence of an end-to-end path between source and destination in a combined connectivity graph formed by overlapping connectivity graphs over a time interval.

This routing paradigm differs significantly from traditional MANET routing models: rather than transmitting message, nodes carry the data around the network by means of their mobility. As a result, knowledge and assumptions regarding node mobility plays an important role in the design and analysis of routing schemes for DTNs.

In this paper, we study the impact of mobility model on the performance analysis of routing methods in DTNs. We show how understanding node mobility characteristics can lead to a better understanding of how to do performance analysis. The mobility model we study is Random Waypoint (RWP). Despite some known problems in its early use [12], RWP is still one of the most widely used mobility models for wireless MANET simulations.

A number of DTN routing schemes assume prior knowledge of node mobility and connectivity, or oracles, to perform message transfers [5][14]. Node contact times and other relevant information such as contact durations and bandwidth during a contact are assumed to be known to nodes so that they can make routing decisions accordingly. Although it is possible for nodes to have prior knowledge of node mobility in some scenarios, in many real life applications such information may not be readily available. In this work, we study routing schemes for the cases where no knowledge about future contacts or mobility patterns is available. Under such circumstances, nodes perform independent routing decisions when they meet each other according to the routing algorithm being used.

In most of the scenarios, all the nodes are assumed to have the same or similar resource constraints. However, in some scenarios it is possible to introduce nodes with greater buffer size and power supply. We call such nodes *cores* or *super nodes*. Resource constrained regular nodes can delegate some or all of their packets to super nodes for delivery to the destination. In this paper we also propose the use of core-assisted schemes and present the results of an extensive set of simulation experiments for Spray and Wait and Core-assisted routing approaches under a variety of settings.

The rest of the paper is structured as follows. Section II goes over related work. Section III describes the proposed approaches for multicasting in DTNs. Section IV gives performance analysis for direct transmission. Section V describes experimental results of the routing approaches. Finally, Section

VI concludes the paper.

II. RELATED WORK

Because of frequent network partitions many of the more traditional routing techniques for MANETs will not work properly [2], [3], [5]. This fact has led to recent interest in developing new approaches for routing in a DTN environment. Here we review some of this work.

The basic routing paradigm for effective routing in DTNs is to use the *store-and-forward* approach, where intermediate nodes keep the messages until new links come up in the path to the destination. One general class of proposed DTN routing algorithms assumes some level of knowledge regarding node mobility and connectivity.

Jain et. al. formulates the DTN routing in terms of a directed multi-graph, where more than one edge may exist between a pair of nodes [5]. Such multiple edges exist because there may be more than one distinct physical connections or different network links may only be available at different time intervals. By using different levels of information regarding connectivity and/or mobility, routing decisions can be made at individual nodes. However, in many cases no such information may be known to the nodes in the network. Under such conditions, different routing approaches are necessary for effective message delivery.

In the *Data Mule* approach proposed in [7] a number of mobiles nodes perform random walks to collect packets, buffer them, and deliver them to wired access points. Zhao '04 introduce a route planning strategy using message ferries that travel on a trajectory to provide communication services [13]. Either the message ferries choose a trajectory to contact nodes, or the nodes can move near to pre-defined trajectory at a certain time to exchange packets. This type of work assumes some level of knowledge regarding node mobility and connectivity.

Other recent DTN routing approaches concentrate on trading off message complexity versus increasing the likelihood of message delivery. To limit the number of messages *single copy* routing schemes allow only one copy of the message at a time to be present in the network. *Direct transmission* is the simplest form of single copy routing, where each source node keeps its messages until it comes into direct contact with the respective destination nodes. Under this scheme only one message transfer is made per delivered message, incurring minimal message passing. However, in intermittently connected networks, such an approach may produce low delivery ratios and has an unbounded delivery delay [4]. An improved scheme is *randomized routing*. In randomized routing, a node A hands over a message another node B with probability $p > 0$. However, the progress of the message towards the destination can be marginal unless contact information is utilized to make routing decisions, as in *utility-based routing* or other hybrid approaches [8].

Generally, single copy schemes are more efficient in terms of reducing traffic overhead. However, message delivery ratios are normally lower while delivery delays are high. One way to improve delivery performance is to use multiple copies of the same message within the network. Each copy can take a different path, thereby increasing the likelihood of delivery as well as decreasing the delay. A variant of this basic approach is to allow each copy to be divided into multiple chunks using techniques such as erasure coding [11]. These approaches allow multiple messages to be reconstructed at the destination.

