HUBCODE: Hub-based Forwarding Using Network Coding in Delay Tolerant Networks

Shabbir Ahmed Salil S. Kanhere
University of New South Wales, Australia
{shabbira,salilk}@cse.unsw.edu.au

Technical Report
UNSW-CSE-TR-1009
March 2010

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Most people-centric delay tolerant networks have been shown to exhibit power-law behavior. Analysis of the temporal connectivity graph of such networks reveals the existence of *hubs*, a fraction of the nodes, which are collectively connected to the rest of the nodes. In this paper, we propose a novel forwarding strategy called *HubCode*, which seeks to use the hubs as message relays. The hubs employ random linear network coding to encode multiple messages addressed to the same destination, thus reducing the forwarding overheads. Further, the use of the hubs as relays, ensures that most messages are delivered to the destinations. Two versions of HubCode are presented, with each scheme exhibiting contrasting behavior in terms of the computational costs and routing overheads. We formulate a mathematical model for message delivery delay and present a closed-form expression for the same. We validate our model and demonstrate the efficacy of our solutions in comparison with other forwarding schemes by simulating a large-scale vehicular DTN using empirically collected movement traces of a city-wide public transport network. Under pragmatic assumptions, which account for short contact durations between nodes, our schemes outperform comparable strategies by more than 20%.

1 Introduction

Delay Tolerant Networks (DTN) are a type of challenged networks, wherein the contacts between the communicating devices are intermittent. Consequently, a contemporaneous end-to-end path between the source and destination rarely exists. Of particular interest, are the networks that are formed by people in urban environments. These include: (i) *Pocket Switched Networks* [18], wherein personal communication devices carried by humans self-organize to form an intermittently connected network and (ii) Vehicle-based DTN [4, 12] , in which, WiFi routers mounted on vehicles can communicate with each other.

Message forwarding is one of the most challenging aspects of DTN due to the inherent intermittent connectivity. However, knowledge of fundamental properties of the underlying network can be helpful in making better forwarding decisions. In particular, the aforementioned people-centric networks have been shown to follow *power-law* behavior [18, 6, 41]. In such networks, a small percentage of the nodes, often referred to as *hubs* [40], are known to have significantly higher connectivity (i.e., high node degree) as compared to the rest of the nodes. Consequently, most nodes can be reached from every other node by a small number of hops, via the hubs.

A few forwarding schemes have been proposed, which exploit these power-law properties [19, 23]. The general idea is to rank nodes based on popularity metrics such as node centrality. A node then forwards a message to another node, if the latter has a higher rank that the former. These schemes have been shown to perform effectively, under the assumption that the nodes can exchange unlimited data in an encounter. However, in reality, contact durations between nodes in people-centric DTN are often only few seconds long [18]. Consequently, the most popular nodes, which concentrate all of the forwarding traffic, can often only exchange limited number of messages and in effect act as bottlenecks in the forwarding process.

In this paper, we seek to address this particular problem by employing the theory of network coding [1], which has been shown to attain maximum information flow in a network. We propose a novel forwarding strategy, *HubCode*, which exploits the power-law properties of the network by directing all forwarding traffic to the hubs. In other words, the hubs form a *data conduit*. Messages are then forwarded within the data conduit (i.e. only among hubs) using random linear network coding, wherein multiple messages addressed to the same destination are combined to form a single encoded message. Since randomly selected coefficients are used in the coding process, each encoded message is useful to the destination, thus reducing the propagation of redundant messages.

In the basic version of HubCode, the hubs exchange the coefficient matrices of the encoded messages prior to data exchange in order to select the messages for forwarding. The resulting overhead, which is $O(n^2)$ for n messages, can be fairly significant. As a result, during short contact durations, the hubs may not get a chance to forward coded messages, since most of the contact opportunity is used for exchanging coefficients. To reduce this overhead, we propose an alternate approach, wherein, the hubs do not exchange the entire coefficient matrices, but rather only exchange a list of native messages. The resulting overhead is just $O(n)$. However, the hubs now need to decode the messages (i.e. solve linear equations), which is computationally expensive. On the contrary, in the basic version, only the destination decodes the messages, thus simplifying the processing at the hubs. These two versions address the important trade-off between routing overhead and computational complexity.

We evaluate the performance of our proposed schemes and compare them with other forwarding protocols using traces collected from a large-scale (> 1000 nodes) real-world bus-based DTN. Under realistic assumptions, which account for the limited data exchange possible during short encounters, our schemes achieve 20% higher delivery ratio than comparable strategies. In addition, our schemes achieve about 50% savings in delivery cost. Empirical evaluations also suggest that our schemes are more resilient against random node failure compared to other protocols. Besides, comprehensive analysis have been carried out in order to examine the effect of varying traffic loads, message lengths and hub sizes on delivery performance of HubCode.

We have also formulated a mathematical model to estimate the message delivery delay. Closed-form expressions are presented for our proposed schemes which serve as an upper bound in estimating message delivery delay. Simulation results corroborate our analytical formulation especially when there are sufficient number of hubs that act as message relays.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 presents the details of the HubCode schemes. The mathematical model to estimate message delivery delay is described in Section 4. In Section 5, we present the results from our simulations and finally, Section 6 concludes the paper. A preliminary version of this work has been presented in [2].

2 Related Work

Ahlswede et al. [1] first introduced the theory of network coding and showed that it can achieve maximum information flow in a network, in the context of multicasting. In recent years, researchers have demonstrated that network coding can improve the throughput in wireless networks for unicast [22, 10, 27, 24] as well as broadcast transmissions [39]. Lun et al. [27] and Li et al. [24] present theoretical results on the application of network coding for unicast transmissions. The work presented in [22] and [10] focuses on practical issues. They demonstrate that network coding can benefit from leveraging the broadcast advantage in wireless networks. In [22], the authors also present empirical results from testbed deployments and show that their proposed method can increase the throughput several folds. However, their methods are suited for densely connected networks such as mesh networks, where the nodes can overhear their neighbors' transmissions. Consequently, these schemes are not effective for intermittently connected networks such as DTN.

A few papers [38, 42, 25, 7] have studied the use of network coding in DTN. Zhang et al. [42] and Widmer et al. [38] have studied the benefits of using Random Linear Coding (RLC) for unicast transmissions in DTN. RLC uses simple flooding to distribute the messages in the network. However, rather than transmitting the native messages, a node combines these messages to form an encoded message and forwards this encoded message to its neighbors. The coefficients used in the encoding process are also transmitted along with the message. The messages are only decoded at the destination, when it receives sufficient number of encoded messages (n linearly independent encoded messages are required to decode n messages). Our proposed scheme also employs network coding for forwarding messages. However, there are two key differences. First, instead of flooding the encoded messages in the network, we leverage the power-law properties of the network and only choose a small fraction of the nodes which have high connectivity (i.e. hubs), as the relay nodes. Second, only the hubs are responsible for coding messages.

In recent years, several researchers [3, 16, 18, 17, 41] have analyzed the properties of people-centric DTN using empirically collected traces. They have found that in all these networks, a small percentage of *popular* nodes are connected to most of the other nodes. In other words, the degree distribution follows a power-law. Freeman [13] defined several centrality metrics to measure the importance of a node in a network. Researchers in [19, 23] have proposed forwarding strategies that exploit the existence of the scale-free structure in the underlying network. In BubbleRap [19], nodes are formed into communities and also ranked according to their centrality. Both global and community rankings are used to find suitable forwarders by using a gradient forwarding approach. Similar ideas are proposed in [23], where each node is assigned a quality metric based on its popularity. Gradient forwarding is then employed. In our work, we also make use of the popular nodes (called hubs) as relay nodes. However, unlike these schemes, which employ gradient forwarding, in HubCode, messages are disseminated amongst the hubs using network coding.

Mathematical modeling of DTN forwarding strategies is a mature field of research. In particular, forwarding schemes based on epidemic forwarding principles have been extensively analyzed in [15, 33, 14, 43, 34]. It is commonplace to use differential equations for modeling system dynamics in DTN [43], due to their simplicity as compared to Markov chains. For example, the system dynamics of forwarding a single packet have been modeled using Ordinary Differential Equations (ODE) [43]. In [26], the authors have extended this to account for a batch of packets for both replication and network coding based forwarding. There also exist some efforts which analyze the performance of 2-hop relay protocols using redundant copies [32, 31]. As mentioned earlier, these works are based on epidemic principle where a node distributes its messages to any other nodes it meets, treating all nodes equally. Consequently, it is difficult to adopt these techniques for modeling our proposed schemes, which differentiate between nodes based on their encounter patterns (hubs vs normal nodes). Instead, in this paper, we use the network model proposed in [14] for studying routing in mobile ad hoc networks. In this model, the characteristics of ad hoc network is captured through a single parameter: the *inter-contact* rate between nodes. We have amended the model to incorporate the effects of network coding and the forwarding policies of our hub-based forwarding scheme in a simplified manner.

3 HubCode

As highlighted in the introduction, empirical analysis of the mobility patterns of several people-centric DTN [18, 6, 41] have revealed that the degree distribution of the network graph follows a power-law. This implies the existence of a small percentage of *hubs*, which are individually connected to a large number of nodes as compared to other nodes. Further, collectively, the hubs are connected with most of the other nodes in the network (i.e. they achieve nearly 100% coverage). Motivated by these properties, we propose a novel forwarding strategy called

HubCode, which uses the hubs as message relays. The hubs are identified by analyzing historical movement patterns of the nodes (e.g. in this paper, we have identified the hubs based on their node degrees). Since, most people-centric networks exhibit significant repeatability (e.g., most people have the same daily routine, buses follow the same schedule), this classification of nodes is reasonably time-invariant. Also, if the network characteristics change, the new set of hubs can be readily identified by repeating the analysis.

All traffic in the network is forwarded to the hubs. Since, each hub concentrates significant traffic, we propose the use of network coding at the hubs to encode multiple messages (addressed to the same destination) into a single encoded message. A hub forwards an encoded message to a neighboring hub if this message is linearly independent with the encoded messages carried by the neighbor. The use of network coding results in significant savings in bandwidth, since a single encoded message is forwarded in place of multiple native messages. Further, since the hubs collectively have contact opportunities with all other nodes, most of the messages can be delivered to the destination.

We first present the basic version of our scheme, *HubCodeV1*, which makes use of the traditional approach to network coding [42]. We argue that this scheme requires the hubs to exchange significant auxiliary information. Next, we present an alternate approach, *HubCodeV2*, which, requires the intermediate hubs to decode the coded messages (in addition to the normal encoding operations). As a result, the hubs only need to exchange message IDs, which reduces the auxiliary data overhead. However, since the hubs decode messages, the computational complexity increases.

3.1 HubCodeV1

In our schemes, message forwarding is a simple three step process: 1) Source nodes forward messages to a hub, 2) a hub encodes multiple messages headed to the same destination and disseminates the encoded messages among other hubs and 3) a hub delivers the encoded message to the destination. To simplify the explanation, we classify nodes into 3 groups: (1) source, (2) destination and (3) hubs and provide a detailed description of the tasks undertaken by each category of node. Note that, a source or destination can also be hub, but for simplicity, we assume the groups are mutually exclusive.

Source

When a source encounters a hub, it creates a copy of the message and forwards the copy to the hub. Recall, that the hub nodes are appropriately labeled by analyzing past behavior of the network. If the source carries a single native message, it is forwarded as-is. However, if more than one message are destined to same address, then the source combines them into a single encoded message using linear network coding (Eq. 3.1), and forwards the encoded message to the hub. The coding technique is described below.

