A Novel Message Scheduling Framework for Delay Tolerant Networks Routing

A Novel Message Scheduling Framework for Delay Tolerant Networks Routing

Ahmed Elwhishi, Pin-Han Ho, K. Naik, and Basem Shihada

Abstract—Multi-copy routing strategies have been considered the most applicable approaches to achieve message delivery in Delay Tolerant Networks (DTNs). Epidemic routing and two-hop forwarding routing are two well reported approaches for delay tolerant networks routing which allow multiple message replicas to be launched in order to increase message delivery ratio and/or reduce message delivery delay. This advantage, nonetheless, is at the expense of additional buffer space and bandwidth overhead. Thus, to achieve efficient utilization of network resources, it is important to come up with an effective message scheduling strategy to determine which messages should be forwarded and which should be dropped in case of buffer is full. This paper investigates a new message scheduling framework for epidemic and two-hop forwarding routing in DTNs, such that the forwarding/dropping decision can be made at a node during each contact for either optimal message delivery ratio or message delivery delay. Extensive simulation results show that the proposed message scheduling framework can achieve better performance than its counterparts.

Index Terms—Routing, Buffer management, Message scheduling, DTN.

1 Introduction

One of the most important characteristics of a delay tolerant network (DTN) is the lack of an end-to-end path for a given node pair for an extended period [1]. To cope with frequent and long-lived disconnections due to node mobility, a node in a DTN is allowed to buffer a message and wait until it finds an available link to the next hop. The next hop node buffers and forwards the received message accordingly if it is not the destination of the message. This process continues until the message reaches its destination. This model of routing constitutes a significant difference from conventional ad hoc routing, and is usually referred to as encounter-based routing, store-carry-forward routing, or mobility-assisted routing [6], [3], [7]. The names come from the fact that the routing of a message in DTNs has taken the nodal mobility as a critical factor in the decision on whether the message should be forwarded.

Most DTN routing protocols have assumed negligible storage overhead [2], [4] without considering that each node could be with a limited buffer space. Note that buffering and forwarding unlimited number of messages may also cause intolerable resources and nodal energy consumption; and it is imperative to set up buffer limitations at the DTN nodes to better account for the fact that each node could be a hand-held and battery-powered device with stringent limitations on buffer space and power consumption. With such buffer limitations at the DTN nodes, message drop/loss could happen due to buffer overflow. This leads to a big challenge in the implementation of most previously reported DTN routing schemes.

There are two widely employed DTN routing schemes, namely; the epidemic (or flooding) and controlled flooding (or two-hop forwarding) schemes [2], [5]. With the Epidemic scheme, whenever two nodes encounter each other, they exchange all messages they do not have in common. Therefore, the message copies are spread like an "epidemic" throughout the network to every node using the maximum amount of resources. With controlled flooding, a limited number of copies of each message are generated and disseminated throughout the network. The source node forwards a message copy to the first $L-1$ nodes it encounters, and then each encountered node keeps a copy of the message until it meets the destination node of the message. This strategy of message forwarding is known as two-hop forwarding or source forwarding (SF). An important issue in such a category of DTN routing is when and to which node the stored messages should be forwarded. Obviously, both the above schemes require additional efforts in order to incorporate with the given buffer space limitation at each node.

This paper studies a novel message scheduling framework for DTNs under epidemic and two-hop forwarding, aiming to enable an effective decision process on which messages should be forwarded and which should be dropped when the buffer is full. Such a decision is made by evaluating the impact of dropping each buffered message according to collected network information for either optimal message delivery ratio or message delivery delay.

To deal with the message propagation prediction under epidemic forwarding and evaluate the delivery

- *Ahmed Elwhishi, Pin Han Ho, and K. Naik are with the Department of Electrical and Computer Engineering, University of Waterloo, Ontario, Canada.*
  E-mail: aelwhish@engmail.uwaterloo.ca, pinhan@bbcr.uwaterloo.ca, snaik@uwaterloo.ca.
- *Basem Shihada. is with MCSE Division, KAUST, Thuwal, Saudi Arabia.*
  E-mail: basem.shihada@kaust.edu.sa

delay and/or delivery ratio at any time instance during message lifetime, Markov chain model has been proven to be the best method in doing such evaluation [5]. Nonetheless, providing numerical solution for such model becomes impractical when the number of nodes is large [9], [8].

To cope with the high computation complexity in directly solving a Markov chain model, we develop a fluid flow limit model and the corresponding ordinary differential equations (ODEs) formulation as our solution. The use of ODEs, although serving as an approximation of the Markov chain result, can nonetheless improve the computation efficiency and provide a closed-form expression. Further, the formulation with the proposed fluid flow limit model is highly scalable to the network size, where the complexity does not increase with the number of network nodes. For example, the problem in [40] that takes up to 178 seconds by solving a continuous-time Markov chain can be solved by an equivalent ODE model with only 2.8 seconds; and it shows a dramatically increase of computation complexity by using Markov chain when the problem state space is getting larger, while the number of corresponding ODEs is constant regardless of the number of components in the system.

The ODE solution gives per-message utility values, which are calculated based on the estimation of two global parameters: the number of message copies, and the number of nodes which have "seen" this message (the nodes that have either carried the message or rejected the acceptance of this message). The per-message utility values are calculated at each node and then used for the decision on whether the buffered messages should be dropped in any contact. We will demonstrate a closed-form solution to the proposed ODE approach, such that each per-message utility can be calculated efficiently. Simulation is conducted and the results confirm the efficiency and effectiveness of the proposed buffer management scheme under the epidemic routing and two-hop forwarding.

The contributions of the paper are as following:

- Developing new utility-based message scheduling mechanism that incorporates with DTN message forwarding, where per-message utility is determined to optimize either message delivery ratio or delivery delay.
- Developing a novel estimation approach for network global knowledge to facilitate decisions on which message should be forwarded/dropped when the buffer of the encountered node is full.
- Evaluating and comparing the proposed scheme with counterparts, and gaining understanding on its tradeoff between computation complexity and performance improvement.

The rest of this paper is organized as follows. Section 2 describes the related work in terms of flooding-based DTN routing, and buffer management and scheduling in DTNs. Section 3 provides the background and system description, including a brief overview on the fluid flow model and network model adopted in this study. Section 4 introduces the proposed message scheduling framework under epidemic and two-hop forwarding routing, including the key functional modules that constitute the whole system, namely Summary Vector Exchange Module (SVEM), Network State Estimation Module (NSEM), Utility Calculation Module (UCM), and Forwarding and Dropping Policy (Decision) Module. Section 5 provides utility function derivation that serves as the core of the proposed system. Section 6 provides experiment results which verify the proposed DTN message scheduling framework and discusses the tradeoff between the computation and performance. Section 7 concludes the paper.

2 Related work

This section gives a survey on the existing literature related to our work in terms of DTN routing schemes and buffer management policies.

2.1 DTN Routing

In context of DTNs, when two nodes encounter each other, they exchange summary vector [2] which contains an index of all messages carried by a node. Based on some specific information, a routing strategy is then applied in order to decide which message to forward. The fastest way to deliver messages is to spread the messages to all hosts, thus forming a type of persistent flooding, which is known as epidemic routing [2]. In this scheme, all the messages are eventually spread to all nodes in the entire network. Although considered to be very robust against node failure and to provide the fastest message delivery, the scheme is very resource-consuming in terms of the number of transmissions and number of message copies stored in each node; and such resource consumption increases exponentially as the number of nodes and the traffic load increases. Clearly, epidemic routing is impractical in most real application scenarios in which bandwidth, buffer space, and energy are scarce resources due to the possible large queuing delay, and a significant number of retransmissions and message drops at each node [18], [17], [3], [12].

Some studies tried to improve the performance of epidemic routing by reducing the resources consumption [19], [20], [21], [23], [24], [25], [26], [29], [17], [3], [22]. These schemes are known as multi-copy (controlled flooding) schemes. Spray routing is a family of multi-copy schemes [12], [7], [6], [28], [27] that was developed to achieve fast message delivery and less transmissions by limiting the number of message copies possibly launched in the network. The schemes under Spray routing generate only a small number of copies to ensure not overloading the network with launched messages. Nonetheless, the performance of these schemes degrades when the traffic demand is higher than the available network resources.

Other schemes are based on social networks analysis, called social network based forwarding [36], [37], [38], [41]. With these schemes, the variation in node popularity, and the detectability of communities, are employed as main factors in forwarding decisions.

Although each previously reported study solved the DTN message scheduling problem in some aspects, most (if not all of them) have generally focused on the performance impact due to nodal mobility and population, while overlooking the possible effect by message drop due to limited nodal resources and potential contention when the nodal density/traffic is high. There is clearly a missing piece for a message scheduling mechanism which can incorporate with message forwarding tasks according to the estimated global network states.