One policy to implement a multi-copy scheme is to use simple flooding. However, due to frequent network partitions and excessive overhead a better approach is to use Epidemic routing [10]. In Epidemic routing when a pair of nodes comes into contact the nodes exchange any missing packets. Given enough storage space, Epidemic routing can be used to reliably disseminate data across the network. By keeping a history of past encounters, nodes can reduce the overhead of Epidemic routing [6]. However, due to its large overhead, a flooding schemes such as Epidemic routing may not be applicable under circumstances where storage and power supplies are limited. In this work, we propose to use the core-assisted routing as an alternative scheme under resource scenarios where most nodes are resource constrained.

To address overhead problems in flooding, different forms of controlled flooding have been proposed. For instance, Spyropoulos et. al. present Spray and Wait [9]. In this method, a total of L copies of a message are initially spread to other "relay" nodes. If the destination is not found in this phase, each of the nodes carrying a copy of the message will perform direct transmission. In essence, Spray and Wait is a type of controlled flood. No mobility or connectivity information regarding the nodes in the network are assumed to be known for this scheme to work. Mathematical analysis and experimental results for the delay characteristics of Spray and Wait are given in [9]. In this work, we extend the performance evaluation of Spray and Wait through simulation to include delivery ratio, delay, and buffer occupancy.

Since many DTN routing schemes rely on node mobility to transfer messages and since much current performance analysis depends upon the particular mobility model employed, it is highly desirable to better understand the statistical properties a model's node inter-arrival meeting times. However, there has been little work regarding this subject. With regard to this, we address the effect of mobility models on DTN routing schemes in this paper.

III. ROUTING IN DTNS

In principle DTN routing can involve either a single-copy or multi-copy approach. Single-copy approaches try to reduce buffer usage and the number of message transfers, but suffers from large delays and low delivery ratios. Multi-copy schemes, on the other hand, achieves lower delays and higher delivery ratio at the cost of buffer space and more message transfers. This section first presents some background information on direct transmission as a simple example of single-copy routing. We then describe Spray and Wait and Epidemic routing, both of which are examples of multi-copy routing. Finally we propose the use of special *core based* nodes as a method to improve performance.

A. Direct transmission

Direct transmission is the simplest single-copy routing scheme: after generating a message, the source waits until it comes into contact with the destination. The message delivery delay for this scheme is unbounded [4]. The main advantage of this scheme is that it incurs minimum data transfers for message deliveries. It also provides a base case for evaluating other routing methods.

B. Epidemic Routing

In terms of message overhead the opposite approach from direct transmission is flooding. In a flooding approach every node that receives a packet broadcasts it to all of its neighbors. However, in intermittently connected networks simply broadcasting packets may not achieve the goal of reaching as many nodes as possible due to network partitions. In such a context, an *Epidemic routing* approach outperforms flooding [10].

When two nodes come into contact, each node will exchange message information to see if there are any messages that the other node has that it has not received. Message indexes are sent as a summary vector. After such pair-wise exchange of messages, each node will get all the messages carried by the other node that it has not received by far. This means that as long as buffer space is available, messages spread like an epidemic disease among nodes through “infection”.

To determine which packets have been previously seen, there must a globally unique message ID. We can use of a tuple $(source\_id, sequence\_number)$ where $source\_id$ is the id of the sending node and $sequence\_number$ is a unique sequence number for each message sent by the node. A time limit is used as the minimum time span between two exchanges for any given pair of nodes to reduce the number of vector exchanges. For a large number of messages, the summary vector can become quiet large. To reduce the size of the summary vector, it may be possible to use compression techniques such as a *Bloom Filter* [1].

C. Spray and Wait

Although Epidemic routing can achieve high delivery ratios and low delays, it requires that nodes to have sufficiently large buffer spaces and can incur large message transfer overhead. An alternative approach is to control the level of flooding using techniques such as *Spray and Wait* [9]. Spray and Wait works as follows: L number of copies are initially spread over the network by the source or other nodes to L distinct relays (spray phase). If the destination is not found during this phase, each node that is carrying a copy of the message performs a direct transmission.

There are different ways to spray messages. For example, the source node can either give a copy to each new node encountered, or give multiple copies so that the relay nodes can also perform spraying. [9] shows that the *binary* spray approach will give the optimal performance. Basically, in binary spray source node give half of the copies of a message to the new relay, and keeps the rest to itself. The source and relays continue in this way until there is only one copy left.