Hubs

When two hubs encounter each other, they first exchange certain auxiliary information, that is used to decide if the hubs should forward messages to each other (these details are explained later). If a hub needs to forward messages to another hub, it encodes all messages with a common destination using random linear coding and forwards the single encoded message. This results in significant savings in the bandwidth. Assume that a hub currently has k messages, $X_1,\ X_2,\ \cdots,\ X_k$ with a common destination. Then the hub creates a linear combination [9, 42] of these k messages to form a single encoded message F_l, using Eq. 3.1,

$$F_l = \sum_{i=1}^{k} a_i X_i,\ \ a_i \epsilon \mathbb{F}_q \tag{3.1}$$

where $a_1,\ a_2,\ \cdots,\ a_k$, represent the coefficients, which are randomly selected from a finite field [29], $\mathbb{F}_q$ where $q = 2^{16}$. All the additions and multiplications are performed over the finite field $\mathbb{F}_q$, so that the encoded message has the same size as the native message. The coefficients a_i and the message IDs (idx_i) of all the native messages are appended to the encoded message prior to transmission. This is because, the receiving hub may perform further encoding. Since, the coefficient vectors are chosen from a large random space, there is a high probability that they are linearly independent. As a result, two coded messages that are created from the same native messages, are still useful to the destination (decoding is explained later).

Note that, hubs do not decode the messages. The encoding and forwarding process described above continues at all intermediate hubs. If a hub holds multiple encoded messages, then these can be be further combined into a single message. For example, assume that a hub has received two encoded messages F_1 and F_2, which have been created as follows,

$$F_1 = a_{11}X_1 + a_{12}X_2 + a_{13}X_3 \tag{3.2}$$

$$F_2 = a_{21}X_1 + a_{22}X_2 + a_{23}X_4 \tag{3.3}$$

Then the hub can combine these two messages to create a single encoded message, F_3, such that, $F_3 = a_1F_1 + a_2F_2$ where a_1 and a_2 are two randomly selected coefficients.

The above discussion has focused on the coding process. We now explain the decision making process involved before a hub encodes messages. Each hub maintains a coefficient matrix for all the encoded messages that it currently holds. There is one such matrix for each destination. The columns of the matrix correspond to the message IDs and there is one row for each encoded message. When two hubs encounter each other, they first exchange the coefficient matrices. These are generally included in the beacons, which are periodically exchanged by nodes. We explain the decision process for a single destination. These steps are repeated for each destination. When a hub receives its neighbor's matrix, it has to decide if transmitting a linear combination of all its messages, will be useful to the neighbor. The hub can determine this by checking if this encoded message is linearly independent to the encoded messages carried by the neighbor. Consider the following example. Let, F_1 be the encoded message created by this hub which is composed of two native messages X_1, X_2 and respective coefficients set A_1, $< a_1, a_2 >$ (i.e. $F_1 = a_1X_1 + a_2X_3$). Also assume that the hub receives coefficient matrix A_2 from its neighbor. A_1

and A_2 are shown below:

$$A_1 = \left| \begin{array}{cc} idx_1 & idx_3 \\ \hline a_1 & a_2 \end{array} \right|, \; A_2 = \left| \begin{array}{cc} idx_1 & idx_2 \\ \hline a_3 & a_4 \\ a_5 & a_6 \end{array} \right|$$

Since the coefficient matrix is accompanied by the message IDs (i.e., idx_i) of the corresponding columns of the matrix, the receiving hub can determine which column is associated with which message. The receiving hub then inserts the coefficient set A_1 in the corresponding columns of A_2. In this particular case, A_2 does not contain any coefficient for the native message X_3 (i.e. there is no column in A_2 for the message ID of X_3). So, a new column for message X_3 is created. The coefficients for the message X_3 in A_2 will be zero. Similarly the coefficient of the message X_2 in A_1 will also be zero. The modified A_2 is shown below:

$$A_2 = \left| \begin{array}{ccc} idx_1 & idx_2 & idx_3 \\ \hline a_3 & a_4 & 0 \\ a_5 & a_6 & 0 \\ a_1 & 0 & a_2 \end{array} \right|$$

If the coefficient sets (i.e. rows of the matrix A_2) are linearly independent then it is assumed that the newly encoded message (F_1) by the hub is useful to its neighbor.

Though this requires the hubs to exchange significant information, they can make more informed decisions about forwarding encoded messages and hence, avoid the transmissions of redundant messages. Eventually, when a hub meets the destination, it forwards an encoded message composed of all messages addressed to that destination.

Destination

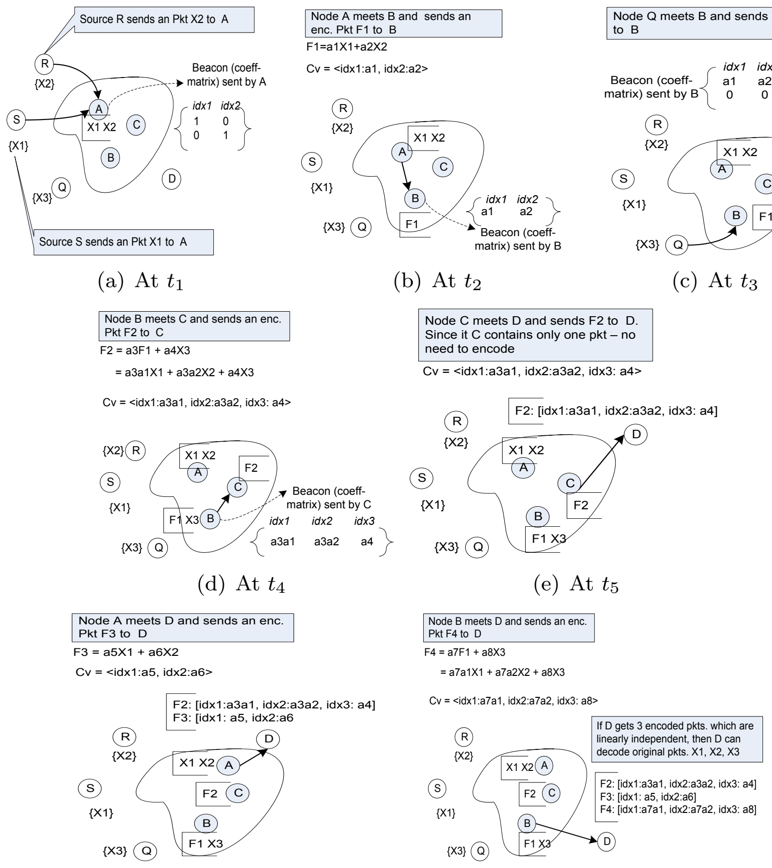
When the hub encounters a destination, it forwards an encoded message to it. Similar to the hubs, the destination also maintains a coefficient matrix. The columns represents the native message ids and each row corresponds to an encoded message. Recall, that each encoded message is a linear combination of the native messages. Consequently, to decode n messages, the destination should receive m linearly independent combinations of these messages, such that $m \geq n$. Note that, since the coefficients are randomly chosen from a large finite space, there is a high probability that all encoded messages are linearly independent. Hence, n encoded messages are sufficient for decoding (i.e., $m = n$). The n linear equations can be solved using matrix inversion.

For example, if the destination receives the following linearly independent encoded messages, F_1, F_2 and F_3: $F_1 = a_{11}X_1 + a_{12}X_2 + a_{13}X_3$, $F_2 = a_{21}X_1 + a_{22}X_2 + a_{23}X_3$, $F_3 = a_{31}X_1 + a_{32}X_2 + a_{33}X_3$, Then the set of linear equations can be written in matrix form $f = Ax$.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \; x = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}, \; f = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

The native messages can be retrieved by matrix inversion:

$$x = A^{-1}f \tag{3.4}$$



(a) At t_1 (b) At t_2 (c) At t_3

(d) At t_4 (e) At t_5

(f) At t_6 (g) At t_7

Figure 3.1: Example illustrating the operation of HubCodeV1

Fig. 3.1 presents an illustrative example of HubCodeV1. There are three hubs, A, B and C. Q, R, S, and D are regular nodes. Let us assume that R, S and Q are source nodes which wish to transmit messages X_2, X_1 and X_3, respectively to a common destination D. The arrows in the figure indicate that the two nodes can communicate with each other. For example, at t_1 both R and S are in the communication range of A. The direction of the arrow indicates the flow of data messages. C_v is the coefficient vector that is appended to the encoded message. It takes the form $\langle idx_i : a_i \rangle$, where idx_i represents the message ID and a_i denotes the coefficient. The figure is self-explanatory and shows the sequence of operations that are involved in delivering the messages to the destination, D.

3.2 HubCodeV2

The main drawback of HubCodeV1 is that the hubs need to exchange their coefficient matrices in order to make the forwarding decision. The overhead of this exchange is $O(n^2)$ for n messages. This overhead is particularly of concern when the contact durations with other hubs are short-lived. This is because, in such instances the exchange of auxiliary information may dominate the entire contact opportunity. Empirical measurements have shown that in real-world DTN [18], contact durations can often be quite short. To solve this problem, we present an alternate approach which seeks to reduce this overhead without penalizing message delivery.

In V1, a hub uses the coefficient matrices received from a neighbor to determine if forwarding an encoded message is beneficial to this neighbor. However, if a hub can decode the coded messages to recover the native messages, then it can simply send a list of native message IDs to its neighbors instead of the coefficient matrix. As a result the neighboring hub can make the same decision. Sending a list of message IDs reduces the auxiliary data overhead to $O(n)$ for n messages, as compared to $O(n^2)$ with V1. However, this gain comes at the expense of extra computation. Since the hubs now decode messages, the computational complexity increase to $O(n^2)$ (solving n linear equations has a complexity of $O(n^2)$). On the other hand, in V1, the hubs only encode messages, which incurs a complexity of $O(n)$. Most personal communication devices (such as smart phones, PDAs) and in-vehicle routers have sufficient processing capabilities and battery power to perform the decoding operations. Hence, this scheme can be readily deployed in most people-centric DTN. However, V2 is not suitable for resource constrained devices such as sensor nodes. The two versions address an important trade-off between computational complexity and routing overhead.

As in V1, we classify nodes in three different groups: (1) Source, (2) Hubs, and (3) Destination and explain the operations performed by each type of node.

Source

As in V1, the source creates a copy of the native message (without encoding) and forwards it to a hub. However, unlike V1, in this scheme, the hubs may posses native messages (since they decode messages). As a result, a source forwards the native message to a hub only if the latter does not have this message. The source can determine this by examining the auxiliary information (i.e., message IDs) transmitted by the hub in the beacons.

Hubs

When a hub receives an encoded message for a destination, it examines the other encoded messages in its queue heading to the same destination. If sufficient encoded messages are present, then the hub decodes these messages (decoding was explained in V1) and stores the native messages. In the event, that sufficient messages have not been received, the encoded messages are stored as-is.

When two hubs encounter each other, they exchange the message IDs of the native messages that they carry. If a hub contains an encoded message, which has not been decoded yet, then the coefficients of this message are not included in the auxiliary information. In other words, only the information of the native messages is exchanged. When a hub receives the native message list of its neighboring hub, it compares this list against the native messages waiting in its queue and also against the messages which are used to compose the encoded messages (if any). If the hub finds at least one message (either native or a part of encoded message) that is not in its neighbor's list, then the hub encodes the missing messages along with any other messages (native or encoded) for that destination and forwards the combinations to that neighbor.

When the hub meets the destination and if it only has encoded messages for that destination (i.e., no native messages) then it sends an encoded combination of these messages. If the hub has one or more native messages in its queue then it simply forwards them to the destination without coding.

Destination

The decoding operation at the destination is exactly similar as in HubCodeV1. Hence, we do not provide details here. The only difference is that, unlike V1, the destination may receive native messages in addition to coded messages. Fig.3.2 highlights the basic operation of HubCodeV2. We have used the same scenario as in the example for HubCodeV1.