2.2 Buffer Management

Only a few studies have examined the impact of buffer management and scheduling policies on the performance of DTN routing. Zhang et al. in [9] addressed this issue in the case of epidemic routing by evaluating simple drop policies such as drop-front and drop-tail, and analyzed the situation where the buffer at a node has a capacity limit. The paper concluded that the drop-front policy outperforms the drop-tail. Lindgren et al. in [10] evaluated a set of heuristic buffer management policies based on locally available nodal parameters and applied them to a number of DTN routing protocols. Fathima et al. in [34] proposed buffer management scheme which divides the main buffer to a number of queues of different priorities. When the entire buffer is full, some of the messages in the lowest priority queue are dropped to give room for new messages.

Similar idea was explored by Dimitriou et al. [39], who proposed a buffer management policy based on two types of queues for respective type of data traffic; namely a low-delay traffic (LDT) queue and a high-delay traffic (HDT) queue.

Noticeably all the above mentioned policies are based only on static and local knowledge of network information. In [32], Dohyung et al. presented a policy which discards first a message with the largest expected number of copies. Erramilli et al. in [35] proposed policies in a conjunction with forwarding algorithms. Two issues are raised in [35]. First, without addressing the message scheduling issue which is of the same importance as buffer management, the scheme in [35] may not be able to fully explore the possible performance gain in the buffer management scheme. Second, the absence of an analytical model leaves the scheme simply a heuristic hard to be evaluated.

Krifa et al. in [11] proposed an interesting approach for solving the problem of buffer management by way of a drop policy and a scheduling scheme. This is the first study that explicitly takes global knowledge of node mobility as a constraint in the task of message scheduling. Specifically, their method estimates the number of copies of message i based on the number of buffered messages that were created before message i. Although interesting, the method may become inaccurate when the number of network nodes is getting larger, especially for newly generated messages. Meanwhile, the effect due to the change of the number of message copies during the remaining lifetime of a message is not considered in the utility function calculation. This means the utility function is only affected by the current message copies and its remaining lifetime. Moreover, it is assumed that every node is aware of all messages that has encountered during contacts with other nodes, which raises practicability issue. Maintaining such message forwarding history is expected to cause very high overhead.

It is clear that all the above mentioned studies leave a large room to improve, where a solution for DTN message scheduling that can well estimate and manipulate the perceived nodal status is absent.

3 Background and System Description

This section presents the background of our mathematical model as well as the network model for encounter-based epidemic routing.

3.1 Background of Fluid Flow Model

In a nutshell, the paper formulates the DTN message scheduling and dropping task under epidemic and two-hop forwarding routing as a fluid-flow Markov-chain process. The fluid flow model can then be used to formulate the rate of message propagation among nodes, analyze the expected time until a given node (destination) is infected, and then calculate the delivery ratio (delivery probability). Since solving the fluid flow model using a Markov chain based approach is subject to extremely high computation complexity, we approximate the problem by using an ODE and derive a close-form solution of the problem. Note that the ODE based approach for solving a Markov chain model has been used for similar networking problems yet under different scenarios from the one of interest in this study with proven efficiency and correctness [13], [9]. The following notations are used throughout the paper.

- $n_i(t)$ denotes the number of nodes with message i in their buffers (also referred to as "infected" at time t), where t is counted from the creation time of message i. The following relation is used to calculate $n_i(t)$:

$$\frac{dn_i(t)}{dt} = \beta n_i(t)(N - n_i(t))) \tag{1}$$

where N is the number of nodes in the network, and β is the meeting time rate between nodes. Solving (1) with the initial condition $n_i(0)$ yields

$$n_i(t) = \frac{Nn_i(0)}{n_i(0) + (N - n_i(0))e^{-\beta N t}} \tag{2}$$

- $P_i(t) = P_i(T_d < t)$ denotes the cumulative density function (CDF) of message i being delivered at time

t, where T_d denotes a random variable for the time instant that the message i is successfully delivered. $P_i(t)$ can be expressed in a differential equation form [9]:

$$\frac{dP_i(t)}{dt} = \beta n_i(t)(1 - P_i(t)) \tag{3}$$

solving (3) with the initial condition $P_i(0) = 0$ yields

$$P_i(T_d < t) = 1 - \frac{N}{N - n_i(0) + n_i(0).e^{\beta N t}} \tag{4}$$

(1) and (4) are valid only for unlimited buffer space. To extend the above relations to the scenario with a limited buffer space, an additional factor should be considered (denoted as P_{fi}), which represents the probability that the encountered node's buffer space is available and the message can be transferred. Note that P_{fi} can be obtained by historical data of nodal encounters. Accordingly, (1) is formulated as

$$\frac{dn_i(t)}{dt} = P_{f_i}\beta n_i(t)(N - n_i(t))) \tag{5}$$

Thus (2) and (4) are reformulated as follows:

$$n_i(t) = \frac{N}{n_i(0) + (N - n_i(0))e^{-P_{f_i}\beta N t}} \tag{6}$$

$$P_i(T_d < t) = 1 - \left(\frac{N}{N - n_i(0) + n_i(0)e^{\beta P_{fi} N t}}\right)^{\frac{n_i(0)}{P_{fi}}} \tag{7}$$

3.2 Network Model

In this paper, a homogeneous DTN is modeled as a set of N nodes, all moving according to a specific mobility model in a finite area, where inter-encounter time between each pair of nodes follows an independent and identical distributed (iid). Let the number of total messages in the network be denoted as $K(t)$, and the buffer capacity of each node be denoted as B messages. The messages are generated arbitrarily between source and destination nodes. Each message is destined to one of the nodes in the network with a time-to-live (denoted as Tx). A message is dropped if its Tx timer expires.

For any given node, a, it is assumed that $J_a(t)$ messages are stored in its buffer at time t. Each message i, $i \in [1,\, J_a(t)]$ is denoted by a tuple of variables denoted in Table 1. Obviously we have $s_i(t) = n_i(t)$ if all the encountered nodes of message i have available buffer space, and $n_i(t) \leq m_i(t) + 1$. Let the inter-encounter time of any two nodes a and b be denoted as $\triangle T_{(a,b)}$, which is defined as the time period taken by the two nodes to enter into their transmission again. The average encounter (or mixing) rate between a and b, denoted as $\beta_{(a,b)}$, is the inverse of the average inter-encounter time for the two nodes: $\beta_{(a,b)} = \frac{1}{E[\triangle T_{(a,b)}]}$. We assume that all inter-encounter times, $\triangle T_{(a,b)}$, $a,\ b \in [1, N]$ are exponentially distributed (or referred to as with an exponential tail [14]). It has been shown that a number of popular mobility models (e.g., Random Walk, Random Waypoint, Random Direction, Community-based Mobility [5], [42], [15]) have such exponential tails under certain conditions, such as the transmission range should be low compared to the simulation area, and small community size (in case of Community-based Mobility). Recent studies based on traces collected from real-life mobility examples [16] argued that the inter-encounter and the encounter durations in these traces demonstrate exponential tails after a specific cut-off point. Based on the iid of the nodal mobility model, the distribution of the inter-meeting time can be obtained, where the historical inter-encounter information between any two nodes a and b can be calculated by averaging all inter-encounter times until current time t. This distribution is common for all nodes in the network. Thus, the parameter of the exponential distribution, denoted as β can be expressed as:

$$\beta = \frac{1}{E[\triangle T_{(a,b)}]} \approx \frac{1}{\frac{1}{n}\sum_{k=1}^{n} \triangle T^{(k)}_{(a,b)}} \tag{8}$$

Where n is the number of encounters until current time t, and $\triangle T^{(k)}_{(a,b)}$ represents the k^{th} inter-encounter between node a and node b. The adaptation of the mobility characteristics becomes more precise with a greater elapsed time as the historical information becomes more viable.

TABLE 1
Notation

Variables	Description
$Sr(t)$	The source of message i
$Dst(t)$	The destination of message i
T_i	Elapsed time since the creation of the message
Tx	Time-to-live of message i
R_i	Remaining lifetime of the message ($R_i = Tx_i - T_i$)
$n_i(t)$	Number of copies of message i
$m_i(t)$	Number of nodes who have "seen" message i
$s_i(t)$	Number of nodes who have seen message$i$ and their buffers were not full
P_{f_i}	Probability of forwarding message i to very encountered node

4 Proposed Message Scheduling Framework

Figure. 1 provides the whole picture on the DTN message scheduling framework, which illustrates the functional modules and their relations. The summary vector exchange module (SVEM) is implemented at a node during a contact; then the network state estimation module (NSEM) is used to estimate the values of $m_i(T_i)$, $n_i(T_i)$, and $s_i(T_i)$ according to the most updated network

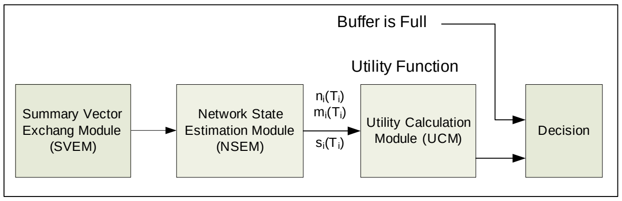
Buffer is Full
Utility Function
Summary Vector Exchang Module (SVEM)
Network State Estimation Module (NSEM)
n_i(T_i)
m_i(T_i)
s_i(T_i)
Utility Calculation Module (UCM)
Decision

Fig. 1. The message scheduling framework

information. The two parameters are further taken as inputs in the calculation of the proposed per-message utility function in the utility calculation module (UCM). The decision of forwarding or dropping the buffered messages is made based on the buffer occupancy status and the utility value of the messages. The rest of the section introduces the details of each functional module.