D. Core-aided Routing

Some DTN systems, such as sensor networks, may have resource constrained nodes. We propose to address this issue by assuming that some nodes are not resource constrained, and to designate these nodes as as *core nodes*. When regular nodes come into contact with core nodes, they can delegate all or some of their packets to the core node.

When nodes meet these core nodes, there are several possible policies. We say that regular nodes can either use a “copy-to-core” or a “dump-to-core” approach. In the copy scheme regular nodes give copies of their messages to the core and keep them. This will increase the number of message carriers, thereby reducing the message delay and increasing message delivery ratio. If there are more than one core nodes in the system, they can further exchange their packets similar to Epidemic routing, as they are not constrained in terms of resources.

In the dump scheme they delete the messages delegated to the core. This can be helpful in environments where buffer storage for most nodes is very stringent. One potential drawback to this approach is a possible increase in message delivery delay. This is because only the core node carries messages, thereby decreasing the chance of contacting destination nodes in a short time.

IV. MODEL ANALYSIS

Under DTN routing schemes, rather than transmitting message, nodes carry the data around the network by means of their mobility. Therefore, knowledge and assumptions regarding node mobility plays an important role in the analysis of routing schemes for DTNs, especially when no prior knowledge about node connectivity is assumed. Under such circumstances, nodes perform independent routing decisions when they meet other nodes according to the current routing algorithm.

Because of such dependence of routing algorithms on the mobility of nodes, we first study the statistical properties the node mobility model through simulations. Then we demonstrate how such information regarding the node mobility can be used to do performance analysis under DTN routing schemes.

Since message transfers only occur when nodes meet each other, studying the characteristics of meeting times, or the inter-arrival times of nodes, plays an important role in the analysis of routing schemes. As a result, we try to study node meeting times and their distribution under our experimental settings. The mobility model used in our simulations are the Random Waypoint mobility model, a common mobility model used in simulation for wireless mobile networks.

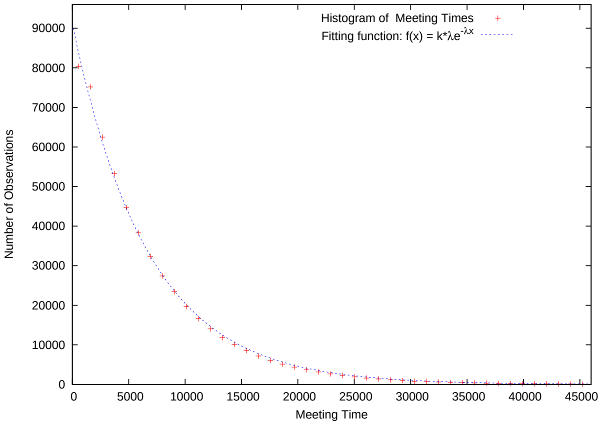

Fig. 1. Histogram of Inter-arrival Times with Curve-fitting

To understand the characteristics of the node arrival times, we record the inter-arrival times of nodes during the simulation. Figure 1 shows the distribution of inter-arrival times of nodes by a histogram. Due to its similarity with the exponential distribution, we perform a curve-fitting with function $f(t) = k * \gamma e^{-\gamma t}$ via coefficient k only. The rate parameter γ is taken as inverse of the average of inter-arrival times observed during simulation. We can see from the graph that the observed inter-arrival times fits very closely to the exponential function $f(t)$. To verify the goodness of fit, we do a Quantile-Quantile (Q-Q) plot of observed inter-arrival times and generated exponential variates with the same rate. Figure 2 shows the linearity of Q-Q plot with a high R^2 (0.99) value. Similar results are obtained if we change the radio transmission range or the movement speed of nodes.

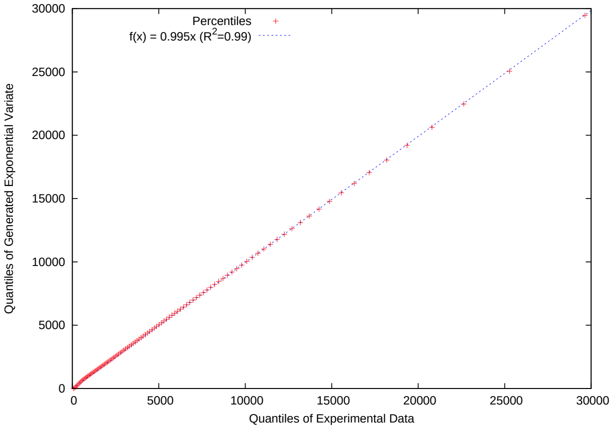

Fig. 2. Quantile-Quantile Plot

From the observations above we can see that the node inter-arrival times are exponentially distributed. We use this property to analyze performance metrics under the Direct Transmission routing scheme. We use *message delivery ratio*, *delay of delivered messages*, and *buffer occupancy* as the main performance metrics. Below we provide analysis for each of these metrics. We assume that inter-arrival times (meeting times) of mobile nodes are exponentially distributed with a rate of γ as shown above, and messages are generated with an average rate of λ from a uniform distribution.