4 Mathematical Analysis

In this section, we present a mathematical model to estimate the message delivery delay in our hub-based forwarding schemes. In particular, we derive closed from expressions for message delivery delay in hub-based forwarding schemes when: a) hubs utilize network coding to disseminate a message among themselves (e.g. HubCode V1, V2), and b) hubs do not perform network coding and merely send copies of the message to other hubs (we refer to this as *Hub-only* scheme). We include the latter scheme in our analysis to quantify the improvements achieved by using network coding.

We use the widely used network model introduced in [14] as the starting point for our analysis. In this model, the characteristics of ad hoc networks are captured through two parameters of the network: a) the number of nodes in the network and b) the intensity of identical and independent Poission processes which model the *meeting instances* between any pair of nodes. Though, originally the network model was proposed for ad hoc networks, it fits well in the context of DTN. This is because of the fact that the meeting instances of the nodes (or inter-meeting durations) determine the message forwarding rate in DTN.

In order to determine the message delivery delay, we need to estimate the average message forwarding rate. In addition to the inter-meeting durations, the message forwarding rate also depends on various forwarding properties, such as: nature of the forwarding protocol, importance of the messages, etc. In the following sections, we start our discussion by introducing the generic routing model. Following this, we begin our mathematical analysis by introducing the factors which affect the message forwarding rate. Once we model the message forwarding rate, we can formulate the expected message delivery delay in a straightforward manner.

4.1 Stochastic Forwarding Model

We assume a network which consist of one source, one destination and $N-1$ relay nodes (i.e. hubs). We also assume that non-hub nodes will not affect

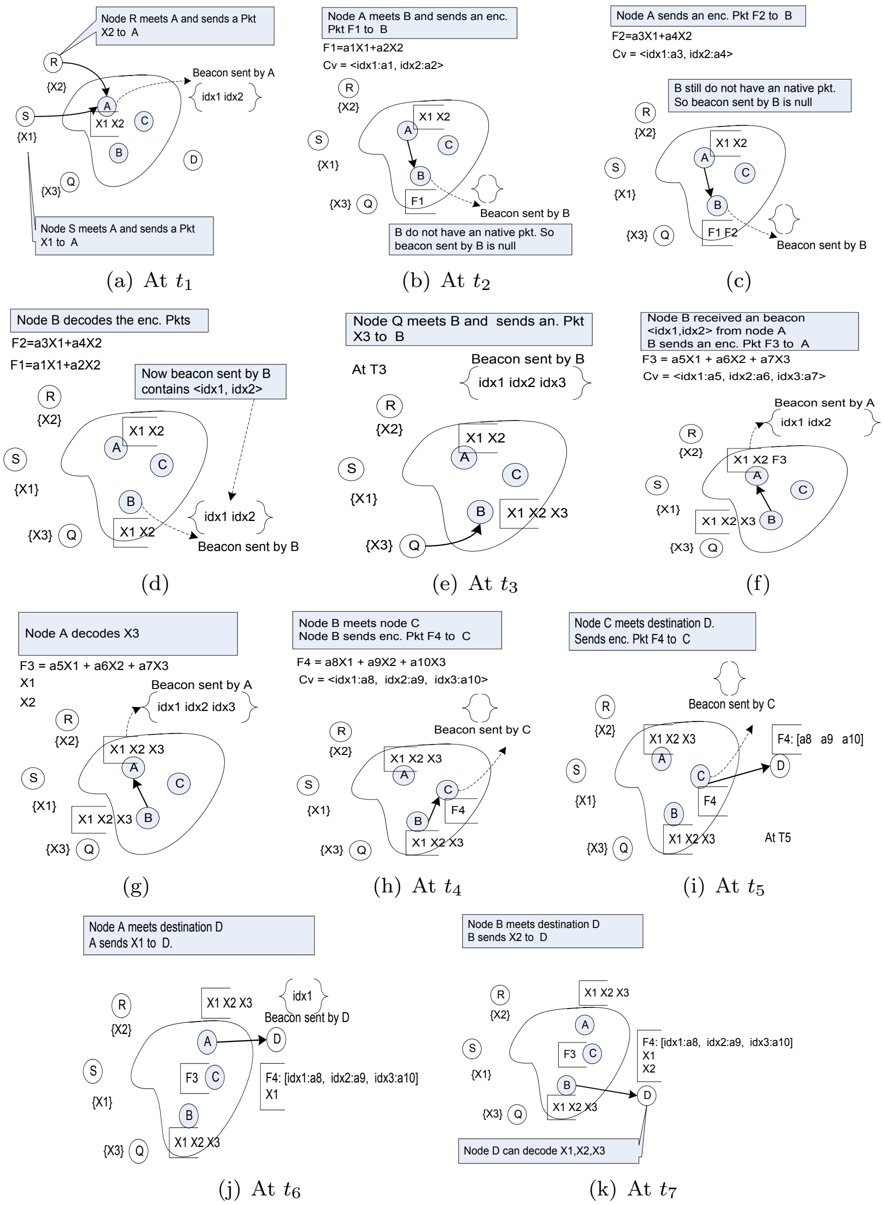


Figure 3.2: Example illustrating the operation of HubCodeV2

forwarding mechanism since message forwarding is performed only by the hubs. Two nodes may only communicate when they are within communication range. The duration when two nodes are connected is referred to as the *meeting time* and the time that elapses between two consecutive meeting times of a given pair of nodes is called the *inter-meeting time*. For simplicity, the transmission time is assumed to be instantaneous. This is a valid assumption if the the size of the message is very small. Obviously, if we had a realistic message traffic model available, then we could have taken a more pragmatic approach in modeling the transmission time. However, modeling message traffic is non-trivial because of its dependence on the network characteristics and network protocol behavior. We have also ignored the effect of processing delays.

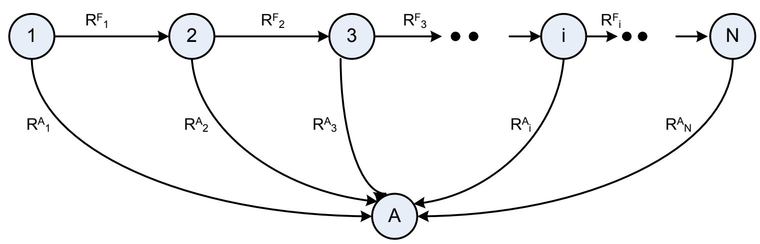


Figure 4.1: State diagram of Generic Routing Model

Fig. 4.1 shows the state diagram of the generic routing model. The system is in state $i \in 1, 2, \ldots, N$ when there are i copies of the message in the network. It is in state A when the message has been delivered to the destination. Let, R_i^F $(i \in 1, 2, \ldots, N-1)$ be the average message forwarding rate to a another hub and R_i^A $(i \in 1, 2, \ldots, N)$ be the average message forwarding rate to the destination when there are i copies of the message in the network. As can seen from Fig. 4.1, the mean time for a message to go from state 1 to sate A (i.e. from source to destination) directly (i.e. without the forwarding via hub) is: $\frac{1}{R_1^A}$. If the message takes one hop to reach the destination, then the mean delay becomes:

$$\frac{1}{R_1^F} + \frac{1}{R_2^A}$$

Similarly, if the message takes two hops to reach the destination, then the mean delay becomes:

$$\left[\frac{1}{R_1^F} + \frac{1}{R_2^F}\right] + \frac{1}{R_3^A}$$

Generalizing, the mean time for a message to go from state 1 to state A in k-th $(k \in 1, 2, 3, \ldots, N-1)$ hop is given by:

$$\sum_{i=1}^{k} \left[\frac{1}{R_i^F}\right] + \frac{1}{R_{k+1}^A} \tag{4.1}$$

In other words, the above expression provides the mean message delivery delay when the message takes k relays to reach the destination.

Using this simple reasoning, we develop our message delay model for both HubCode and Hub-only forwarding schemes in next sections.

As mentioned earlier the message forwarding rate depends not only on the inter-meeting durations between nodes but also on other factors, such as: forwarding principle, number of copies in the network, etc. Next, we discuss about the major factors which affect the message forwarding rates (R_i^F and R_i^A) and quantify the impact of these factors on the forwarding rate.

4.2 Factors that Influence the Message Forwarding Rate

- **Contact Rate Between Nodes:** Since nodes can only forward messages during contact periods, the contact rate between nodes directly influences

Table 4.1: Notations

Term	Definition
$E[T_{HO}]$	Expected message delivery delay of Hub-only (w/o coding) scheme
$E[T_{HC}]$	Expected message delivery delay of HubCode scheme
R	Average rate of contact opportunities
R_i^F	Average msg. fwd. rate to another hub when i copies of msg. are in network
R_i^A	Average msg. fwd. rate to destination from a hub or source when i copies of the msg. are in network
α	Power-law tail index
x_{min}	Min. inter-contact duration
p	Probability of msgs. relevance to the peer
f	Probability that a hub meets another hub
N	Number of hubs
M	Number of msgs. destined to common address

the message forwarding rate. Inter-meeting durations determine the contact rate. In a study on the association patterns of wireless access points [17], the authors found that the inter-meeting durations between nodes of a large network can be modeled using generalized Pareto distributions. Our study with VANET trace also shows similar findings. For example, in Fig. 4.6, we have plotted the PMF of inter-contact durations of the Seattle trace during 3pm-8pm period and a close-fit pareto distribution in order to show the goodness of the fit.

If the inter-meeting durations between a pair of nodes are random variables (X) with a power-law distribution (e.g. Pareto distribution), then the expected inter-meeting duration $E(X)$ (details can be found in Appendix) is:

$$E(X) = \frac{\alpha x_{min}}{\alpha - 1} \tag{4.2}$$

where x_{min} is the minimum value of X and α is a positive parameter often termed as the *tail index* of the power-law distribution. The value of α and x_{min} can be obtained empirically by using the method of maximum likelihood [5]. In fact, the value of α (i.e. the tail index) can be approximated graphically by plotting the data sample on a log-log plot and taking the slope of close-fit straight line of the data points.

The average contact rate R is inversely proportional to the average inter-contact duration $E(X)$. Therefore, R is:

$$R = \frac{\alpha - 1}{\alpha x_{min}} \tag{4.3}$$

- **Number of Copies of Messages:** At the beginning, when the number of copies of a message is low, the message forwarding rate increases exponentially. This exponential growth can be explained intuitively by considering the fact that at the beginning, the source copies the message to another node, then these two nodes can copy the message to another two nodes, thus, making the total copies equal to four. Now, these four nodes can copy the message to another four nodes, thus, making the total copies to eight, and so on. However, this rate declines as the copies saturates the network. We elaborate this effect of number of copies (of a message) in forwarding rate in the next section.

- **Forwarding Mechanism:** The mechanism of any particular routing protocol may change the message forwarding rate. For example, in our routing protocol (e.g. non-coding Hub-only version), a node forwards a message to only a fraction of the total nodes (i.e. highly connected nodes) and thereby reduces the message forwarding rate by a factor f. Intuitively, f is the probability that the next encountered neighbor is a hub.

- **Relevance of the Messages:** The message forwarding rate also depends on the relevance of the message to other nodes (i.e. a redundant message is not relevant to a node). Let, p be the probability that a message is important to the forwarded node. For network coding based schemes, $p \approx 1$, since almost all the messages are important (i.e. linearly independent) due to the application of random linear encoding scheme. In particular, it has been shown [9] that for network coding based schemes, $p = 1 - 1/d$ where d is the coefficients space. Intuitively, the linear independence among the messages depends on the size of the random coefficients. The larger the coefficients, the greater the probability that the messages are linearly independent. Since the finite field size d is often chosen as 2^{16}, p approaches to 1. On the other hand, when network coding is not employed, analogous to the *Coupon Collector's Problem* [28], p becomes $1 - \frac{j-1}{M}$, where M is the total number of unique messages destined to same address and j-th ($j <= M$) message has been received successfully.

$$p = \begin{cases} 1 & \text{for HubCode} \\ 1 - \frac{j-1}{M} & \text{for Hub-only (w/o coding)} \end{cases} \tag{4.4}$$

In the following two sections, we accommodate the above mentioned factors in the average message forwarding rates (R_i^F and R_i^A) and estimate the message delivery delay for both HubOnly and HubCode schemes.