4.1 Summary Vector Exchange Module (SVEM)

During each contact, the network information summarized as a "summary vector", is exchanged between the two nodes, which includes the following data: (1) statistics of inter-encounter time of every node pair maintained by the nodes, (2) statistics regarding the buffered messages, including their IDs, remaining time to live (R_i), destinations, the stored $n_i(T_i)$, $m_i(T_i)$, and $s_i(T_i)$ values for each message that were estimated in the previous contact. The SVEM ensures the above information exchange process, and activates NSEM for the parameter estimation based on the newly obtained network statistics right after each contact.

4.2 Proposed Network State Estimation Module (NSEM)

The NSEM is used to obtain the estimated $m_i(T_i)$, $n_i(T_i)$, and $s_i(T_i)$ such that the UCM can make decision in the buffer management process. Since acquiring global information about a specific message may take a long time to propagate and hence might be obsolete when we calculate the utility function of the message, we come up with a time-window based estimation approach. Rather than using the current value of $m_i(T_i)$ and $n_i(T_i)$ for a specific message i at an elapsed time T_i, we use the measure of the two parameters over the messages that node a is aware of (has "seen") during an elapsed time T_i. These estimations are then used in the evaluation of the per-message utility.

For this purpose, we propose a novel estimation approach called Global History-Based Prediction (GHP), which estimates the parameters by considering their statistics since the corresponding message was created. Let $M_i(T_i)$, $N_i(T_i)$, and $S_i(T_i)$ denote random variables that fully describe the parameters $m_i(T_i)$, $n_i(T_i)$, and $s_i(T_i)$ at elapsed time T_i, respectively. We have: $E[M_i(T_i)] = \frac{\sum_{i=1}^{j} m_i(T_i)}{j}$, $E[N_i] = \frac{\sum_{i=1}^{j} n_i(T_i)}{j}$, and $E[S_i] = \frac{\sum_{i=1}^{j} s_i(T_i)}{j}$, where j is the total number of messages that have been seen by node a. These messages include the messages stored in the buffer of a that are considered more senior than message i. In the same manner, the average elapsed times for all messages that were generated before message i is calculated as $\widehat{T} = \frac{\sum_{i=1}^{j} T_i}{j}$. Thus, we can have the following estimations for message i: $\widehat{m_i(T_i)}$, $\widehat{n_i(T_i)}$ and $\widehat{s_i(T_i)}$. These values are then incorporated into the per-message utility metrics, which are calculated as $\widehat{m_i(T_i)} = \frac{T_i E[M_i]}{\widehat{T}}$, $\widehat{n_i(T_i)} = \frac{T_i E[N_i]}{\widehat{T}}$ and $\widehat{s_i(T_i)} = \frac{T_i E[S_i]}{\widehat{T}}$.

4.3 Utility Calculation Module (UCM)

Based on the problem settings and estimated parameters, the UCM answers the following question at a node during each nodal contact: Given $n_i(T_i)$, $m_i(T_i)$, $s_i(T_i)$ and limited buffer space for supporting epidemic or two-hop forwarding routing [2], [5], what is an appropriate decision on whether the node should drop any message in its buffer or reject any incoming message from the other node during the contact, such that either the average delivery ratio or delivery delay can be optimized? Section 5 describes in details how the per-message utility function is derived. The scenarios of interest in the study are as following:

- Maximizing the delivery ratio under epidemic routing
- Maximizing the delivery ratio under two-hop forwarding
- Minimizing the average delivery delay under epidemic routing
- Minimizing the average delivery delay under two-hop forwarding

4.4 Forwarding and Dropping Policy

With the per-message utility, the node firstly sorts the buffered messages in a descending manner. The messages with smaller utility values have higher priorities to be dropped when the node's buffer is full, while the messages with higher utility values have higher priorities to be forwarded to a encountered node. Fig. 2 illustrates the forwarding and dropping actions: if the utility U_j of message j (the message with the highest utility value) buffered in a is higher than U_B of message i (the message with the lowest utility value) at node b, then message i is dropped and replaced by a copy of message j if the buffer of b is full during the contact of the two nodes.

5 Utility Function Derivation

5.1 Maximization of Delivery Ratio

Let us assume that the buffer is full at node b and there is a message i with elapsed time T_i in a network that has K messages at the moment at which the decision should be made by a node with respect to dropping a message from all messages in its buffer.

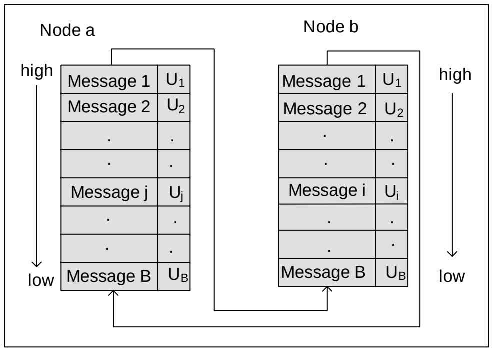

Fig. 2. The forwarding and dropping at a node

5.1.1 Epidemic Forwarding

Theorem 1. *To maximize the average delivery ratio is to drop message i_{min} that satisfies the following:* $i_{min} = argmin_i \left[\left(1 - \frac{m_i(T_i)}{N-1}\right)^2 * \right.$

$$\left(\frac{N}{N - n_i(T_i) + n_i(T_i).e^{\beta N R_i P_{f_i}}}\right)^{m_i(T_i)+1}$$

$$\left. * \left[e^{\beta N R_i P_{fi}} \left(\beta R_i n_i(T_i) + \frac{m_i(T_i)}{N}\right) - \frac{m_i(T_i)}{N} \right] \right] \quad (9)$$

where P_{fi} is the probability of forwarding message i to every encountered node which can be estimated as $P_{fi} = \frac{n_i(t)}{m_i(t)}$.

Proof: The probability that a copy of message i will not be delivered by a node is given by the probability that the next meeting time with the destination is greater than its remaining lifetime R_i, assuming that the message i has not yet been delivered. The probability that message i will not be delivered (i.e., none of its copies will be delivered) can be expressed as $Pr\{message\, i\, not\, delivered \,|\, not\, delivered\, yet\} =$

$$P(T_d < T_i + R_i \,|_{T_d > T_i}) = 1 - \left[\left(1 - \frac{m_i(T_i)}{N-1}\right) \right.$$

$$\left. * \left(\frac{N}{N - n_i(T_i) + n_i(T_i) e^{\beta P_{fi} N R_i}}\right)^{\frac{n_i(T_i)}{P_{f_i}}} \right] \quad (10)$$

The proof of (10) is provided in Appendix. By assuming network homogeneity, there is an equal likelihood that the message is "seen" by each node. Thus, the probability that message i has been already delivered to the destination is equal to

$$Pr\{\, message\, i\, already\, delivered\,\} = \frac{m_i(T_i)}{N-1} \quad (11)$$

By combining (10) and (11), the probability that message i is successfully delivered before its Tx expires can be calculated as follows: $Pr_i = 1 - P\{message\, i\, not\, yet\, delivered\}$

$$*P\{message\, i\, will\, not\, be\, delivered\, within\, R_i\}$$

$$Pr_i = 1 - \left(1 - \frac{m_i(T_i)}{N-1}\right)^2$$

$$* \left(\frac{N}{N - n_i(T_i) + n_i(T_i).e^{\beta P_{fi} N R_i}}\right)^{m_i(T_i)} \quad (12)$$

When a node is operating at its maximum buffer capacity, it should drop one or multiple messages so as to achieve the best gain in the increase of the global delivery ratio $Pr = \frac{1}{K(t)} \sum_{i=1}^{K(t)} Pr_i$. To make the optimal decision locally at the node, Pr_i is differentiated with respect to $n_i(T_i)$, and $\partial n_i(T_i)$ is then discretized and replaced by $\triangle n_i(T_i)$. The best drop policy is one that maximizes $\triangle Pr_i$:
$\triangle Pr_i = \frac{\partial Pr_i}{\partial n_i(T_i)} * \triangle n_i(T_i) =$
$\Big[\big[e^{\beta N R_i P_{f_i}} (\beta R_i n_i(T_i) + \frac{m_i(T_i)}{N}) - \frac{m(T_i)}{N} \big] \left(1 - \frac{m_i(T_i)}{N-1}\right)^2$
$* \left(\frac{N}{N - n_i(T_i) + n_i(T_i) e^{\beta N R_i P_{fi}}}\right)^{m_i(T_i)+1} \Big] \triangle n_i(T_i)$

Thus, the maximum delivery ratio can be achieved if the message that causes the least decrease in $\triangle Pr_i$ is discarded. On the other hand, when message i is discarded, the number of copies of message i in the network decreases by 1, which results in $\triangle n_i(T_i) = -1$. Thus the optimal buffer dropping policy that can maximize the delivery ratio based on the locally available information at the node is to discard the message with the smallest value of $\frac{\partial Pr_i}{\partial n_i(T_i)}$, which is equivalently to choose a message with a value for i_{min} that satisfies (9). This derivation is an attempt to handle changes in the number of copies of a message that may be increased in the future during new encounters. This goal can be achieved by predicting P_f, the probability of forwarding a copy of message i to any node encountered, which is incorporated into the estimation of the delivery ratio. It is clear that the accuracy of P_f is based mainly on the precision in estimating the values of $m_i(T_i)$ and $n_i(T_i)$.