A. Message Delivery Ratio

Provided that node arrival times are exponentially distributed with a rate of γ, for a message entering in the queue at time 0 the probability that the message is delivered before it is expired can be given in the form of CDF as follows:

$$E[R] = 1 - e^{-\gamma T_x} \tag{1}$$

where T_x is the message expiration time. Here we assume that no messages are dropped due to buffer overflow.

B. Delay of Delivered Messages

Given message expiration time T_x, messages get delivered if the destination is reached within T_x, or it will be dropped. From an application point of view, we are only interested in the expected time that the delivered messages spend in the buffer queue before it gets delivered, i.e. the delay of delivered messages.

Since we assume that the inter-arrival times of nodes are exponentially distributed with a rate of γ, the probability of a message being delivered to the destination at time t after it enters the queue can be given by

$$f(t) = \gamma e^{-\gamma t}$$

As we are only interested in delivered messages, the probability function given above becomes a conditional probability for the messages that are delivered:

$$f_d(t) = \frac{f(t)}{P(t < T_x)} = \frac{\gamma e^{-\gamma t}}{1 - e^{-\gamma T_x}} \tag{2}$$

where $P(t < T_x)$ denotes the probability that the destination is reached before T_x, which is given by the CDF of $f(t)$.

Therefore, the expected waiting time of a delivered message can be written as

$$\begin{aligned}
E(T) &= \int_0^{T_x} t f_d(t) dt \\
&= \frac{\gamma}{1 - e^{-\gamma T_x}} \int_0^{T_x} t e^{-\gamma t} dt \\
&= \frac{\gamma}{1 - e^{-\gamma T_x}} \left[\frac{-t e^{-\gamma t}}{\gamma} \bigg|_0^{T_x} + \frac{1}{\gamma} \int_0^{T_x} e^{-\gamma t} dt \right] \\
&= \frac{\gamma}{1 - e^{-\gamma T_x}} \left[\frac{-T_x e^{-\gamma T_x}}{\gamma} + \frac{1}{\gamma} \frac{-e^{-\gamma t}}{\gamma} \bigg|_0^{T_x} \right] \\
&= \frac{\gamma}{1 - e^{-\gamma T_x}} \left[\frac{-T_x e^{-\gamma T_x}}{\gamma} + \frac{1 - e^{-\gamma T_x}}{\gamma^2} \right] \\
&= \frac{1}{\gamma} - \frac{e^{-\gamma T_x}}{1 - e^{-\gamma T_x}} T_x
\end{aligned} \tag{3}$$

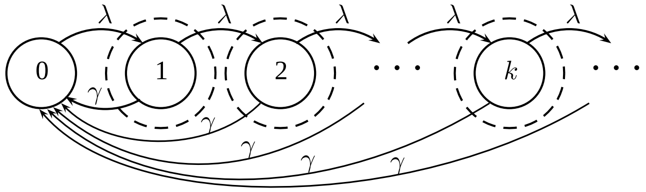

Fig. 3. State Transition Diagram for Number of Message in the Buffer

From (3) we can see that the expected delay of delivered messages will be less then $1/\gamma$, and will approach $1/\gamma$ when $T_x \to \infty$. In other words, the expected delay is bounded by the expected arrival time of the destination node.

C. Buffer Occupancy

Given the message generation rate, λ and inter-arrival (meeting) rate of nodes, γ, we can find the number of nodes in the buffer using a queueing system model with vacations, where the message arrival rate is λ and vacation time is distributed exponentially with a rate of γ. Since the message transfer time is very small compared to expected arrival times, we can take the service rate as infinite for simplification. We assume that no messages are dropped due to buffer overflow.