4.3 Message Delivery Delay in Hub-only Scheme

In this sub-section, we develop the message delay model of Hub-only forwarding scheme. A brief definition of the notations used here can be found in Table 4.1. Fig. 4.2 shows the state diagram of the Hub-only (w/o coding) forwarding scheme. Recall that in Hub-only forwarding scheme, the source forwards the message to hubs and hubs disseminate it among themselves in Epidemic manner. Eventually one of the copies reaches the destination.

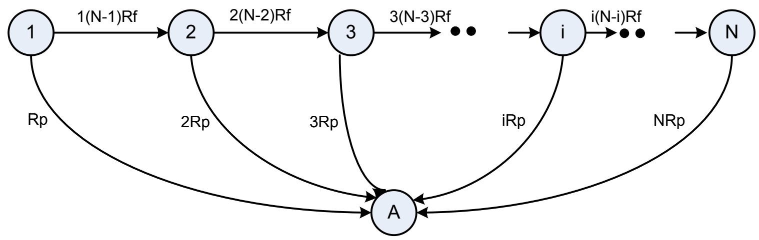


Figure 4.2: State diagram of Hub-only forwarding scheme

The message forwarding rate R_i^F at each state i is a function of the number of copies i of the message in the network. We term the nodes as *infected* which are carrying the copies of the message. When there are i $(1 < i \leq N)$ copies of the same message in the network, then each infected node either sends a new copy to the $N - i$ hubs which do not have a copy yet at a rate of $R_i^F = i(N-i)Rf$, or meets the destination at a rate of $R_i^A = iRp$. The rationale behind the introduction of the factors p and f are discussed in the previous subsection.

Therefore, the mean time for the j-th $(j \in 1, 2, 3, \ldots, M)$ message to go from state 1 to state A (i.e. from source to destination) in k-th $(k \in 1, 2, 3, \ldots, N-1)$ step (Fig. 4.2) is:

$$\sum_{i=1}^{k} \left[\frac{1}{iR(N-i)f} \right] + \frac{1}{(k+1)Rp}$$

Since, in real-world people-centric DTN, most contact durations are very short [41, 37], it can be assumed that only one message can be transferred at each contact opportunity. Therefore, the mean time for all M messages to go from state 1 to state A in k-th step is given by:

$$M \sum_{i=1}^{k} \frac{\alpha x_{min}}{i(\alpha-1)(N-i)f} + \sum_{j=1}^{M} \frac{\alpha x_{min}}{(k+1)(\alpha-1)\left(1 - \frac{j-1}{M}\right)}$$

[after replacing the values of R and p (from Eq. 4.3,4.4)].

So, the expected message delivery delay $E[T_{HO}]$ for all M messages to reach state A from state 1 is:

$$\begin{aligned} E[T_{HO}] = {} & \frac{1}{N-1} \sum_{k=1}^{N-1} \left[M \sum_{i=1}^{k} \frac{\alpha x_{min}}{i(\alpha-1)(N-i)f} \right] \\ & + \frac{1}{N-1} \sum_{k=1}^{N-1} \sum_{j=1}^{M} \frac{\alpha x_{min}}{(k+1)(\alpha-1)\left(1 - \frac{j-1}{M}\right)} \end{aligned} \tag{4.5}$$

A closed formed expression (Eq. 4.6) of the expected message delivery delay is obtained from Eq. 4.5 by employing calculus approximation methods (discussed in Appendix).

$$\begin{aligned} E[T_{HO}] = \frac{M\alpha x_{min}}{Nf(N-1)(\alpha-1)} \Bigg[& (N-2)ln(N-1) + \\ & fNln\left(M\right)ln\left(\frac{N}{2}\right) \Bigg] \end{aligned} \tag{4.6}$$

4.4 Message Delivery Delay in HubCode

In this sub-section, we sketch the message delay model of our HubCode forwarding schemes. Recall that the HubCodeV1 and HubCodeV2 only differ in how they broadcast auxiliary data in beacons. For example, HubCodeV1 exchanges coefficient matrix whereas HubCodeV2 exchanges native message ids in beacons. In HubCodeV2, in order to meet the purpose of exchanging the auxiliary data (i.e. message ids), the hubs try to decode the messages. However, the basic principle of forwarding messages to hubs which in turn disseminate messages amongst other hubs using network coding, is the same for both schemes. Since our model does not incorporate the beaconing mechanism, the same forwarding model is applicable to both HubCodeV1 and HubCodeV2. We use the generic term *HubCode* to refer to both HubCodeV1 and HubCodeV2.

Fig. 4.3 shows the state diagram of the HubCode forwarding scheme. Note that in HubCode, the source forwards the message to hubs and hubs encode and disseminate the encoded message among themselves. The destination receives multiple encoded messages from different hubs. Unlike the Hub-only (non-coding) scheme, practically all the encoded messages are important to the destination due to the application of linear encoding method. The destination decodes the original messages by solving the set of linear equations (i.e. encoded messages together with coefficient vector).

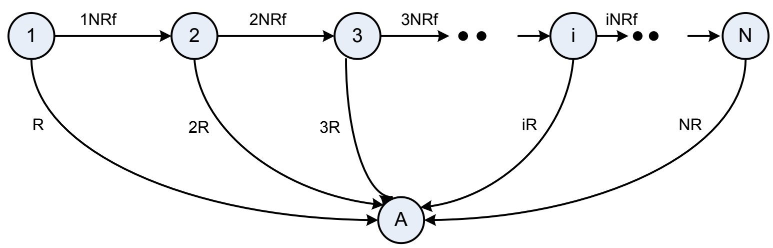


Figure 4.3: State diagram of HubCode forwarding scheme

When there are i $(1 < i \leq N)$ copies of the same message in the network, then unlike Hub-only (non-coding) forwarding scheme, each of those nodes either sends a newly encoded message to the N hubs (since all messages are important to *all* hubs) at a rate of $R_i^F = iNRf$, or meets the destination at a rate of $R_i^A = iRp$ or iR (since $p \approx 1$ in case of HubCode (Eq. 4.4)).

Now, the mean time for a message to go from state 1 to state A in k-th $(k \in 1, 2, 3, \ldots, N-1)$ step (Fig. 4.3) is:

$$\sum_{i=1}^{k} \left[\frac{1}{iRNf} \right] + \frac{1}{(k+1)R}$$

Therefore, the mean time for all M messages to go from state 1 to state A in k-th step is (assuming that only one message can be transferred at each contact opportunity):

$$M \sum_{i=1}^{k} \left[\frac{\alpha x_{min}}{i(\alpha-1)Nf} \right] + \frac{M\alpha x_{min}}{(k+1)(\alpha-1)}$$

after replacing the value of R (from Eq. 4.3).

So, the expected message delivery delay $E[T_{HC}]$ for all M messages to reach state A from state 1 is:

$$E[T_{HC}] = \frac{1}{N-1}\sum_{k=1}^{N-1}\left[M\sum_{i=1}^{k}\frac{\alpha x_{min}}{i(\alpha-1)Nf}\right] + \frac{1}{N-1}\sum_{k=1}^{N-1}\frac{M\alpha x_{min}}{(k+1)(\alpha-1)} \tag{4.7}$$

A closed formed expression (Eq. 4.8) of the expected message delivery delay is obtained from Eq. 4.7 by using calculus approximation methods (discussed in Appendix).

$$E[T_{HC}] = \frac{M\alpha x_{min}}{Nf(N-1)(\alpha-1)}\left[(N-1)ln(N-1) + fNln\left(\frac{N}{2}\right) - N + 2\right] \tag{4.8}$$

4.5 Model Validation

In order to validate our theoretical model, we compare it with the simulation results. The details of the mobility trace and other simulation parameters have been discussed thoroughly in Section 5.

In the first set of our simulations, 10 randomly selected sources (from a pool of 1065 nodes) sent 10 messages each to a single common destination (randomly selected). That is, a total of 100 messages are heading towards a single common destination.The message payload is 1000 bytes. The source nodes generate messages between the time period 3000 to 4000 seconds after the start of the simulation. The average delivery delay is measured for only the messages which reach the destination. The messages which are not received during the the 5 hour lifetime of the simulation period are ignored. We have found that approximately 63% of the messages reach the destination in average. The number of hubs are varied from 5 to 150 in the following steps: $\langle 5, 25, 50, 75, 100, 125, 150\rangle$. The simulation is repeated 20 times for each instance. We measure the message delivery delay and plot the average value in Fig. 4.4. We also plot the 95% confidence intervals. We compare the simulations results to the message delay derived in Eq. 4.8 and Eq. 4.6. The parameters, $\langle \alpha, x_{min}, f\rangle$ in Eq. 4.8 and Eq. 4.6 are derived empirically from the mobility traces as, $\langle 1.01, 30, 0.8\rangle$ respectively. To observe the impact of increasing the traffic, we increase the number of messages sent by each random source from 10 to 20 and repeat the above simulations. The corresponding results are plotted in Fig. 4.5. The theoretical curve of Hub-only scheme (w/o coding) is also plotted in order to compare its delivery delay with that of HubCode. Recall that in the Hub-only scheme, unlike HubCode, the hubs do not encode multiple messages into a single message.

One can observe from Fig. 4.4 and 4.5 that there is significant disparity between the simulation and analytical results when the number of hub nodes are small (< 50). However, as the number of hub nodes increases (> 50), the

]h]

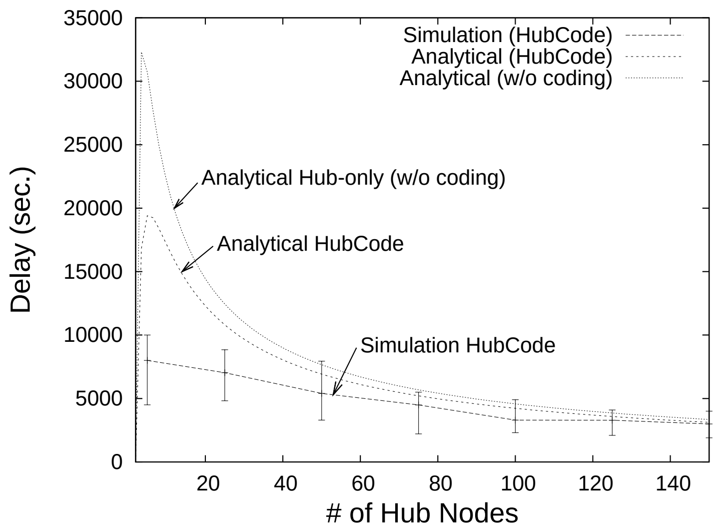


Figure 4.4: Model validation: (10 message sources)

two results converge. Further, both results exhibit the same general trend, in that the delivery delay decreases as the number of hubs increase.

In the following, we discuss possible causes of the disparity between the theoretical model and the simulation results when the number of hubs are small (< 50).

- *Theoretical Limitation:* First of all, the approximate closed form mathematical models impose some theoretical limits on possible values of hubs. For example, a cursory look at the closed form equations Eq. 4.8 and Eq. 4.6 reveals that the equations are undefined when $N < 3$ (i.e. $hubs < 3$). This explains the peculiar shape of the theoretical model at low values of N.