5.1.2 Two-hop Forwarding

Since only L message copies are allowed to be spread by the source node, it is important to estimate the time at which the L message copies have been spread in the network, which is denoted as T_{Li}. Whether the value of T_i is less or greater than T_{Li} plays key role in formulating the utility function. To simplify the notation, we use term T_{Ri} for the period $T_{Li} - T_i$, and T_{XLi} for $Tx - T_{Li}$.

Theorem 2. *The local optimal buffer management policy that maximizes the average delivery ratio is to drop message i_{min} that satisfies the following:*

$$argmin_i =$$

$$\begin{cases} \left[\frac{1}{Pf_i} \left(1 - e^{-\beta P_{fi} T_{Ri}}\right) \left(1 - \frac{m_i(T_i)}{N-1}\right)^2 e^{\beta N T_{Ri}} \right. \\ \left. e^{\frac{1}{Pf_i}(N - n(T_i))(e^{-\beta P_{fi} T_{Ri}} - 1)} \right] e^{-\beta L (T_{XLi})}, \; T_i < T_{Li} \\ \\ \beta P_{fi} R_i \left(1 - \frac{m_i(T_i)}{N-1}\right)^2 e^{-\beta P_{fi} L R_i}, \quad T_i \geq T_{Li} \end{cases} \quad (13)$$

P_{fi} under two-hop forwarding is estimated as $P_{fi} = \frac{s_i(T_i)}{m_i(T_i)}$. Note that the estimation of P_{fi} is different from that of epidemic forwarding since we deal with a controlled flooding case.

Proof: The probability that message i will be delivered (i.e., that n_i copies are delivered) within the remaining lifetime of the message can be expressed by

$$Pr_i\{message\, i\, will\, be\, delivered\, within\, R_i\} =$$

$$P(T_d \leq T_i + R_i \mid_{T_d > T_i}) =$$

$$= \begin{cases} 1 - \left[(1 - \frac{m_i(T_i)}{N-1})e^{\beta N T_{Ri} + \frac{1}{P_{fi}}(N - n(T_i))} \right. \\ \left. e^{(e^{-\beta P_{fi} T_{Ri}} - 1)}\right] e^{-\beta L(T_{XLi})},\ T_i < T_{Li} \\ \\ 1 - \left(1 - \frac{m_i(T_i)}{N-1}\right) e^{-\beta P_{fi} L R_i},\ \ T_i \geq T_{Li} \end{cases} \tag{14}$$

The proof of equation (14) is included in Appendix. Since the node's mobility is iid, the probability that message i has been already delivered is equal to

$$Pr\{\, message\, i\, already\, delivered\, \} = \frac{m_i(T_i)}{N-1} \tag{15}$$

When (14) and (15) are combined, the probability that message i will be delivered before its Tx expires is given by the total probability law as

$$Pr_i = 1 - P\{message\, i\, not\, yet\, delivered\} *$$

$$P\{message\, i\, will\, not\, be\, delivered\, within\, R_i\}$$

$$Pr_i = \begin{cases} 1 - \left[\left(1 - \frac{m_i(T_i)}{N-1}\right)^2 e^{\beta N T_{Ri} + \frac{1}{P_{fi}}(N - n_i(T_i))} \right. \\ \left. e^{(e^{-\beta P_{fi} T_{Ri}} - 1)}\right] e^{-\beta L(T_{XLi})},\ T_i < T_{Li} \\ \\ 1 - \left(1 - \frac{m_i(T_i)}{N-1})\right)^2 e^{-\beta P_{fi} L R_i},\ \ T_i \geq T_{Li} \end{cases} \tag{16}$$

In the case of congestion, a DTN node should drop the message that leads to the best gain in the global delivery ratio. To find the local optimal decision, Pr_i is differentiated with respect to $n(T_i)$ if $T_i < T_{Li}$ and to L otherwise, and ∂n is then discretized and replaced by $\triangle n$. $\triangle Pr_i = \frac{\partial Pr_i}{\partial n} * \triangle n(T_i) =$

$$\begin{cases} \left[\frac{1}{Pf}\left(1 - e^{-\beta P_{fi} T_{Ri}}\right)\left(1 - \frac{m_i(T_i)}{N-1}\right)^2 e^{\beta N T_{Ri}} \right. \\ \left. e^{(e^{-\beta P_{f_i} T_{Ri}} - 1) + \frac{1}{P_{f_i}}(N - n(T_i))}\right] e^{-\beta L(T_{XLi})},\ T_i < T_{Li} \\ \\ \beta P_{f_i} R_i \left(1 - \frac{m_i(T_i)}{N-1}\right)^2 e^{-\beta P_{fi} L R_i},\ \ T_i \geq T_{Li} \end{cases}$$

After the $T_i \geq T_L$, the number of message copies will be subject to be decreased due to discarding the message that has the highest number of message copies. Therefore the second part can be differentiated with respect to L:

$\frac{\partial P}{L} = \beta P_{f_i} R_i \left(1 - \frac{m_i(T_i)}{N-1}\right) e^{-\beta P_{fi} L R_i}$ The optimal buffer dropping policy that maximizes the probability of delivery is thus to discard the message that has the smallest value of $\frac{\partial Pr_i}{\partial n(T_i)}$, that is to choose a message with a value for i_{min} that satisfies (13).

5.2 Minimization of Average Delivery Delay

To minimize the average delivery delay, node a should discard a message such that the expected delivery delay of all messages can be reduced the most. Since the delivery delay of the messages is mainly affected by the nodal enter-encounter time, we assume that all message have infinite or large enough Tx and derive the utility function such that it is affected by number $m_i(T_i)$, $n_i(T_i)$, P_{fi}, and enter-encounter time.

5.2.1 Epidemic Forwarding

Theorem 3. *To achieve the minimum average delivery delay, node a should drop the message that satisfies the following:*

$$i_{min} = \left(1 - \frac{m_i(T_i)}{N-1}\right) *$$

$$\left[\frac{Ln(N).m(T_i)\left[N - 2n_i(T_i)\right]}{\left(Nn_i(T_i) - n_i^2(T_i)\right)^2 \beta}\right] \tag{17}$$

Proof: The expected delay in delivering a message that still has copies existing in the network can be expressed $D_i = P\{message\, i\, not\, deliverd\, yet\} * \frac{1}{P_{fi}} E[T_d \mid T_d > T_i]$

$$D_i = \left(1 - \frac{m_i(T_i)}{N-1}\right) * \frac{1}{P_{fi}} E[T_d \mid T_d > T_i] \tag{18}$$

where

$$E[T_d \mid T_d > T_i] = \left[T_i + \frac{ln(N)}{P_{fi}\beta(N - n_i(T_i))}\right] \tag{19}$$

Proof: The proof of (18) is provided in Appendix.

$$D_i = \left(1 - \frac{m_i(T_i)}{N-1}\right)\left[T_i + \frac{m_i(T_i) ln(N)}{\beta(Nn_i(T_i) - n_i^2(T_i))}\right] \tag{20}$$

When a node buffer is full, the node should make a drop decision that leads to the largest decrease on D_i . To find the local optimal decision, D_i is differentiated with respect to $n_i(T_i)$, and ∂D_i is then discritized and replaced by $\triangle D_i$: $\triangle D_i = \frac{\partial D_i}{\partial n_i(T_i)} * \triangle n_i(T_i)$

$$= \left(1 - \frac{m_i(T_i)}{N-1}\right)\left[\frac{-ln(N).m_i(T_i)\left[N - 2n_i(T_i)\right]}{\left(Nn_i(T_i) - n_i^2(T_i)\right)^2 \beta}\right] \triangle n_i(T_i)$$

To reduce the delivery delay of all messages existing in the network, the best decision is to discard the message that maximizes the total average of the delivery delay, $D = \frac{1}{K(t)} \sum_{i=1}^{K(t)} D_i$, among all the messages. Therefore, the optimal buffer-dropping policy that maximizes the delivery delay is thus to discard the message that has the min value of $\mid \frac{\partial D_i}{\partial n_i(T_i)} \mid$, which is equivalently to choose a message with a value for i_{min} that satisfies (17).