Figure 3 shows the state transition diagram for the number of messages. Because in steady state the flow-in equals flow-out for any state, it can be shown from the diagram that for any $k = 1, 2, \cdots$ that the following relationship holds:

$$p_k\lambda + p_k\gamma = p_{k-1}\lambda$$

where p_k is the probability that there are k messages in the buffer queue. Equivalently,

$$p_k = \frac{\lambda}{\lambda+\gamma}p_{k-1} = p_0\left(\frac{\lambda}{\lambda+\gamma}\right)^k, \quad k = 1, 2, \cdots \qquad (4)$$

Since the sum of all probabilities is 1, we have

$$\sum_{k=0}^{\infty} p_k = \sum_{k=0}^{\infty} p_0\left(\frac{\lambda}{\lambda+\gamma}\right)^k = 1$$

Noting that left hand side is a geometric series, we can give p_0 as follows:

$$p_0 = \left[\sum_{k=0}^{\infty}\left(\frac{\lambda}{\lambda+\gamma}\right)^k\right]^{-1} = \left[\frac{1}{1-\frac{\lambda}{\lambda+\gamma}}\right]^{-1} = \frac{\gamma}{\lambda+\gamma} \qquad (5)$$

With p_0 known, (4) can now be rewritten as

$$p_k = p_0\left(\frac{\lambda}{\lambda+\gamma}\right)^k = p_0(1-p_0)^k, \quad k = 1, 2, \cdots \qquad (6)$$

The expected number of messages in the queue, $\bar{n}$, can be given as

$$\begin{aligned}
\bar{n} &= \sum_{k=0}^{\infty} k p_k \\
&= \sum_{k=0}^{\infty} k p_0 (1-p_0)^k \\
&= p_0 \sum_{k=0}^{\infty}(k+1)(1-p_0)^k - p_0\sum_{k=0}^{\infty}(1-p_0)^k \\
&= p_0 \frac{1}{(1-(1-p_0))^2} - 1 \\
&= \frac{1}{p_0} - 1 \\
&= \frac{\lambda}{\gamma} \qquad (7)
\end{aligned}$$

The third line is reached by expanding k as $(k+1)-1$, and first term denotes the expression for the expected value of geometric series with a scale factor of p_0, while the second term represents the sum of all probabilities $p_k, k = 0, 1, \cdots$, which is equal to one.

Intuitively, since the expected arrival time of the destination node is given by $1/\gamma$ and the message generation rate is λ, the average number of messages generated between two arrivals is $\lambda * 1/\gamma = \lambda/\gamma$.

Given that the inter-arrival times are i.i.d in a system of M mobile nodes, and the message expiration time of T_x, we can give the expected number of messages in the buffer for each node, $\bar{N}$ as

$$\bar{N} = \frac{\lambda}{\gamma}(M-1)\left(1 - e^{-\gamma T_x}\right) \qquad (8)$$

V. EXPERIMENTAL RESULTS

In this section, we present our experimental study for the performance analysis in the previous section, as well as experimental results for the Spray and Wait and Core-assisted routing approaches under different settings. Our results were obtained from simulation experiments using ns-2 and our own code.

A. Metrics and Methodology

All of our experiments use the standard ns-2 mobile wireless models, including the default transmission model which has a $250m$ radio range. We collected statistics for the average message delivery rate, the average message delivery delay, and the buffer occupancy.

The *average message delivery ratio* is the ratio of delivered messages to the number of messages that should have been delivered to destination nodes. The average message delivery ratio reflects the overall efficiency of the method in delivering messages. The *average delivery delay* is the average delay of all the messages delivered to destinations. The delay of a delivered message is calculated by subtracting the delivery time by the message generation time. We also use *buffer occupancy* as a metric to evaluate the buffer requirements of a specific routing method.

The default settings in our simulations are as follows. Each simulation run has 40 nodes in a $6000m \times 6000m$ area. Each node generates a message to another node at random at every 50 seconds on average (the time between two message generations is chosen randomly and uniformly from 0 to 100 seconds).

The simulations use the random-waypoint (RWP) mobility model. In RWP nodes randomly choose a point in the area and moves towards that destination with an average speed uniformly distributed between V_{min} and V_{max}. In our simulations, V_{min} is 3, V_{max} is 10. Pause time after reaching the destination point is 3 seconds.

Each node has a storage space that can hold 1000 messages. Message expiration times are changed between 3000∼8000 seconds for performance evaluations. Messages will not be delivered if expired. The simulation time is 175,000 seconds. We run each experiment with a random seed for at least 25 times. All the data points presented are plotted with 95% confidence interval.