- *Biased Expectation of Inter-contact Rates:* The accuracy of the expected inter-contact rate directly affects the message delivery delay in Eq. 4.8 and Eq. 4.6. In an empirical study, we have found that the inter-contact durations follows power-law distribution and modeled it with a simple *Pareto* distribution. However, due to the nature of power-law distribution (i.e. most inter-contact durations are short-lived but there exist few very long inter-contact durations), the expected inter-contact duration exhibits bias towards fewer very long inter-contact durations. For example, Fig. 4.6 plots a theoretical Pareto curve that fits the empirical inter-contact distribution of the mobility trace (from 3pm-8pm slot) that we have used in our simulation. The expected value of the theoretical Pareto distribution is found to be ≈ 11 minutes. However, from the Fig. 4.6, it can be seen that the probability of occurrence of this expected value is only ≈ 0.008. Hence, the expected value of inter-contact durations does not accurately represent the rate of contact opportunities. In other words, it contributes towards a more pessimistic message delay.

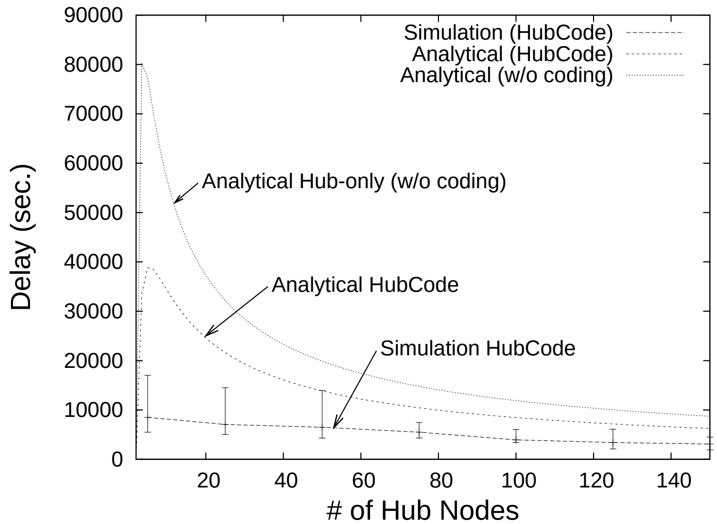


Figure 4.5: Model validation: (20 messages sources)

- *Multiple Transfer per Contact Opportunity:* Recall that in our model, we assumed that only a single message can be transferred at a contact duration. Though most of the contact durations are very small (e.g. < 10 seconds),few contact durations are indeed large enough to transfer multiple messages (this is a typical behavior of power-law based distributions). As a direct consequence of the fact, the theoretical model tends to predict larger message delivery delay than the results obtained from simulation settings; especially when the number of messages destined to common address is increased (Fig. 4.5).

- *Problem of Averaging with Fewer Terms:* Our model exhibits instability when the number of hubs (acting as message relays) is very small (between 1 - 6). If we delve into the the message delay model in Eq. 4.1, Eq. 4.7 and Eq. 4.5, we find that averaging with fewer terms (i.e. when number of hubs is very small) is responsible for the instability. We explain this by considering the generalized delay model in Eq. 4.1, which is restated below for convenience:

$$\text{Mean delay for arriving at state } A \text{ in } k \text{ steps} =$$
$$\left[\underbrace{\left(\frac{1}{R_1^A} + \frac{1}{R_2^A} + \ldots + \frac{1}{R_{k-1}^A} \right)}_{\text{1st part}} + \underbrace{\left(\frac{1}{R_k^F} \right)}_{\text{2nd part}} \right]$$

In the above equation, the first part represents the cumulative forwarding delay among hubs and the second part represents the delay when a relay delivers the message to the destination. The values of the trailing terms in first part rapidly decrease as the value of k (i.e. hubs) increases due to the rapid growth rate of R_i^A. As a consequence, the average value calculated (in Eq. 4.7 and Eq. 4.5) with fewer terms becomes significantly larger than

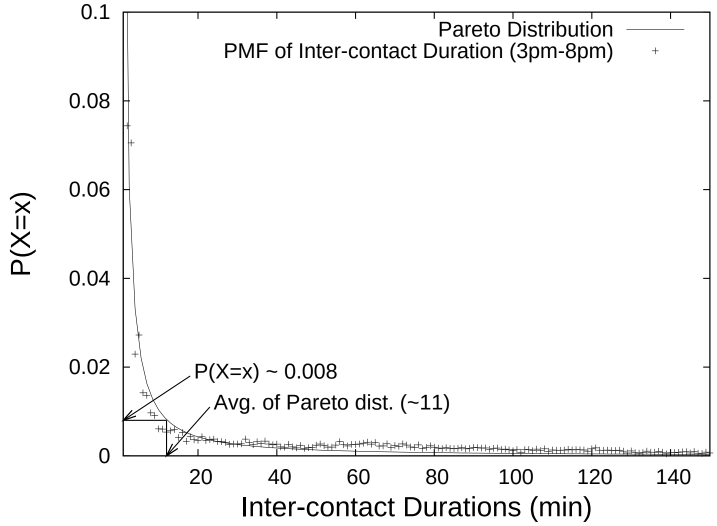


Figure 4.6: Biased Expected Value of Power-law Distribution

that of more terms since first few terms are much larger than the rest of the terms. For example, the average of $\left[\frac{1}{R_1^A} + \frac{1}{R_2^A}\right]$ is greater than that of $\left[\frac{1}{R_1^A} + \frac{1}{R_2^A} + \frac{1}{R_3^A}\right]$. This explains the initial large discrepancy between the theoretical and simulation results at fewer number of hubs.

As mentioned earlier the theoretical curve of Hub-only scheme (w/o coding) is included in order to compare its delivery delay with that of HubCode. As expected, the theoretical message delivery delay (Fig. 4.4 and 4.5) in Hub-only scheme is greater than that of HubCode. The difference becomes even more prominent when the number of messages to common destination are increased (Fig. 4.5). Since more messages are headed towards common destination (Fig. 4.5), the coding opportunity of HubCode increases, hence showing improved performance than Hub-only scheme. In both Hub-only and HubCode schemes, the theoretical models provide us with upper bounds on the delivery delay.

5 Simulation based Evaluations

In this section, we present simulation-based evaluations that compare the performance of the proposed HubCode schemes with other DTN forwarding schemes. We use mobility traces of a large-scale vehicular DTN network. In all our simulations, we take a pragmatic approach, wherein, the data exchanged by two adjacent nodes is proportional to the contact duration. Further, we also account for the auxiliary information exchanged by the nodes. In the first set of experiments, we compare the performance of our forwarding schemes against others schemes in terms of message delivery ratio and delivery cost. We also evaluate the robustness of our schemes against random node failure. In the second set of simulations, we investigate the impact of various parameters on the forwarding performance. In particular, we study the impact of varying the % of nodes

Table 5.1: Properties of Mobility Traces

Data Set	**Intel[30]**	**Cambridge Comp Lab[30]**
Period	05-01-06 to 05-01-11	05-01-25 to 05-01-31
Device	iMote	iMote
Network	Bluetooth	Bluetooth
Nodes	9	12
Coverage	Top 2 covered 9	Top 2 covered 12

Table 5.2: Properties of Mobility Traces

Data Set	**INFOCOM'05[18]**	**Seattle Buses[20]**
Period	05-03-07 to 05-03-10	01-10-30 to 01-11-26
Device	iMote	None
Network	Bluetooth	WiFi (simulated)
Nodes	41	1163
Coverage	Top 4 covered 41	Top 100 covered 865

classified as hubs, the message size and the traffic load.

5.1 Mobility trace details

In recent years, several researchers have conducted empirical measurements to study the behavior of people-centric DTNs. In these experiments, communicating devices (bluetooth, zigbee, etc) are either handed to a volunteer group [18] or are mounted on moving vehicles [4, 11]. The devices record all opportunistic contacts with other devices in the participant set and also with external devices. The external contacts are often excluded in the analysis, since complete information about their encounters is not available. Due to practical limitations of conducting empirical experiments, the node population in all the traces is quite small (see Table 5.1 and Table 5.2). Further, it has been observed that few hubs are connected to *all* other nodes (i.e. have full degree distribution). This is an artifact of the small population in the traces and is not representative of real-world networks. As a result, we have found that schemes, which exploit the power-law properties such as BubbleRap, achieve close to optimal performance (results are excluded for brevity). Hence, employing network coding at the hubs offers little advantage.

Therefore, in our evaluations, we use mobility traces from a significantly larger network, which captures the movement of public transport buses from the King County Metro bus system in Seattle [20]. This transport network consists of 1163 buses plying over 236 distinct routes covering an area of 5100 square kilometers. The traces were collected over a three week period in Oct-Nov 2001. The traces are based on location update messages sent by each bus. Each bus logs its current location using an automated vehicle location system, its bus and route id along with a timestamp. The typical update frequency is 30 seconds. These traces have been primarily used in literature to study the performance of routing protocols in vehicular ad hoc networks [8]. The traces can be readily used to simulate a bus-based DTN, similar to DieselNet [41]. As in [41], we assume that each bus is equipped with a 802.11b radio. Buses exchange messages when they are within the communication range of each other, which is assumed to be 250m.

The traces were post-processed to generate fine-grained location information. The details have been omitted for reasons of brevity. The trace also shows power-law behavior. However, unlike the other traces, no single hub connects to most of the other nodes (the maximum degree of a hub is 0.15). This is representative of a large real-world people-centric DTN.

5.2 Simulation Settings

We use a custom discrete event simulator. We assume that each node broadcasts a beacon every 5 seconds, for neighbor discovery. The beacons also contain additional information as required by the routing scheme (e.g., HubCode). Since, we wish to study the performance of the routing schemes in isolation, the 802.11 MAC is not implemented. In each simulation we inject 1000 messages to 100 destinations (both source and destination are randomly selected from the pool of 1065 nodes). We assume that the message inter-arrival duration at the source is exponentially distributed, with the average inter-arrival duration set to 30 seconds. The message payload is 1000 bytes. The source nodes generate messages between the time period 3000 to 4000 seconds after the start of the simulation. This is because several nodes are inactive in the initial period. Each simulation lasts for 5 hours (18000 seconds). We choose the 3pm-8pm period from two successive weekdays, 31 Oct 2001 (Wed) and 1 Nov 2001 (Thu). We simulate each trace 20 times for statistical significance. The results presented are averaged over the 20 runs. The 95th percentile confidence intervals are all within 10% of the average.

To evaluate the performance, we use the following metrics: (i) delivery ratio, which is the ratio of the messages delivered to the messages created, and (ii) delivery cost, which is measured as the total number of messages transmitted, normalized by the total number of unique messages created. Note that, the delivery cost does not include the auxiliary messages exchanged by the nodes in the beacons.

We compare the performance of the HubCode schemes (we use the term HubCode to refer both HubCodeV1 and HubCodeV2) with several other DTN forwarding schemes. *Epidemic* (i.e. flooding) [36] is included since it achieves the highest delivery ratio when the network is not congested. We choose *Spray and Wait* [35], as a representative restrictive multiple-copy forwarding scheme. In this scheme, the source initially makes n copies of a message and forwards half of those to a neighbor it meets. This node in turn repeats this strategy, i.e. it forwards half of the messages to the next encountered node and retains the other half. This process repeats until a node is left with one copy, which is only forwarded to the destination. In our simulation, we choose initial number of copies n to 8, which is a fair compromise between the delivery ratio and the cost [35]. *RLC* [42] and *BubbleRap* [19] are included since they are closely related to our work (see Section II). We use the same methods as described in [19] to rank the nodes.