5.2.2 Two-hop Forwarding

Theorem 4. *To minimize the delivery delay of all messages, node i should discard the message that increases the expected delivery delay of all messages. To minimize the average delay of all messages, a node should therefore drop message i_{min} that satisfies the following:*

$$argmin_i = \begin{cases} \frac{1}{P_f}\left(1 - \frac{m_i(T_i)}{N-1}\right)\left[\frac{1}{(n_i(T_i))^2\beta}\right], T_i < T_{Li} \\ \\ \frac{1}{P_{f_i}}\left(1 - \frac{m_i(T_i)}{N-1}\right)\left[\frac{1}{L^2(T_i)\beta}\right], T_i \geq T_{Li} \end{cases} \quad (21)$$

Proof: The expected delay in delivering a message that still has copies existing in the network is

$$D_i = P\{message\, i\, not\, deliverd\, yet\} * \frac{1}{P_{fi}} E[T_d \mid T_d > T_i]$$

$$D_i = \begin{cases} \left(1 - \frac{m_i(T_i)}{N-1}\right)\left[T_i + \frac{1}{P_{f_i}}\left(\frac{1}{(n_i(T_i))\beta}\right)\right. \quad T_i < T_{Li} \\ \\ \left(1 - \frac{m_i(T_i)}{N-1}\right)\left[T_i + \frac{1}{P_{f_i}L\beta}\right], \quad T \geq T_{Li} \end{cases} \quad (22)$$

The proof of (22) is included in Appendix. When a node buffer is full, a node should make a drop decision that leads to the largest decreasing on D_i. To find the local optimal decision, D_i is differentiated with respect to $n(T_i)$ if $T_i < T_{Li}$ and with respect to L otherwise, and ∂D_i is then discritized and replaced by $\triangle D_i$. $\triangle D_i = \frac{\partial D_i}{\partial(n(t)/L)} * \triangle(n(T_i)/L)$. To reduce the delivery delay of all messages in the network, the best decision is to discard the message that maximizes the total delivery delay, $\mid \triangle D \mid$, of all messages. Therefore, the optimal buffer-dropping policy that maximizes the delivery delay is thus to discard the message that has the minimum value of $\mid \frac{\partial D_i}{\partial(n(t)/L)} \mid$, that is to choose a message with a value for $i_{min} = argmin\, \triangle D_i$ that satisfies equation (21). This policy drops a message that has the highest number of message copies within shortest elapsed time since the creation of the message.

6 Simulation Study

6.1 Experimental Setup

To examine the efficiency of the proposed message scheduling approach, we conducted experiments, and presented the results in this section. To better understand the performance of the proposed estimation strategy–GHP, we implemented two other estimation strategies for the values of $m_i(T_i)$, $n_i(T_i)$, and $s_i(T_i)$, namely Global Knowledge-based Management (GKM), and Encounter History-Based Prediction (EHP). GKM assumes the knowledge of exact values of $m_i(T_i)$, $n_i(T_i)$, and $s_i(T_i)$, and is supposed to achieve the best performance as an ideal case. Since such an assumption is not practical [12], the result of GKM is taken as a benchmark for the proposed GHP scheme. With EHP, two encountered nodes update each other with respect to all the messages they have in common, and the values of $m_i(T_i)$, $n_i(T_i)$, and $s_i(T_i)$ are updated accordingly. This policy of update provides a sub-optimal solution and has been employed in [29] and [11]. In addition to the above prediction strategies, we compared the proposed message scheduling approach with three well-known policies listed as follows:

- History-based drop (HBD) [11] is based on the history of all messages (on average) in the network after an elapsed time. The variables of the message utility are estimated by averaging the variables of all messages in the network after the elapsed time.
- Drop oldest (DO) drops the message with the shortest remaining time to live.
- Drop front (DF) drops the message that entered the queue the earliest when the buffer is full. This policy obtains the best performance of all the policies used by Lindgren et al. in [10].

We assume a message issued at a node (termed sourced messages) has the highest priority at the node. If all buffered messages are sourced ones and the newly arrived message is also a source message at the node, then the oldest one is dropped. This idea was examined in [9] and has been proved with improved delivery ratio. To evaluate the policies, a DTN simulator similar to that in [31] is implemented. The simulations are based on two mobility scenarios; a synthetic one based on Random Waypoint mobility model, and a real trace-like mobility model based on a real traces of Zebranet experiment. The real trace was collected as part of the ZebraNet wildlife tracking experiment described in [33], [27]. The mobility under this model is constructed from distributions that match the traces collected from real movements of zebras. The speed and the turning angle selection process are repeated for the whole experimental study duration. The simulation parameters are as shown in table 2. Each node has a transmission range, $D = 30$ meters, to obtain sparsely populated network. The distribution of pause time of RWP model is uniformly distributed within a range, [$T_{min} = 25$, $T_{max} = 40$]. Euclidean distance is used to measure the proximity between two nodes (or their positions). A slotted collision avoidance MAC protocol with Clear-to-Send (CTS) and Request-to-Send (RTS) features was implemented in order to arbitrate between nodes that contend for a shared channel. The message inter-arrival time is uniformly distributed in such a way that the traffic can be varied from low (10 messages generated per node) to high (70 messages generated per node). The buffer size is set to a low capacity (15 messages), to push the network towards a congestion state by increasing the network traffic. We assume sufficient time for completing the possible message exchange for every contact.

Message delivery ratio and the delivery delay are taken as two performance measures. Each data is the average of the results from 30 runs. A PC with Intel 2.0 Ghz Core 2 Duo processor and 2 GB RAM is used for running the simulations.

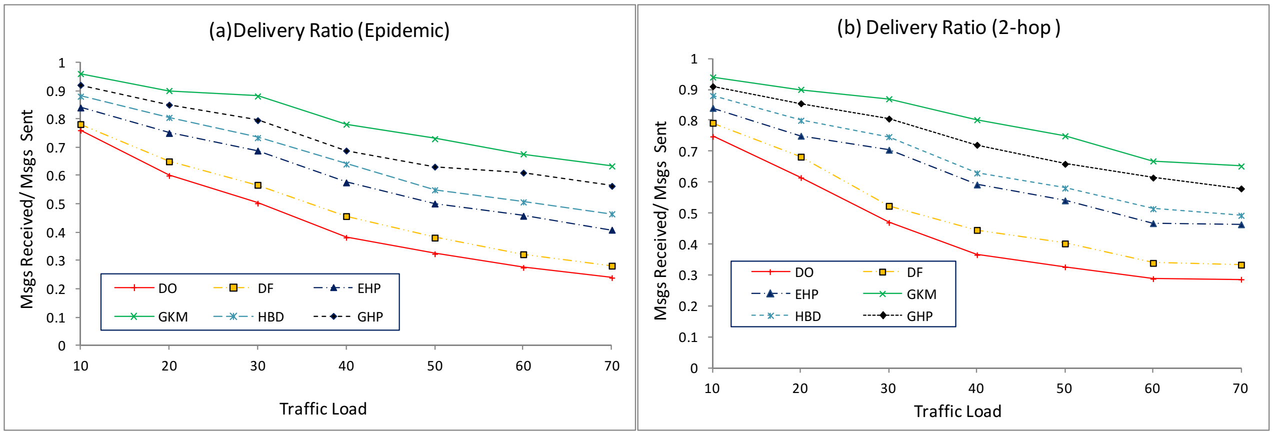

Fig. 3. (a)(b) The effect of traffic load on the delivery ratio under Random Waypoint mobility model

TABLE 2
Simulation parameters

Mobility pattern	RWP	ZebraNet
Simulation duration (hours)	11	15
Simulation area	$800x800m^2$	$1000x1000m^2$
No. of Nodes	130	80
Node speed distribution	$[v_{min} = 2,$ $v_{max} = 5]$	-
TTL (hours)	2	3

6.2 Proposed Policy for Maximizing Delivery Ratio

This section examines the proposed policy for maximizing the average delivery ratio under the considered scenarios. The plots of the delivery ratio obtained for epidemic and two-hop forwarding under random Waypoint mobility model is shown in Fig. 3(a)(b).

It can be seen that GKM gives the best performance under all the traffic loads for both routing schemes due to the complete and global mobility information. The GHP policy provides the next best result and is competitive with the GKM in the case of low traffic. As the traffic increases, the performance of all policies degrades, while the GHP outperforms all other policies except GKM. For epidemic routing, it can achieve a delivery ratio up to 215% higher than that achieved by DO, 70% higher than DF, 22% higher than HBD, 32% higher than EHP, and only 15% worse than GKM. For two-hop forwarding, GHP provides results can be up to 200% higher than DO, 80% higher than DF, 17% higher than HBD, 30% higher than EHP, and 11% worse than GKM. It is noted that all policies obtain better results for 2-hop forwarding than that for epidemic routing, which proved that the controlled flooding mechanism improves performance in the case of limited buffer capacity. Fig. 4 shows the results of delivery ratios under Zebranet trace. As can be seen, GHP can achieve a delivery ratio 249% higher than that achieved by DO, 93% higher than DF, 16% higher than HBD, 29% higher than EHP, and only 13% worse than GKM. For two-hop forwarding, GHP provides results can be 210% higher than DO, 73% higher than DF, 15% higher than HBD, 28% higher than EHP, and 9% worse than GKM.