The $HELLO$ messages, which act as *heart beat* or *beacon* messages, are generated every 3 seconds. Upon receiving a $HELLO$ message, a node looks to see if there are any messages waiting to be delivered to the sender, and, if so, sends the messages. Also, if time the passed since last $HELLO$ message has exceeded some threshold T_{th}, it is counted as a new arrival and the time passed since the end last encounter is recorded as an inter-arrival time. Ideally, T_{th} would be set only slightly larger than the $HELLO$ interval (3 seconds in this case). In our simulation environment, ns-2, however, we have found that setting the T_{th} smaller than 3.5 seconds would steeply increase the inter-arrival rate. We believe that this is caused by message delivery jitter, causing a neighboring node to appear a newly arrived node. Although some amount of jitter is known to exist, a high jitter value of 0.5 second was unexpected.

B. Message Delivery Ratio

To measure the message delivery ratio, nodes record the message generations and arrivals. Under normal simulation settings, messages generated towards the end of the simulation have a lower chance of getting delivered. Specifically, given simulation duration T_{sim} and message expiration time T_x, a message generated after time $T_z = T_{sim} - T_x$ has a lower likelihood of delivery than one generated before T_z. This will negatively affect the observed delivery ratio. To account for this, nodes stop generating messages after time T_z, but continue receiving till the end of simulation.

Theoretical results from Equation (1) and experimental values are shown in Figure 4.

C. Delay of Delivered Messages

Similar to the arguments given above, observed delay of delivered messages is also affected by messaged that are generated after T_z. Although messages generated after T_z have lower likelihood of getting delivered, the ones that do get delivered have a lower delay. We stop message generation after T_z to avoid such errors.

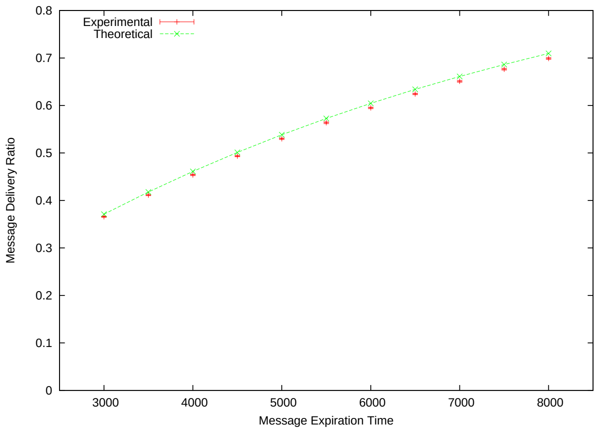

Fig. 4. Message Delivery Ratio

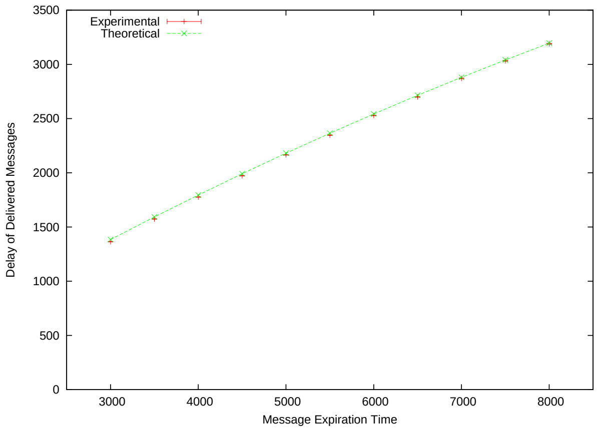

Fig. 5. Delay of Delivered Messages

Simulation results and the corresponding theoretical results obtained from (3) are shown in Figure 5.

D. Buffer Occupancy

As opposed to the delivery ratio and delay, nodes should keep generating messages as usual after T_z to get an accurate observation of the steady state behavior, since stopping message generation will lower the observed buffer occupancy than the actual value.

From the arguments above for delivery ratio, delay, and buffer occupancy we can see that different performance metrics with different semantics should be treated accordingly to obtain accurate simulation results.

Figure 6 shows the experimental values and theoretical results from (8) for buffer occupancy when the message expiration time changes.