To implement HubCode, it is necessary to identify the hub nodes in the network. For this, we analyze the traces from an entire weekday and rank the nodes according to the number of unique nodes they have encountered. We find that the ranking is similar for all the weekdays, because the buses follow repeatable patterns. Most people-centric networks are known to exhibit such repeatable behavior. The top 10% of nodes (i.e., 116 buses) are classified as the hubs.

Contact durations are finite and short lived in real world scenario. For example, analyzing the bus traces we found that several contact durations are smaller than 30 sec. Similar behavior is observed in other real-world networks [18, 11]. Hence, we assume that the amount of data exchanged is proportional to the length of the contact duration. For simplicity, we assume the following linear relationship. The amount of data, D, exchanged during a contact duration T_c seconds is given by,

$$D = (T_c - T_a) \times 4Mbps \tag{5.1}$$

Empirical experiments have shown that in 802.11b, the typical goodput (accounting for overheads) at the highest data rate of 11Mbps is around 4Mbps [21]. T_a refers to the association time, which includes typical time to associate with an access point. For simplicity, we assume a fixed value of 10 seconds for T_a. We also account for the time required to exchange the beacon messages (note that, the size of the beacons vary depending on the forwarding scheme employed).

5.3 Delivery Ratio and Delivery Cost

Figs. 5.1 and 5.2 plot the delivery ratio and delivery cost, respectively for all the forwarding schemes. The delivery ratio for HubCodeV2 is approximately 72% which outperforms all other schemes by about $15\% - 20\%$. On the contrary the delivery ratio for HubCodeV1 is about 60%. Recall that, in HubCodeV2, the hubs do not exchange the complete coefficient matrices as in HubCodeV1. Rather, they only exchange native message ids. Hence, in HubCodeV1, particularly when the contact opportunities are short, significant time is utilized in exchanging the coefficient matrices, thus leading to several wasted opportunities for transferring data. Fig. 5.2 shows that V1 still outperforms V2 slightly (by 20%) in terms of the delivery cost, as a consequence of the extra information exchanged in the beacons. Also observe that, the HubCode schemes are far superior in comparison with all other schemes (e.g., Epidemic incurs 300% excess costs as compared to V1). This is because our schemes restrict the message forwarding within the hubs and encode multiple messages together.

The delivery ratio of Epidemic is about 60%, which is less than that of HubCodeV1. This is because a node may not be able to transfer all the messages it carries to other nodes during an encounter. When contact durations are bounded, Epidemic, essentially resembles a restricted flooding scheme such as Spray and Wait, as evident from their similar delivery ratio. The delivery cost of Epidemic is still significantly high as compared to other schemes. Spray and Wait, reduces the cost by about 15% due to the cap on the number of copies exchanged between nodes. Despite employing network coding, the performance of RLC is poor in comparison with HubCode. This is because, in RLC, encoded messages are flooded in the entire network. Hence, the overhead of unnecessary message replications and dominating auxiliary data exchange exhaust scarce bandwidth.

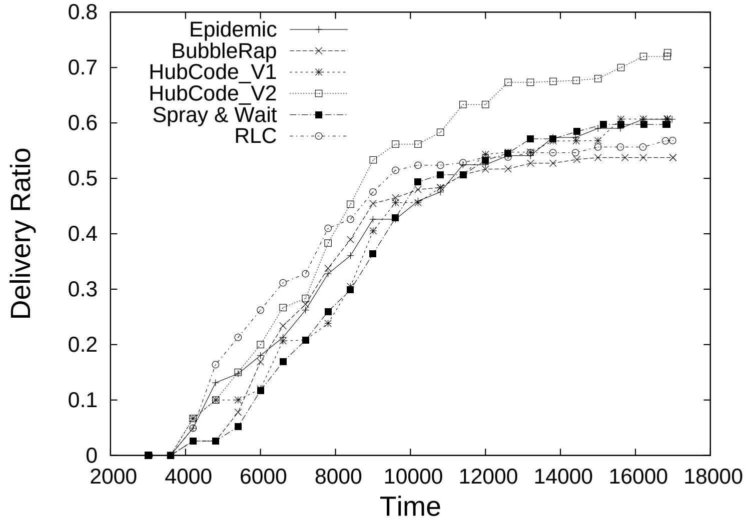


Figure 5.1: Delivery Ratio

5.4 Effect of Node Failure

In people-centric networks, node failures are a reality. A node failure may occur due to software/hardware failures or energy depletion. Further, a node may also cease to participate in message forwarding activity. In this section, we investigate the impact of node failures on the forwarding schemes under consideration. A randomly chosen fraction of nodes is made inactive over the entire duration of the simulation. We observe the impact of varying the percentage of inactive nodes on the delivery ratio. This allows us to investigate the robustness of our scheme to node failures. As before, the simulation is repeated 20 times and the average delivery ratio is plotted. Note that, we consider two different cases. In the first instance (Fig. 5.4), we assume that the inactive node are chosen from the entire set of nodes. In the second case, we only pick hubs as the inactive nodes (Fig. 5.3).

As can be seen from the graph (Fig. 5.3), that even a 20% failed hub nodes has only a minor effect on the delivery ratio (a 7% decline). Recall that the hubs are usually well connected (i.e. have high out degrees) and that HubCode disseminates messages among hubs which collectively form a data conduit. The high inter-connectivity among the hubs provides alternate paths to disseminate messages among themselves. As a result, our schemes are robust to the failure of a few hubs. Fig. 5.3 also shows that other schemes such as Epidemic and RLC are also not significantly affected by node failures. This is because in both of these schemes, messages are copied to all encountered nodes. Therefore, failure of some random nodes does not impact the message delivery ratio.

If the number of failed nodes is drawn from the entire node population (i.e. not merely from the group of hubs), the probability that the failed nodes contains hubs decreases. Since, HubCode only uses hubs (a fraction of total nodes) as delivery messengers, failure of non-hub nodes does not affect its delivery performance. Fig. 5.4 shows the effect of random hub failure on the delivery ratio of HubCodeV2. The same trend is also observed in HubCodeV1 (not shown in

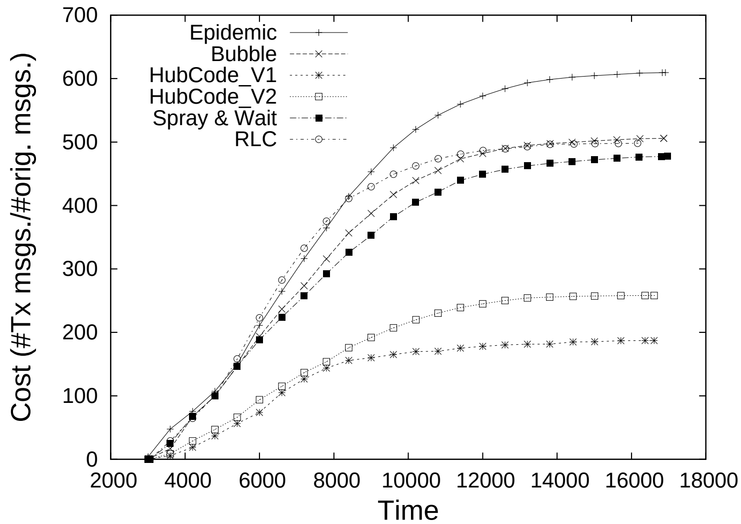


Figure 5.2: Delivery Cost

Figure).

5.5 Effect of the number of hubs

Recall that, HubCode uses the hubs as the message relays. In the previous experiments, we assumed that the top 10% of the nodes, ranked according to the degree distribution, are classified as the hubs. A natural question arises: what is the impact of increasing the number of hubs on the performance? In this set of experiments, we seek to answer this question. We consider the same parameters as in Section 5.3. Fig. 5.5 illustrates the impact of increasing the percentage of nodes that constitute the hubs on the delivery ratio. The delivery ratio for both schemes initially increase as we increase the number of hubs. This is expected since, the number of nodes that the hubs can collectively contact increases. As a result, the number of messages that can reach the destinations increase. However, after a certain knee point (around $20\% - 30\%$), the delivery ratio begins to decrease. This is because an increase in the number of hubs, reduces the number of messages carried by each hub. As a result, the opportunities for coding of multiple messages decrease. Also, the increased overhead caused by the increasing number of hubs is also a key factor which reduces the delivery rate. Since the overhead of HubCodeV2 is less than that of HubCodeV1, the knee point in HubCodeV2 is slightly higher than that of HubCodeV1 (30% as compared to 20% in HubCodeV1). Note that, when 100% of the nodes are regarded as the hubs, HubCode degenerates into RLC. The result concerning delivery costs is not shown in this paper since the findings are obvious: the costs continues to increase with increasing number of hubs. These results suggest that only a small percentage of the nodes ($20\% - 30\%$), which are highly connected should be designated as hubs.

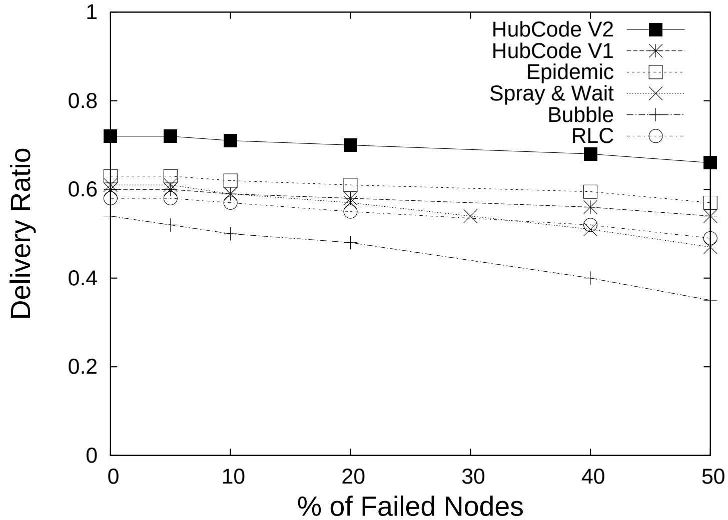


Figure 5.3: The effect of random node failure on message delivery

5.6 Effect of Message Size

In this section, we investigate the impact of the size of the message blocks on the delivery ratio. When message block size is small, the auxiliary headers make up for a significant percentage of the packet size, which reduces the forwarding efficiency. On the other hand, if the message block is too large, a node may fail to forward it if the contact duration between the nodes is small. In this experiment we seek to find the optimum block size when transferring a long file.

We assume that each file is $100KB$ long and there are 1000 such files to send. The number of destinations is 100 which are selected randomly from our pool of 1065 nodes. We vary the message block size from 1 to 3000 (in 500 bytes step) and observe the delivery ratio. Other parameters remain the same as in previous simulations. The experiment is repeated 20 times and average value of the delivery ratio is plotted (Fig. 5.6).

In HubCode versions, the coding opportunities increase when block size is small because the probability of the number of blocks headed towards common destinations increases. However, this effect is diminished by the negative impact of the increased amount of overhead caused by larger coefficient vectors and headers. As a result, the HubCode versions show very poor performance when block size < 50 bytes.

The delivery ratio begins to increase when the message block size is beyond 50 bytes. However, after reaching to a cutoff point (~ 500 in this case), the delivery ratio declines in response to further increase in message block size. This is because, as the message size becomes larger, the contact opportunities required for successful message transfer becomes scarce (i.e. small contact opportunities may fail to transfer large blocks) and negatively affects the delivery ratio.