6.3 Proposed Policy for Minimizing Delivery Delay

This subsection evaluates the effect of the policy of each routing scheme on message delivery delay using the same scenarios in the previous section.

Figure. 5 shows the results. Similarly, the GKM gives the best performance under all traffic loads for both routing techniques, while the GHP is the second best and is competitive with the GKM in the case of low traffic. As the traffic increases, the demand on the wireless channel and buffers increases, causing a long queuing delays and substantial message loss that negatively affect the performance of all the examined policies. We have observed that for both routing schemes the GHP outperforms all other policies. GHP under epidemic routing is better than DO by 42%, DF by 53%, HBD by 10%, EHP by 20%, and a longer delay of only 8% of that achieved by GKM.

Under two-hop forwarding, GHP can reach delivery delays up to 57% shorter than DF, 44% shorter than DO, 17% shorter than HBD, 27% shorter than EHD, and only 10% longer than GKM. Figure 6. shows the results of delivery delay under ZebraNet trace. As can be seen, GHP under epidemic routing is better than DF by 81%,DO by 71%, HBD by 15%, EHP by 24%, and a longer delay of only 11% of that achieved by GKM. Under two-hop forwarding, GHP can reach delivery delays up to 66% shorter than DF, 53% shorter than DO, 14% shorter than HBD, 22% shorter than EHD, and only 12% longer than GKM.

6.4 Additional Complexity due to GHP

It is clear that a DTN form a distributed system for global dissemination of network states and is subject

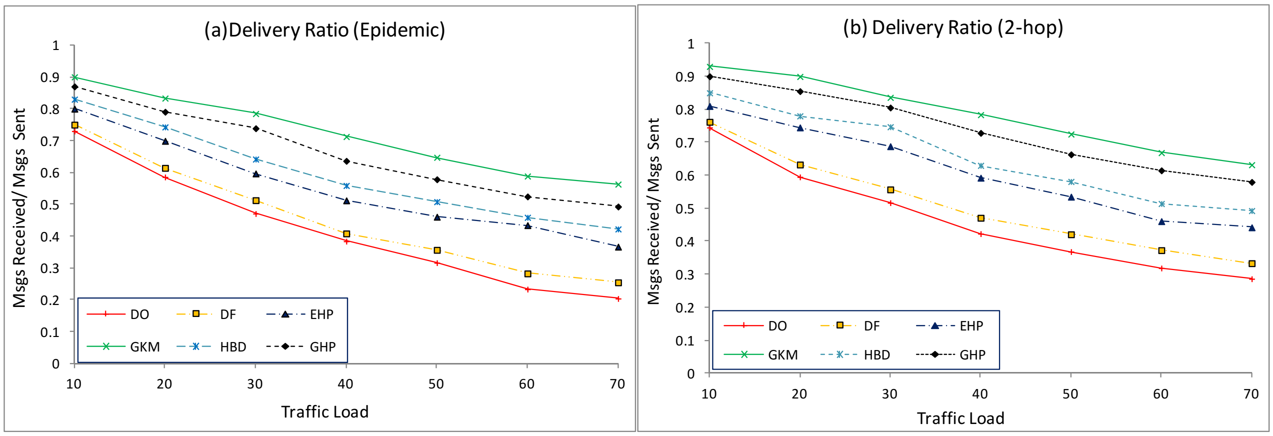

Fig. 4. (a)(b) The effect of traffic load on the delivery ratio under the Zebra trace

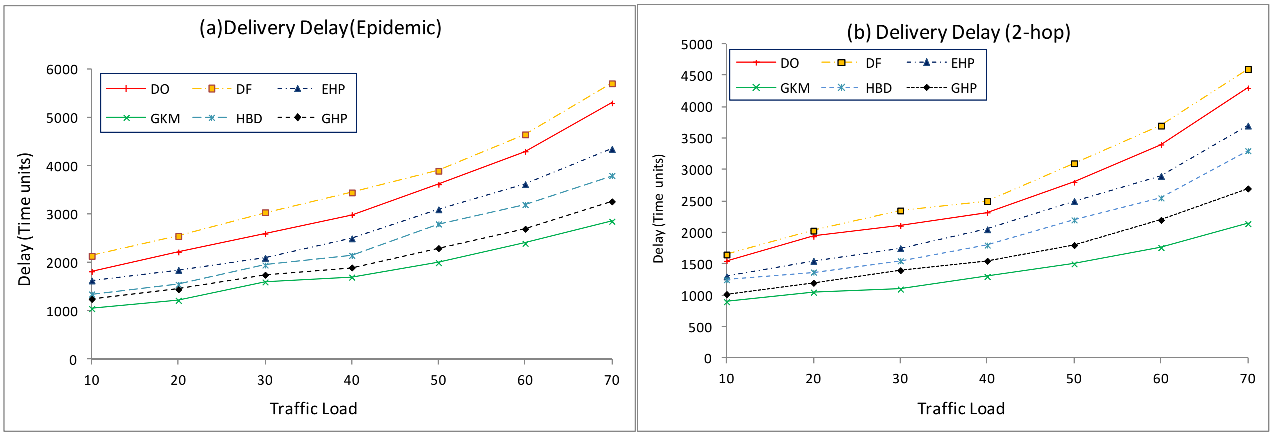

Fig. 5. (a)(b) The effect of traffic load on delivery delay under Random Waypoint Mobility model

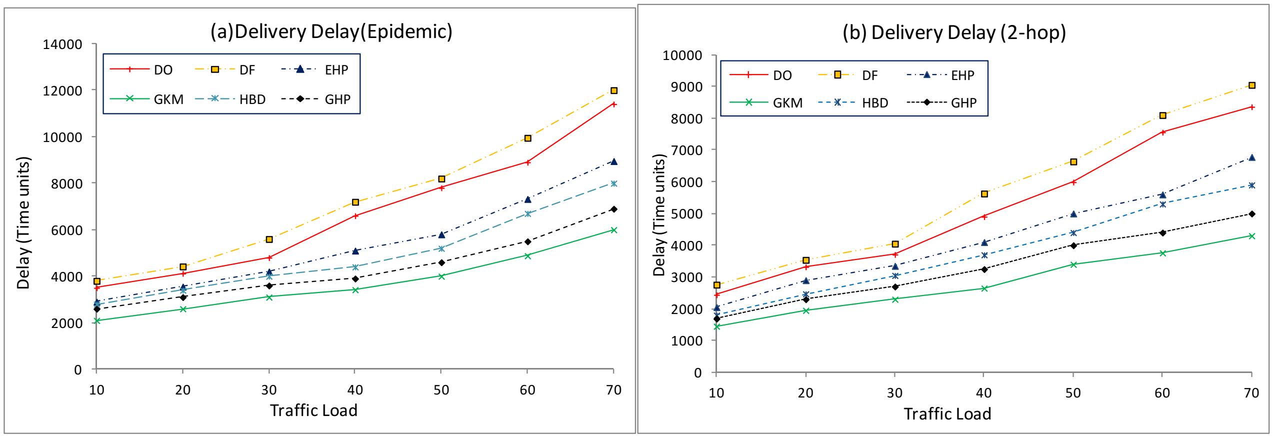

Fig. 6. (a)(b) The effect of traffic load on delivery delay under the ZebraNet trace

to percolation that could impair the precision of the global statistics estimation and cause additional operational complexity. The study is interested in the scenarios where the DTNs have densely distributed nodes and the encounter frequency is relatively high. Therefore, the imprecision caused by percolation should yield quite limited impact to the system stability and performance, which will further be proved via extensive simulation.

There is some operation complexity caused by the proposed scheme. One is the additional operational complexity due to information exchange, which nonetheless could be minimal because the information exchange is performed on top of the message exchange during each nodal contact. On the other hand, the additional computation complexity is considered the main source of overhead that drains nodal energy. To examine the relation between the additional computation complexity and the performance gain, the following paragraphs provide our analysis.

6.4.1 Relation between Computation and Performance Gain

It is clear that GHP outperforms their counterparts thanks to the adaptive calculation of utility values using a couple of global network state parameters, i.e., $m(T)$ and $n(T)$. Such performance gain, nonetheless, is at the expense of larger computation complexity, which in turn causes longer computation time. We evaluate the proposed message scheduling approach as follows in terms of the impact due to computation complexity.

It is clear that the main source of computation complexity lies in calculating the utility function, which is in turn dominated by the number of messages involved in the solution process. Another fact is that, considering more messages is expected to yield better precision in the utility function calculation (so as for the overall performance) at the expense of longer computation time. Let *Sampling List* (SL) be the subset of randomly selected messages for consideration in the utility function calculation at a node. The size of SL stands for the amount of statistics collected at the node to make the message forwarding/dropping decision. Our strategy is to get the relation between the performance and the size of subset, and then the relation between the size of subset and the computation time. Thus we will be able to observe the performance gain due to the longer computation time compared with its counterparts.

To examine the desired scenario with high congestion, the buffer size is set to 20, and traffic load is set high (90 generated message per node). Without loss of generality, this scenario is performed under epidemic routing using Random Waypoint mobility model.