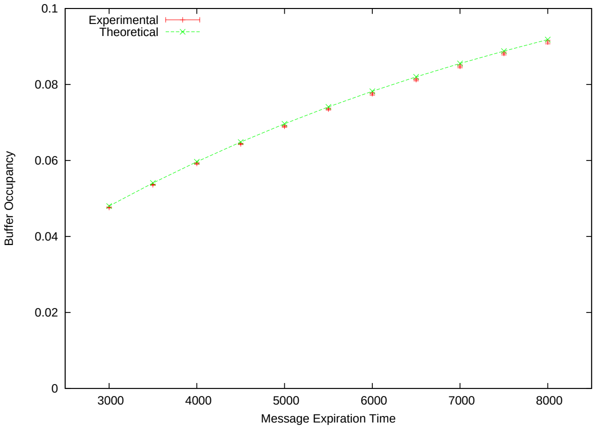

Fig. 6. Buffer Occupancy

E. *Effects of Core Nodes*

Figure 7 shows that case where the ratio of core nodes is increased from 0% (no cores) to 20%. When regular nodes meet core nodes, they send their packets to cores. They can either use "copy" scheme, where they keep the copies of their messages, or use "dump" scheme to delete them. From Figure 7 we can see that even a small increase increase of core ratio in the "copy" scheme results in a noticeable increase in the delivery ratio. In the "dump" scheme, on the other hand, that there is almost no change in the delivery ratio.

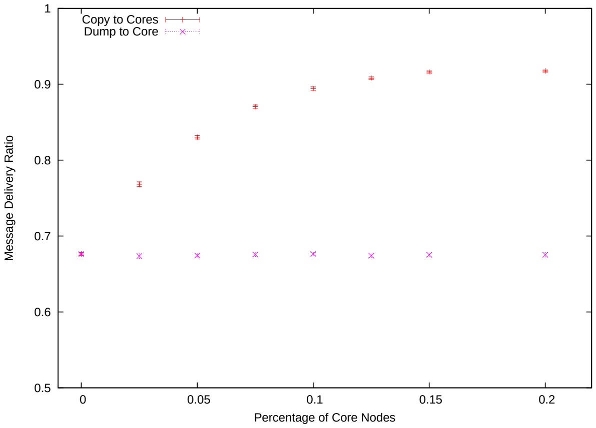

Fig. 7. Effect of Cores on MDR

Basically, the "copy" case doubles the number of carriers of message. The probability of this happening depends on the number of core nodes and the arrival rate. The "dump" case only transfers the message to another carrier, the core, to deliver it to the destination. Practical scenarios would lie between these two extremes if a regular node has enough buffer size hold newly generated messages till the core nodes arrives, but not enough to hold all the messages until they are delivered to the destination by the node itself.

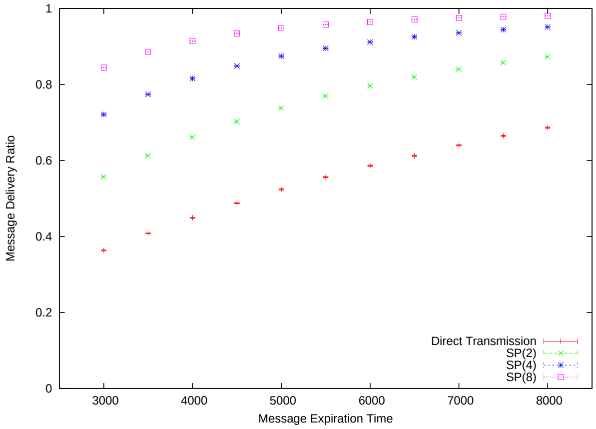

Fig. 8. Message Delivery Ratio: Direct Transmission and Spray-and-Wait

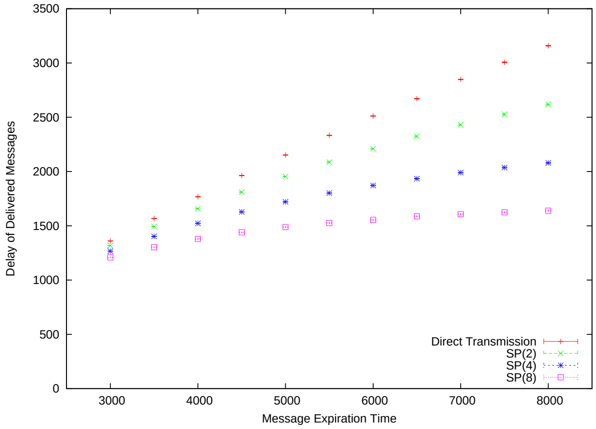

Fig. 9. Delay: Direct Transmission and Spray-and-Wait

F. *Performance of Spray-and-Wait*

Beside direct transmission and cores, we also examine the performance of Spray-and-Wait routing scheme. We use 2-, 4-, and 8-copy schemes for spraying a message and examine its effects on performance metrics when message expiration times varies. For optimal performance, we use the Binary Spray-and-Wait as described in [9]. To avoid sending multiple copies of a message, a field is used to denote the number copies under the current scheme, L, when each message is generated. When a new node arrives when $L > 1$, the message is sprayed delegated to the new node with L set to the new value of $\lfloor L/2 \rfloor$, and the sender decreases its own L to $\lceil L/2 \rceil$. Direct transmission can be regarded as a special case where L is set to 1 at message generation.