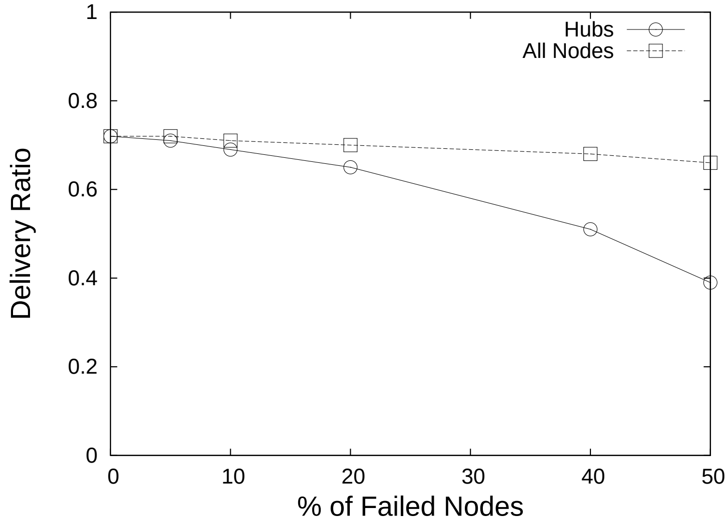


Figure 5.4: The effect of random hub failure on message delivery (HubCode V2)

5.7 Effect of traffic load

In this experiment, we gradually increase traffic load in order to observe its effect on the delivery performance of the routing protocols. As before, 100 destinations are randomly selected from the pool of 1065 nodes. We vary the number of injected messages from 500 to 3000 (incremented in blocks of 500). Messages are injected in the network 3000 seconds after the simulation begins at random intervals (using exponential random variable with $\lambda = 30$). The whole experiment is repeated 20 times and the average values are plotted (Fig. 5.7).

Since the number of destination remains the same (i.e. 100) in all cases, the increased traffic load implies that more messages are directed to common destinations. The delivery ratio initially increases steadily as traffic load is increased upto a knee point (1500 messages in this scenario) and then declines slowly when the traffic load is increased further. This is because, as the traffic load is increased, the probability that more messages have common destination increases, and so do the coding opportunities among the hubs. An increase in coding opportunity implies more efficient utilization of BW and hence increases the delivery ratio. However, if the traffic load is increased further beyond the knee point, the auxiliary data overhead (due to increased size of coefficients) dominates and exhausts scarce bandwidth resources (recall that most contact durations are very small). As a result, the delivery ratio begins to decrease when the traffic load is increased beyond the threshold.

6 Conclusion

In this paper, we proposed a novel forwarding strategy called HubCode for people-centric DTN that exhibit power-law behavior. HubCode uses the highly connected nodes as message relays. Further, messages are forwarded amongst the hubs using linear network coding. We presented two alternate implementa-

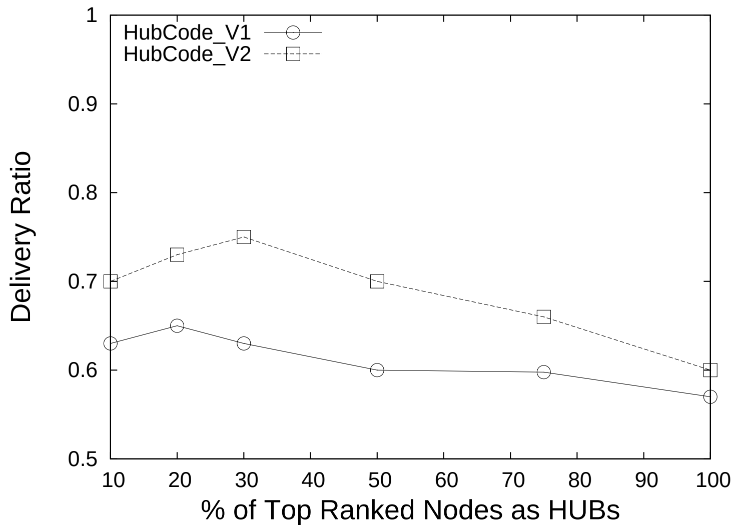


Figure 5.5: Delivery ratio as a function of the percentage of hubs

tions of HubCode to address the important trade-off between routing overhead and computational complexity. Our simulation-based evaluations of a large-scale vehicular DTN demonstrated the efficacy of our schemes. In particular, under pragmatic assumptions, our schemes were shown to achieve 20% higher delivery ratio and less than half of the delivery costs of comparable strategies.

We have derived closed-form expressions for message delivery delay of our proposed scheme and validated our model by comparing with simulation results. We show that our mathematical model serves as a lower-bound for the message delivery delay incurred under normal operational conditions.

A Expected inter-contact duration

A. Expected inter-contact duration

The probability density function of Pareto distribution is :

$$f(x; \alpha, x_m) = \frac{\alpha x_m^\alpha}{x^{\alpha+1}} \; for \; x \geq x_m$$

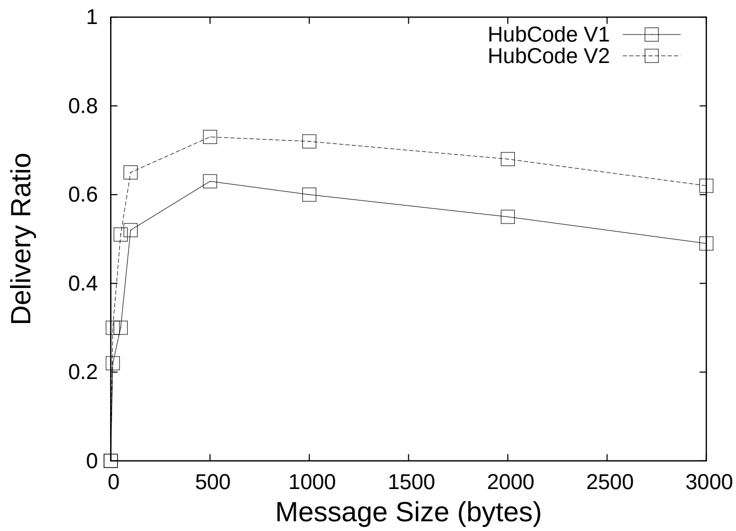


Figure 5.6: Delivery ratio as a function of message size

Therefore, the expected inter-contact duration $E(X)$ is:

$$
\begin{aligned}
E(X) &= \int_{x_m}^{\infty} x \frac{\alpha x_m^{\alpha}}{x^{\alpha+1}}\, dx \\
&= \alpha x_m^{\alpha} \int_{x_m}^{\infty} x^{-\alpha}\, dx \\
&= \alpha x_m^{\alpha} \cdot \left. \frac{x^{-\alpha+1}}{-\alpha+1} \right|_{x_m}^{\infty} \\
&= 0 - \alpha x_m^{\alpha} \cdot \frac{x_m^{-\alpha+1}}{-\alpha+1} \\
&= \frac{\alpha x_m}{\alpha - 1}
\end{aligned}
$$

B Expression for Coupon Collector's problem

B. Expression for Coupon Collector's problem

Here we show how coupon collector's problem affect the Hub-only (w/o coding) scheme. Let, M be the number of messages destined to common address. Also, let T_i be the time to collect $i-th$ message after $(i-1)$ messages have been collected.

In Hub-only forwarding scheme, when hubs replicate messages in epidemic manner, probability p_1 that the 1-st message is a new one is:

$$p_1 = \frac{M}{M} = 1$$

probability p_2 that the 2-nd message is a new one is:

$$p_2 = \frac{M-1}{M}$$

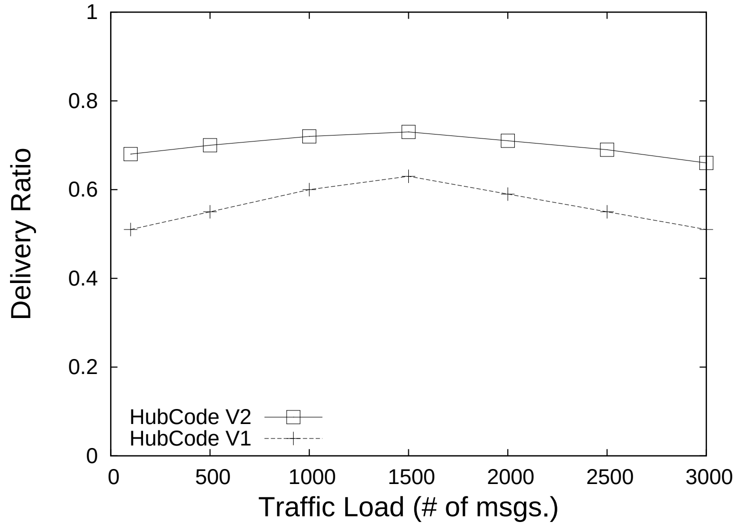


Figure 5.7: Delivery ratio as a function of traffic load

Similarly, the probability p_j that the j-th message is a new one is:

$$\begin{aligned} p_j &= \frac{M-(j-1)}{M} \\ &= 1 - \frac{j-1}{M} \end{aligned}$$

C Expected Message delivery delay in Hub-only forwarding scheme

C. Expected Message delivery delay in Hub-only forwarding scheme

The expression (Eq. 4.5) for Hub-only (w/o coding) forwarding scheme is restated below for convenience:

$$\begin{aligned} E[T_{HO}] = {} & \frac{1}{N-1}\sum_{k=1}^{N-1}\left[M\sum_{i=1}^{k}\frac{\alpha x_{min}}{i(N-i)(\alpha-1)f}\right] \\ & + \frac{1}{N-1}\sum_{k=1}^{N-1}\sum_{j=1}^{M}\frac{\alpha x_{min}}{(\alpha-1)(k+1)\left(1-\frac{j-1}{M}\right)} \end{aligned}$$

Let,

$$E[T_{HO}] = A + B \tag{C.1}$$

where,

$$A = \frac{1}{N-1}\sum_{k=1}^{N-1}\left[M\sum_{i=1}^{k}\frac{\alpha x_{min}}{i(N-i)(\alpha-1)f}\right] \tag{C.2}$$

and,

$$B = \frac{1}{N-1}\sum_{k=1}^{N-1}\sum_{j=1}^{M}\frac{\alpha x_{min}}{(\alpha-1)(k+1)\left(1-\frac{j-1}{M}\right)} \tag{C.3}$$

Using calculus approximations, we can re-write Eq. C.2 as:

$$
\begin{aligned}
A &= \frac{M}{(N-1)f} \int_1^{N-1} \int_1^k \left[\frac{\alpha x_{min}}{y(N-y)(\alpha-1)} dy \right] dk \\
&= \frac{M\alpha x_{min}}{(N-1)(\alpha-1)f} \int_1^{N-1} \int_1^k \left[\frac{1}{y(N-y)} dy \right] dk
\end{aligned}
$$

Re-writing the above equation:

$$
\begin{aligned}
A &= \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f} \int_1^{N-1} \int_1^k \left(\frac{1}{y} + \frac{1}{N-y} \right) dy\, dk \\
&= \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f} \int_1^{N-1} \Big[ln(y) - ln(N-y) \Big]_1^k dk \\
&= \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f} \int_1^{N-1} \Big[ln(k) - ln(N-k) \\
&\quad + ln(N-1) \Big] dk \\
&= \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f} \int_1^{N-1} \Big[ln(k) - ln(N-k) \Big] dk \\
&\quad + \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f} (N-2) ln(N-1)
\end{aligned}
$$

Using the relation: $\int ln(x)\, dx = x\, ln(x) - x$,

$$
\begin{aligned}
A = & \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f} \Big[k\, ln(k) - k \\
& + (N-k)\, ln(N-k) - (N-k) \Big]_1^{N-1} \\
& + \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f}\ (N-2) ln(N-1) \\
& = \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f} \Big[k\, ln(k) \\
& + (N-1)\, ln(N-k) - N \Big]_1^{N-1} \\
& + \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f}\ (N-2) ln(N-1) \\
& = \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f} \Big[(N-1)\, ln(N-1) - N \\
& - (N-1)\, ln(N-1) + N \Big] \\
& + \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f}\ (N-2) ln(N-1) \\
& = \frac{M\alpha x_{min}}{N(N-1)(\alpha-1)f}\ (N-2) ln(N-1)
\end{aligned}
$$