The performance impact on GHP by reducing the amount of collected statistics is shown in Fig. 7. It is shown that increasing the SL size in GHP results in the corresponding performance improvement over EHP.

When the SL size is 1, GHP is degraded as EHP since it becomes completely based on local information. As the SL size is increased, the performance of GHP improves considerably. Fig. 8 shows the relation between the computation complexity and the SL size.

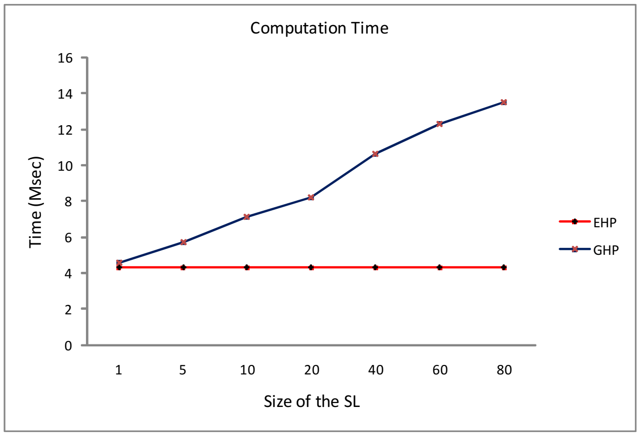

Fig. 8. The effect of SL size on the policy computation time

Note that HBD and GHP under a unlimited SL size yield 6 and 7.5 times of longer computation time than that by GHP at SL size of 40 messages, respectively, which are not shown in the chart. The very long computation time is due to the fact that all the messages that a node has been learned from all encountered nodes are considered in the calculation of the utility function.

Our results suggest that, with a carefully designed statistics collection strategy, the proposed GHP scheme can be manipulated to achieve a graceful tradeoff among the computation time (which is directly related to nodal power consumption) and performance according to any desired target function. In addition, we have seen that GHP only takes a small fraction of computation time compared to the case by maintaining a complete view on all the messages older than message i, while without significantly affecting the performance.

7 Conclusions

This paper has investigated a novel message scheduling framework for epidemic and two-hop forwarding routing in homogeneous delay tolerant networks (DTNs), aiming to optimize either the message delivery ratio or message delivery delay. The proposed framework incorporates a suite of novel mechanisms for network state estimation and utility derivation, such that a node can obtain the priority for dropping each message in case of buffer full. Using simulations based on two mobility models; a synthetic (Random WayPoint) and a real trace-model (ZebraNet), the simulation results show that the proposed buffer management policies, named GHP, can significantly improve the routing performance in terms of the performance metrics of interest under limited network information.

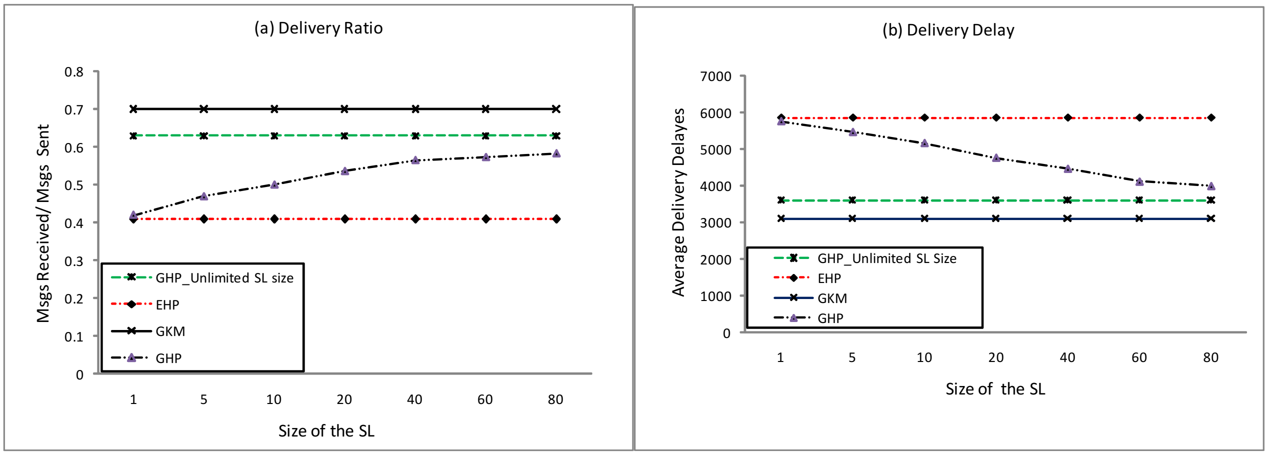

Fig. 7. (a)(b) Impact of SL size on the delivery ratio and delivery delay

APPENDIX

Proof of (10): Given $m_i(T_i)$, $n_i(T_i)$,and $P(T_i) = \frac{m_i(T_i)}{N-1}$, as initial values at T_i , the delivery probability in the interval $t : T_i < t < T_i + R_i$, $P(T_d < T_i + R_i \mid_{T_d > T_i})$, can be constructed using (3) as follows: $P_i'(t) = \frac{dP}{dt} = \beta n_i(t)(1 - P_i(t))$ $\frac{dP}{1-p} = \beta n_i(t)dt$ $\frac{dP}{1-p} = \beta \frac{N n_i(0)}{n_i(0)+(N-n_i(0))e^{-P_{fi}\beta Nt}} dt$ Integrate both sides for the interval R_i, we get

$$P(T_d < T_i + R_i \mid_{T_d > T_i}) = 1 - \left[\left(1 - \frac{m_i(T_i)}{N-1}\right)\right.$$

$$\left. * \left(\frac{N}{N - n_i(T_i) + n_i(T_i)e^{\beta P_{fi} N(Ri)}}\right)^{\frac{n_i(T_i)}{P_f}}\right]$$

Proof of (14): Delivery probability within R_i and the initial state is at T_i Calculating T_L value: *Given* $n_i(T_i)$ we can expect the time, T_L, at which L message copies in the network are spread as following: $n(T_{L-}T_i) = N - (N - n(T_i))e^{-\beta P_f(T_{L-}T_i)}$ $\frac{N-L}{N-n(T_i)} = e^{-\beta P_f(T_{L-}T_i)} \rightarrow$ $-\beta P_f(T_{L-}T_i) = Ln\frac{N-L}{N-n(T_i)}$ $T_L - T_i = \frac{1}{\beta P_f} Ln(\frac{N-n(T_i)}{N-L})$ $T_L = T_i + \frac{1}{\beta P_f} Ln(\frac{N-n(T_i)}{N-L})$

- Delivery within R_i and $P(T_i) = \frac{m_i(T_i)}{N-1}$,

Two cases are identified: (1) $T_i < T_{L_i}$, (2) $T_{Li} \leq T_i \leq R_i$. *Case (1):* $T_i < T_{Li}$, which has two periods $(T_i, T_L)$and $(T_L, Tx - T_{Li})$ 1-Period (T_i, T_{Li}) : We have: $\frac{dP(t)}{dt} = P'(t) = \beta n(t)(1 - P(t))$ with initial conditions $P(T_i)$ and $n(T_i)$. $\frac{dP}{1-P} = \beta \left[N - (N - n(T_i))e^{-\beta P_f t}\right]$. Integrating both sides for the interval $T_{Li} - T_i$, we get

$$P(T_d \leq T_{Li} \mid T_d \geq T_i) = 1 - \left[(1 - P(T_i))\, e^{\beta N(T_{Li}-T_i)}\right.$$

$$\left. * e^{\frac{1}{P_f}(N-n(T_i))e^{-\beta P_f(T_{Li}-T_i)} - \frac{1}{P_f}(N-n(T_i))}\right], \ \ T_i < T_{Li}$$

2-For period (T_L, Tx): $\frac{dP(t)}{dt} = P'(t) = \beta L(1 - P(t))$ $\frac{dP}{1-P} = \beta L dt$. Integrating both sides for the interval $Tx - T_{Li}$, we get $P(Tx - T_{Li}) = 1 - \left[(1 - \frac{m_i(T_i)}{N-1})e^{\beta N(T_{Li}-T_i)}\right.$ $\left. * e^{\frac{1}{P_f}(N-n(T_i))e^{-\beta P_f(T_{Li}-T_i)} - \frac{1}{P_f}(N-n(T_i))}\right] e^{-\beta L(Tx-T_{Li})}$

Therefore the total delivery probability at $T_i < R_i$ is given by $P(T_{Li} < T_d \leq Tx - T_{Li}) = 1 - \left[(1 - \frac{m_i(T_i)}{N-1})e^{\beta N(T_{Li}-T_i)}\right.$ $\left. * e^{\beta \frac{1}{P_f}(N-n(T_i))(e^{-\beta P_f(T_{Li}-T_i)}-1)}\right] e^{-\beta L(Tx-T_{Li})}$