Figure 8 shows the message delivery ratios under different SP schemes when the message expiration time, T_x, is varied. Direct transmission is also included for base comparison. As expected, the MDR increases as we increase the number of message copies, L, to spread. However, the effect of increasing L quickly diminishes, especially when T_x is set higher.

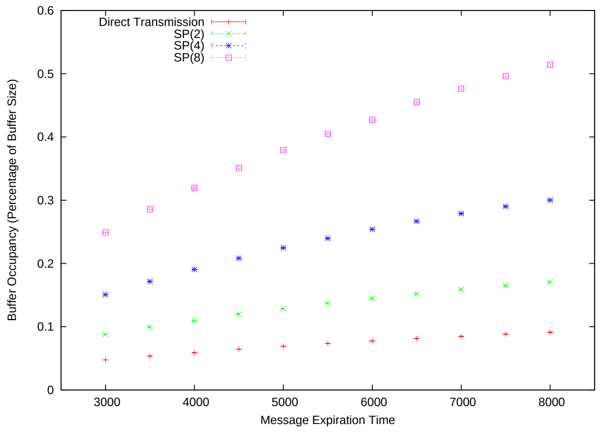

Fig. 10. Buffer Occupancy: Direct Transmission and Spray-and-Wait

Figure 9 depicts the message delays for direct transmission and SP. We can observe that the delay increases as T_x is increased, and the difference in performance also becomes more apparent.

Buffer occupancies for different routing schemes are shown in Figure 10. As expected, the direct transmission routing scheme has the lowest buffer occupancy.

As we can see, increasing L generally increases the delivery ratio and lowers the delay, at the cost of higher buffer occupancy. Parameters can be set in actual scenarios according to the power and buffer constraints to achieve the desired performance.

VI. CONCLUSIONS

In this paper we discussed the impact of the mobility model on routing in delay tolerant networks. We used the Random Waypoint mobility model for underlying node mobility in our stimulation's. We observed the exponentiality of node inter-arrival times. Through extensive simulations studies, we have shown how such information node mobility can be used to obtain accurate analysis of important performance metrics under DTN routing schemes. Such results are helpful for the analysis and design of routing algorithms for DTNs. Besides, we also proposed the use of Core-assisted schemes under circumstances where most nodes are resource constrained. We described different routing methods and presented corresponding simulation results for the Core-assisted routing scheme. Finally, we also presented a results of an extensive set of simulation experiments for Spray and Wait routing scheme under a variety of settings.

REFERENCES

[1] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[2] Scott Burleigh and Kevin Fall. Delay tolerant networking: An approach for interplanetary internet. *IEEE Communications Magazine*, June 2003.

[3] Kevin Fall. A delay-tolerant network architecture for challenged internets. pages 27–34, Karlsruhe, Germany, August 2003. ACM SIGCOMM'03.

[4] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *INFOCOM*, pages 1360–1369, 2001.

[5] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. Portlan, Oregon, USA., August 2004. ACM SIGCOMM'04.

[6] A. Lindgren, A. Doria, and O. Scheln. Probabilistic routing in intermittently connected networks. Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), 2003.

[7] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. IEEE SNPA Workshop, May 2003.

[8] T. Spyropoulos, K. Psounis, and C. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Sensor and Ad Hoc Communications and Networks*, pages 235–244. IEEE, Oct 2004.

[9] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. Philadelphia, PA, USA, August 2005. ACM SIGCOMM'05 Workshops.

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

[11] Yong Wang, Sushant Jain, Margaret Martonosi, and Kevin Fall. Erasure-coding based routing for opportunistic networks. Philadelphia, PA, USA, August 2005. ACM SIGCOMM'05 Workshops.

[12] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. INFOCOM, 2003.

[13] Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. ACM MobiHoc, ACM, May 2004.

[14] Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. Multicasting in delay tolerant networks: Semantic models and routing algorithms. In *SIGCOMM'05 Workshops*, pages 268–275. ACM, August 2005.