Similarly using calculus approximations, Eq. C.3 can be written as:

$$
\begin{aligned}
B &= \frac{\alpha x_{min}}{(N-1)(\alpha-1)} \int_1^{N-1} \int_1^{M} \frac{dy\, dk}{(k+1)\left(1 - \frac{y-1}{M}\right)} \\
&= \frac{\alpha x_{min}}{(N-1)(\alpha-1)} \int_1^{N-1} \frac{M}{k+1} \int_1^{M} \frac{dy}{M-y+1}\, dk \\
&= \frac{M\alpha x_{min}}{(N-1)(\alpha-1)} \int_1^{N-1} \frac{1}{k+1} \Big[-\, ln(M-y+1) \Big]_1^{M} dk \\
&= \frac{M\alpha x_{min}}{(N-1)(\alpha-1)} \int_1^{N-1} \frac{1}{k+1} \Big[ln(M) \Big]\, dk \\
&= \frac{M\, ln(M) \alpha x_{min}}{(N-1)(\alpha-1)} \int_1^{N-1} \frac{dk}{k+1} \\
&= \frac{M\, ln(M) \alpha x_{min}}{(N-1)(\alpha-1)} \Big[ln(k+1) \Big]_1^{N-1} \\
&= \frac{M\, ln(M) \alpha x_{min}}{(N-1)(\alpha-1)}\ ln\left(\frac{N}{2}\right)
\end{aligned}
$$

Replacing the values of A and B in Eq. C.1, we get the expected delivery delay $E[T_{HO}]$ for Hub-only (w/o coding) forwarding scheme:

$$E[T_{HO}] = \frac{M\alpha x_{min}}{Nf(N-1)(\alpha-1)}\Bigg[(N-2)ln(N-1) + Nf\,ln(M)\,ln\left(\frac{N}{2}\right)\Bigg]$$

D Expected Message delivery delay in HubCode

D. Expected Message delivery delay in HubCode

The expression (Eq. 4.7) for expected message delivery delay in HubCode is:

$$E[T_{HC}] = \frac{1}{N-1}\sum_{k=1}^{N-1}\left[M\sum_{i=1}^{k}\frac{\alpha x_{min}}{iN(\alpha-1)f}\right] + \frac{1}{N-1}\sum_{k=1}^{N-1}\frac{M\alpha x_{min}}{(\alpha-1)(k+1)}$$

Using similar calculus approximations, we can write:

$$\begin{aligned}
E[T_{HC}] &= \frac{M}{N-1}\int_{1}^{N-1}\int_{1}^{k}\frac{\alpha x_{min}}{yN(\alpha-1)f}\,dy\,dk \\
&+ \frac{M}{N-1}\int_{1}^{N-1}\frac{\alpha x_{min}}{(\alpha-1)(y+1)}\,dy \\
&= \frac{M}{N-1}\int_{1}^{N-1}\frac{\alpha x_{min}}{N(\alpha-1)f}\Big[ln(y)\Big]_{1}^{k}dk \\
&+ \frac{M\alpha x_{min}}{(N-1)(\alpha-1)}\Big[ln(y+1)\Big]_{1}^{N-1} \\
&= \frac{M}{N-1}\int_{1}^{N-1}\frac{\alpha x_{min}}{N(\alpha-1)f}ln(k)\,dk \\
&+ \frac{M\alpha x_{min}}{(N-1)(\alpha-1)}\left[ln(N)-ln(2)\right] \\
&= \frac{M\alpha x_{min}}{N(\alpha-1)(N-1)f}\int_{1}^{N-1}ln(k)\,dk \\
&+ \frac{M\alpha x_{min}}{(N-1)(\alpha-1)}ln\left(\frac{N}{2}\right)
\end{aligned}$$

Using the relation: $\int ln(x)\, dx = x\, ln(x) - x$,

$$
\begin{aligned}
E[T_{HC}] &= \frac{M\alpha x_{min}}{N(\alpha-1)(N-1)f}\Big[kln(k) - k\Big]_1^{N-1} \\
&+ \frac{M\alpha x_{min}}{(N-1)(\alpha-1)} ln\left(\frac{N}{2}\right) \\
&= \frac{M\alpha x_{min}}{N(\alpha-1)(N-1)f}\left[(N-1)ln(N-1) - N + 2\right] \\
&+ \frac{M\alpha x_{min}}{(N-1)(\alpha-1)} ln\left(\frac{N}{2}\right) \\
&= \frac{M\alpha x_{min}}{Nf(N-1)(\alpha-1)}\Big[(N-1)ln(N-1) \\
&+ Nfln\left(\frac{N}{2}\right) - N + 2\Big]
\end{aligned}
$$

Bibliography

[1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung. Network information flow. In *IEEE Transaction on Information Theory*, volume 46, pages 1204–1216, July 2000.

[2] S. Ahmed and S. S. Kanhere. Hubcode: Message forwarding using hub-based network coding in delay tolerant networks. In *The 12th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM 2009*, pages 288–296, October 2009.

[3] M. Balazinsky and P. Castro. Characterizing mobility and network usage in a corporate wireless local-area networks. In *Proceedings of MobiSys, New York*, pages 303–316, May 2003.

[4] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption tolerant networks. In *IEEE INFOCOM*, Barcelona, Spain, April 2006.

[5] L. L. Cam and G. L. Yang. *Asymptotics in Statistics: Some Basic Concepts, 2nd Edition.* Springer, July 2000.

[6] L. J. Chen, Y. C. Chen, T. Sun, P. Sreedevi, K. T. Chen, C. H. Yu, and H. H. Chu. Finding self-similarity in opportunistic people networks. In *INFOCOM*, May 2007.

[7] L. J. Chen, C. H. Yu, T. Sun, Y. C. Chen, and H. H. Chu. A hybrid routing approach for opportunistic networks. In *Proceedings of ACM SIGCOMM workshop on Challenged Networks, CHANTS*, September 2006.

[8] Q. J. Chen, S. S. Kanhere, M. Hassan, and K.-C. Lan. Adaptive position update in geographic routing. In *Proceedings of IEEE International Conference on Communications*, June 2006.

[9] S. Deb, M. Medard, and C. Choute. Algebraic gossip: a network coding approach to optimal multiple rumor mongering. In *IEEE transaction on information theory*, volume 52, pages 2486–2607, June 2006.

[10] Q. Dong, J. Wu, W. Hu, and J. Crowcroft. Practical network coding in wireless networks. In *Proceedings of ACM MobiCom*, September 2007.

[11] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Feasibility study of bluetooth-based content distribution stations on public transit systems. In *MobiShare'06: Proceedings of the 1st international workshop on decentralized resource sharing in mobile computing and networking*, pages 63–65, September 2006.

[12] B. H. et al. Cartel: A distributed mobile sensor computing platform. In *Proceedings of ACM SenSys*, October 2006.

[13] L. Freeman. A set of measuring centrality based on betweenness. In *Sociometry*, pages 35–41, 1977.

[14] R. Groenevelt, P. Nain, and G. Koole. Message delay in mobile ad hoc networks. In *Proceeings of PERFORMANCE*, October 2005.

[15] Z. J. Hass and T. Small. A new networking model for biological applications of ad hoc sensor networks. In *IEEE/ACM Transactions on Networking*, volume 14(1), pages 27–40, February 2006.

[16] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless networks. In *Proceedings of ACM MobiCom*, pages 187–201, September 2004.

[17] W. J. Hsu and A. Helmy. On nodal encounter patterns in wireless lan traces. In *The 2nd IEEE Intl. Workshop on Wireless Network Measurement (WiNMee)*, April 2006.

[18] P. Hui, A. Chaintreau, J. Scott, R. Gass, C. Diot, and J. Crowcroft. Pocket switched networks and human mobility in conference environments. In *SIGCOMM*, August 2005.

[19] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay tolerant networks. In *MobiHoc*, May 2008.

[20] J. G. Jetcheva, Y. C. Hu, S. P. Chaudhuri, A. K. Saha, and D. B. Johnson. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In *Proceedings of fifth IEEE workshop on Mobile Computing Systems and Applications*, pages 32–43, October 2003.

[21] J. Jun, P. Peddabachagari, and M. L. Sichitiu. Theoretical maximum throughput of ieee 802.11 and its application. In *Network Computing and Applications*, pages 249–256, April 2003.

[22] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. Xors in the air: Practical wireless network coding. In *Proceedings of ACM SIGCOMM*, June 2006.

[23] J. Lebrun and C. N. Chuah. Delegation forwarding. In *Proceedings of MobiHoc*, pages 26–30, May 2008.

[24] Z. Li and B. Li. Network coding: the case of multiple unicast sessions. In *in Proceedings of the 42nd Allerton Annual Conference on Communication, Control, and Computing*, September 2004.

[25] Y. Lin, B. Li, and B. Liang. Stochastic analysis of network coding in epidemic routing. In *Proceedings of ACM MobiOpp*, June 2007.

[26] Y. Lin, B. Li, and B. Liang. Stochastic analysis of network coding in epidemic routing. In *ACM MobiOpp*, 2007.

[27] D. S. Lun, M. Medard, and R. Koetter. Network coding for efficient wireless unicast. In *International Zurich Seminar on Communications*, pages 74–77, February 2006.

[28] L. T. Philippe Flajolet, Danile Gardy. Birthday paradox, coupon collectors, caching algorithms and self-organizing search. In *Discrete Applied Mathematics*, volume 39, pages 207–229, 1992.

[29] L. Rudolf and N. Harald. *Finite Fields (2nd ed.)*. Cambridge University Press, 1997.

[30] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD data set cambridge/haggle (v. 2006-09-15). Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, Sept. 2006.

[31] G. Sharam and R. Mazumdar. On achievable delay/capacity trade-offs in mobile ad hoc networks. In *IEEE WiOpt*, 2004.

[32] T. Small and Z. Hass. Resource and performance tradeoffs in delay tolerant wireless networks. In *ACM SIGCOMM Workshop on Delay Tolerant Networks (WDTN)*, 2005.

[33] T. Small and Z. J. Hass. Quality of service and capacity in constrained intermittent-connectivity networks. In *IEEE Transactions on Mobile Compuitng*, volume 6(7), 2007.

[34] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient routing in intermittently connected networks. In *IEEE/ACM Transactions on Networking*, 2007.

[35] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of ACM SIGCOMM*, August 2005.

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

[37] S. Wang. Predicting the lifetime of repairable unicast routing paths in vehicle-formed mobile ad hoc networks on highways. In *Proceedings of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 2815–2819, September 2004.

[38] J. Widmer and J.-Y. L. Boudec. Network coding for efficient communication in extreme networks. In *Proceedings of ACM SIGCOMM workshop on delay tolerant networking and related topics (WDTN)*, August 2005.

[39] Y. Xi and E. M. Yeh. Distributed algorithms for minimum cost multicast with network coding in wireless networks. In *4th International symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, April 2006.

[40] E. Yoneki, P. Hui, and J. Crowcroft. Distinct types of hubs in human dynamic networks. In *SocialNets*, September 2008.

[41] X. Zhang, J. Kursoe, B. Levine, D. Towsley, and H. Zhang. Study of a bus-based disruption tolerant network: Mobility modeling and impact on routing. In *Mobicom*, September 2007.

[42] X. Zhang, G. Neglia, J. Kursoe, and D. Towsley. On the benefits of random linear coding for unicast applications in disruption tolerant networks. In *2nd Workshop on Network Coding, theory and applications (NetCod)*, April 2006.

[43] X. Zhang, G. Neglia, J. Kursoe, and D. Towsley. Performance modeling of epidemic routing. In *Elsevier Computer Networks*, 2007.