Case (2): $T_i \geq T_{Li}$ The initial condition $P(T_i) = \frac{m_i(T_i)}{N-1}$. $\frac{dP(t)}{dt} = P'(t) = \beta P_f L(1 - P(t))$ (multiply by P_f to consider the situation the message with L copies is most likely get dropped) $\frac{dP}{1-P} = \beta L dt$. Integrating both sides for the interval $Tx - T_i = R_i$: $\int_{P(T_i)}^{P(Tx)} \frac{1}{1-P} dp = \beta P_f L \int_0^{R_i} dt$ The final expression is: $P(T_i < T_d < T_i + R_i) = 1 - \left[1 - \frac{m_i(T_i)}{N-1}\right] e^{-\beta P_f L R_i}$

Proof of (18): $E[T_d \mid T_d > T_i] = T_i + \int_0^{\propto}(1 - (P(t))dt$

$$E[T_d \mid T_d > T_i] = T_i + \int_0^{\propto} \left(\frac{N}{N-n(T_i)+e^{\beta P_f NTi}}\right)^{\frac{1}{p_f}} dt$$

$$E[T_d] = \frac{-1}{\beta N}\left((N - n(T_i))e^{-\beta P_f Nt}\right)^{\frac{1}{P}}$$

$$* \left(\frac{N}{N - n(T_i) + e^{\beta P_f Nt}}\right) .F \mid_0^{\infty}$$

According to the saddle point approximation [9], the final formula is obtained as $E[T_d] = \frac{lnN}{\beta P_f(N-n(T_i))}$. The expected delivery delay at any elapsed time instance

$$E[T_d \mid T_d > T_i] = T_i + \frac{Ln(N)}{\beta P_f(N - n(T_i))}$$

Proof of (22): $D_i = P\{message\, i\, not\, deliverd\, yet\} * \frac{1}{P_f} E[T_d \mid T_d > T_i]$.

$E[T_d \mid T_d > T_i] = \left[T_i + \int_0^{\infty} tf(t)dt\right]$

since the mobility and nodes are exponentially distributed, the respected delay of a message can be calculated as $\left[T_i + \int_0^{\infty} t\beta e^{-\beta t}dt\right] = T_i + \frac{1}{\beta}$. In case of a message is carried by L nodes, $E[T_d] = T_i + \frac{1}{L\beta}$. In case of a message is carried by $n_i(T_i) < L$, with message reaming life time R_i, $E[T_d] = T_i + \frac{1}{n_i(T_i)\beta}$.

The final formula of (22) is derived by combining the the above equations with probability of a message not yet delivered for either cases $T_i < T_L$ or $T_i \geq T_L$.

$$D_i = \begin{cases} \left(1 - \frac{m_i(T_i)}{N-1}\right) \left[T_i + \frac{1}{P_f}\left(\frac{1}{n_i(T_i)\beta}\right)\right., & T_i < T_L \\ \\ \left(1 - \frac{m_i(T_i)}{N-1}\right) \left[T_i + \frac{1}{P_f L\beta}\right], & T \geq T_L \end{cases}$$

References

[1] “Delay tolerant networking research group,” http://www.dtnrg.org.

[2] A. Vahdat and D. Becker, “Epidemic Routing for Partially Connected Ad Hoc Networks,” *Technical Report CS-200006, Duke Univ., Apr. 2000.*

[3] A. Lindgren, A. Doria and O. Schelén, “Probabilistic Routing in Intermittently Connected Networks,” *SIGMOBILE Mobile Computing Comm. Rev., vol. 7, no. 3, pp. 19-20, July 2003.*

[4] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, “Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case,” *IEEE Trans. Networking, vol. 16, no. 1, pp. 63-76, Feb. 2008.*

[5] R. Groenevelt, P. Nain, and G. Koole. 2005, “Message delay in MANET,” *Proc. ACM SIGMETRICS, 2005.*

[6] A. Elwhishi and P. Ho, “SARP - a novel multi-copy routing protocol for intermittently connected mobile networks,” *Proc. IEEE GLOBECOM, 2009.*

[7] V. Erramilli, M. Crovella, A. Chaintreau , C. Diot, “ Delegation forwarding,” *Proc. ACM MobiHoc, 2008.*

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

[9] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, “Performance modeling of epidemic routin,” *Proc. IFIP Networking, 2006.*

[10] A. Lindgren and K. Phanse, “Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks,” *First International Conference on Communication System Software and Middleware, 2006.*

[11] A. Krifa, C. Barakat, and T. Spyropoulos, “Optimal buffer management policies for delay tolerant networks,” *Proc. IEEE SECON, 2008.*

[12] S. C Nelson, M. Bakht, R. Kravets, and Al. F. Harris, “Encounter: based routing in DTNs,” *Proc. IEEE INFOCOM, 2009.*

[13] T. Spyropoulos, T. Turletti, and K. Obraczka, “Routing in delay-tolerant networks comprising heterogeneous node populations,” *IEEE Trans. Mobile Computing, vol. 8, no. 8, pp. 1132–1147, August 2009.*

[14] D. Aldous and J. Fill, “Reversible markov chains and random walks on graphs. (monograph in preparation.),” http://statwww.berkeley.edu/users/aldous/RWG/book.html.

[15] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Performance analysis of mobility-assisted routing,” *Proc. ACM MOBIHOC, 2006.*

[16] M. V. Thomas Karagiannis, J. Le Boudec, “Power law and exponential decay of inter contact times between mobile devices,” *Proc. ACM/IEEE MobiCom, 2007.*

[17] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. “The broadcast storm problem in a mobile ad hoc network”. *Wireless Networks, 8(2/3), 2002.*

[18] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: Efficient routing in intermittently connected mobile networks” *Proc. ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN), 2005.*

[19] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, and K. Scot, “Delay-tolerant networking: an approach to interplanetary internet” *IEEE Communications Magazine, 2003.*

[20] “Disruption tolerant networking,” http://www.darpa.mil/ato/solicit/DTN.

[21] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Pocket switchednetworks: Real-world mobility and its consequences for opportunistic forwarding,” *Technical Report UCAM-CL-TR-617, University of Cambridge, 2005.*

[22] T. Small and Z. J. Haas , “Resource and performance tradeoffs in delay-tolerant wireless networks,” *Proc. ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN’05), August 2005.*

[23] Q. Li and D. Rus, “Communication in disconnected ad hoc networks using message relay. *Journal of Parallel and Distributed Computing, 63(1), 2003.*

[24] K. A. Harras and K. C. Almeroth, “Transport layer issues in delay tolerant mobile networks,” *Proc. IFIP-TC6 Networking, 2006.*

[25] J. A. Davis, A. H. Fagg, and B. N. Levine, “Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks,” *Proc. ISWC, 2001.*

[26] J. Widmer and J.-Y. L. Boudec, “Network coding for efficient communication in extreme networks,” *Proc. ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN), 2005.*

[27] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, “Energy- efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet,” *Proc. ASPLOS, 2002.*

[28] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, “Low-cost communication for rural internet kiosks using mechanical backhaul,” *Proc. MobiCom, 2006.*

[29] A. Balasubramanian, B. N. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. In Proceedings of ACM SIGCOMM, 2007.

[30] S. C Nelson, M. Bakht, R. Kravets, and A. F. Harris, “Encounter: based routing in DTNs” *SIGMOBILE Mob. Comput. Commun. Rev. 13, 1 (June 2009), 56-59.*

[31] “DTN java simulator,” *http://people.ee.ethz.ch/ spyropot/dtnsim.html.*

[32] H. P. Dohyung Kim and I. Yeom, “Minimizing the impact of buffer overflow in dtn,” *Proc. International Conference on Future Internet Technologies (CFI), 2008.*

[33] Y. Wang, P. Zhang, T. Liu, C. Sadler and M. Martonosi, "Movement Data Traces from Princeton ZebraNet Deployments," *CRAWDAD Database.*

[34] G. Fathima, and R.S.D. Wahidabanu, “ A new queuing policy for delay tolerant networks,” *Int. J. Computer. Application, vol. 1, no. 20, 2010, pp. 56-60.*

[35] V. Erramilli, M. Crovella, “Forwarding in opportunistic Networks with Resource constraints,” *Proc. ACM CHANTS, 2008.*

[36] E.Daly and M.Haahr, “Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs” *Proc. ACM MobiHoc, 2007.*

[37] J. Freyne, S. Berkovsky, E. M. Daly, and W. Geyer, “ Social networking feeds: recommending items of interest,” *Proc. ACM RecSys, 2010.*

[38] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: Social-based forwarding in delay tolerant networks,” *Proc. ACM MobiHoc, 2008.*

[39] S. Dimitriou and V. Tsaoussidis, "Effective buffer and storage management in DTN nodes," *Proc. ICUMT, 2009.*

[40] J. Hillston. 2005, “Fluid Flow Approximation of PEPA models,” *Proc. QEST,2005.*

[41] T. Abdelkader, K. Naik, A. Nayak, and N. Goel, "A Socially-Based Routing Protocol for Delay Tolerant Networks", *Proc. GLOBECOM, 2010.*

[42] T. Spyropoulos, A. Jindal, and K. Psounis, “An analytical study of fundamental mobility properties for encounter-based protocols,” *Inter- national Journal of Autonomous and Adaptive Communications Systems, vol. 1, no. 1, 2008, pp. 4–40.*