

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

Shabbir Ahmed

School of Computer Science and Engineering
The University of New South Wales
Sydney, Australia
shabbira@cse.unsw.edu.au

Salil S. Kanhere

School of Computer Science and Engineering
The University of New South Wales
Sydney, Australia
saliik@cse.unsw.edu.au

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, 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 simulate a large-scale vehicular DTN using empirically collected movement traces of a city-wide public transport network and demonstrate the efficacy of our solutions in comparison with other forwarding schemes.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: [Wireless communication]; C.2.2 [Network Protocol]: Routing protocols

General Terms

Measurement, Performance

Keywords

DTN, network coding, power law, routing, VANET

1. INTRODUCTION

Delay Tolerant Networks (DTN) are a type of challenged networks, wherein the contacts between the communicating devices are intermittent. Consequently, an 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* [14], wherein personal communication devices carried by humans self-organize to form an intermittently connected network and (ii) *Vehicle-based DTN* [3, 10], 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 [14, 4, 30]. In such networks, a small percentage of the nodes, often referred to as *hubs* [29], 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 [15, 19]. 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 than 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 short [14]. Consequently, the most popular nodes, which concentrate all of the forwarding traffic, can often only exchange limited number of messages. Hence, their performance is significantly hampered.

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 in the data conduit using random linear network coding, wherein multiple messages addressed to the same destination are combined to form a single encoded message. Consequently, hubs only forward encoded messages, which can be readily accomplished in short contact periods, and also reduces the associated routing costs. Further, 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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MSWiM'09, October 26–29, 2009, Tenerife, Canary Islands, Spain.
Copyright 2009 ACM 978-1-60558-616-9/09/10 ...\$10.00.

coefficient matrices of the encoded messages to make the forwarding decisions. 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 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 incur significant savings in delivery costs (at least half of that of other schemes).

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 presents the details of the HubCode schemes. In Section 4, we present the results from our simulations. Finally, Section 5 concludes the paper.

2. RELATED WORKS

Ahlsweede 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 [18, 8, 22, 20] as well as broadcast transmissions [28]. Lun et al. [22] and Li et al. [20] present theoretical results on the application of network coding for unicast transmissions. The work presented in [18] and [8] focuses on practical issues. They demonstrate that network coding can benefit from leveraging the broadcast advantage in wireless networks. In [18], 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 [27, 31, 21, 5] have studied the use of network coding in DTN. Zhang et al. [31] and Widmer et al. [27] 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 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 [2, 12, 14, 13, 30] 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 [11] defined several centrality metrics to measure the importance of a node in a network. Researchers in [15, 19] have proposed forwarding strategies that exploit the existence of the scale-free structure in the underlying network. In BubbleRap [15], 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 [19], 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.

3. HUBCODE

As highlighted in the introduction, empirical analysis of the mobility patterns of several people-centric DTN [14, 4, 30] 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. 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. 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 [31]. 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.

3.1.1 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. 1), and forwards the encoded message to the hub. The coding technique is described below.

3.1.2 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, \dots, X_k with a common destination. Then the hub creates a linear combination [7, 31] of these k messages to form a single encoded message F_l , using Eq. 1,

$$F_l = \sum_{i=1}^k a_i X_i, \quad a_i \in \mathbb{F}_q \quad (1)$$

where a_1, a_2, \dots, a_k , represent the coefficients, which are randomly selected from a finite field [23], \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 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,

$$F_1 = a_{11}X_1 + a_{12}X_2 + a_{13}X_3 \quad (2)$$

$$F_2 = a_{21}X_1 + a_{22}X_2 + a_{23}X_4 \quad (3)$$

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 $A_1, \langle a_1, a_2 \rangle$ (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:

$$A_1 = \begin{vmatrix} idx_1 & idx_3 \\ a_1 & a_2 \end{vmatrix}, \quad A_2 = \begin{vmatrix} idx_1 & idx_2 \\ a_3 & a_4 \\ a_5 & a_6 \end{vmatrix}$$

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:

$$A_2 = \begin{vmatrix} idx_1 & idx_2 & idx_3 \\ a_3 & a_4 & 0 \\ a_5 & a_6 & 0 \\ a_1 & 0 & a_2 \end{vmatrix}$$

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.

3.1.3 Destination

When the hub encounters a destination, it forwards an encoded message to it. Similar to the hubs, the destination also

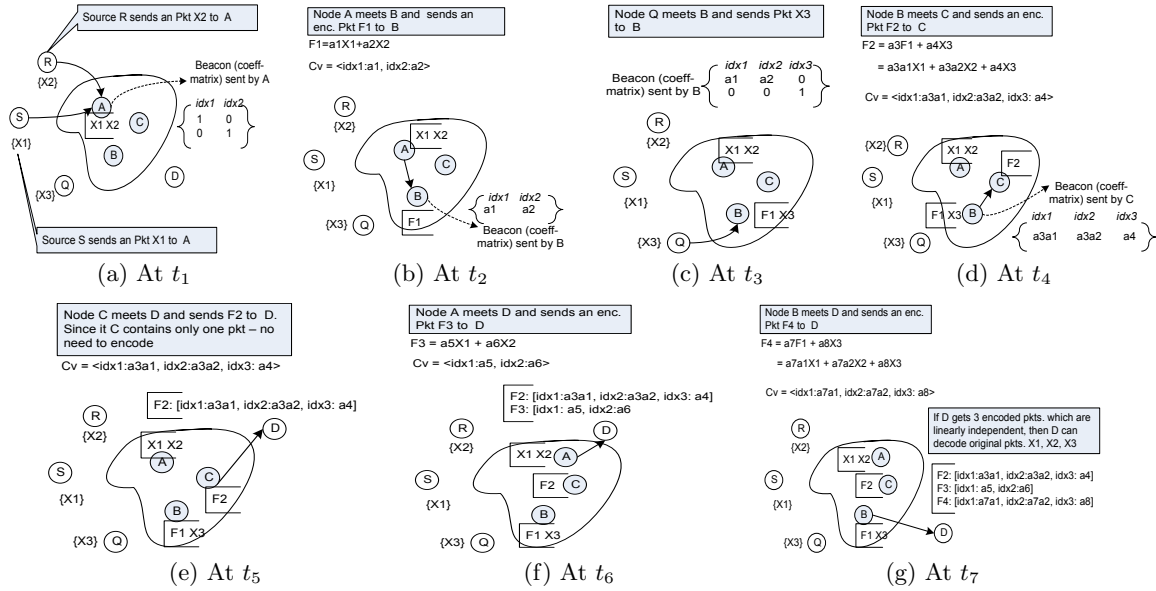


Figure 1: Example illustrating the operation of HubCodeV1

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:

$$x = A^{-1}f \quad (4)$$

Fig. 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 [14], 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.

3.2.1 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 possess 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.

3.2.2 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.

3.2.3 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.2 highlights the basic operation of HubCodeV2. We have used the same scenario as in the example for HubCodeV1.

4. 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 the first set of experiments, we assume an ideal scenario, in which, the nodes encountering each other can exchange all desired messages (i.e. infinite bandwidth). In the next set of simulations, we take a more 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 final set of simulations, we evaluate the impact of changing the number of nodes that are classified as hubs on the performance of the HubCode schemes.

4.1 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 (blue-tooth, zigbee, etc) are either handed to a volunteer group [14] or are mounted on moving vehicles [3, 9]. 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 1). 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 [16]. 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 [6]. The traces can be readily used to simulate a bus-based DTN, similar to DieselNet [30]. As in [30], 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 real-world people-centric DTN.

4.2 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 1165 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

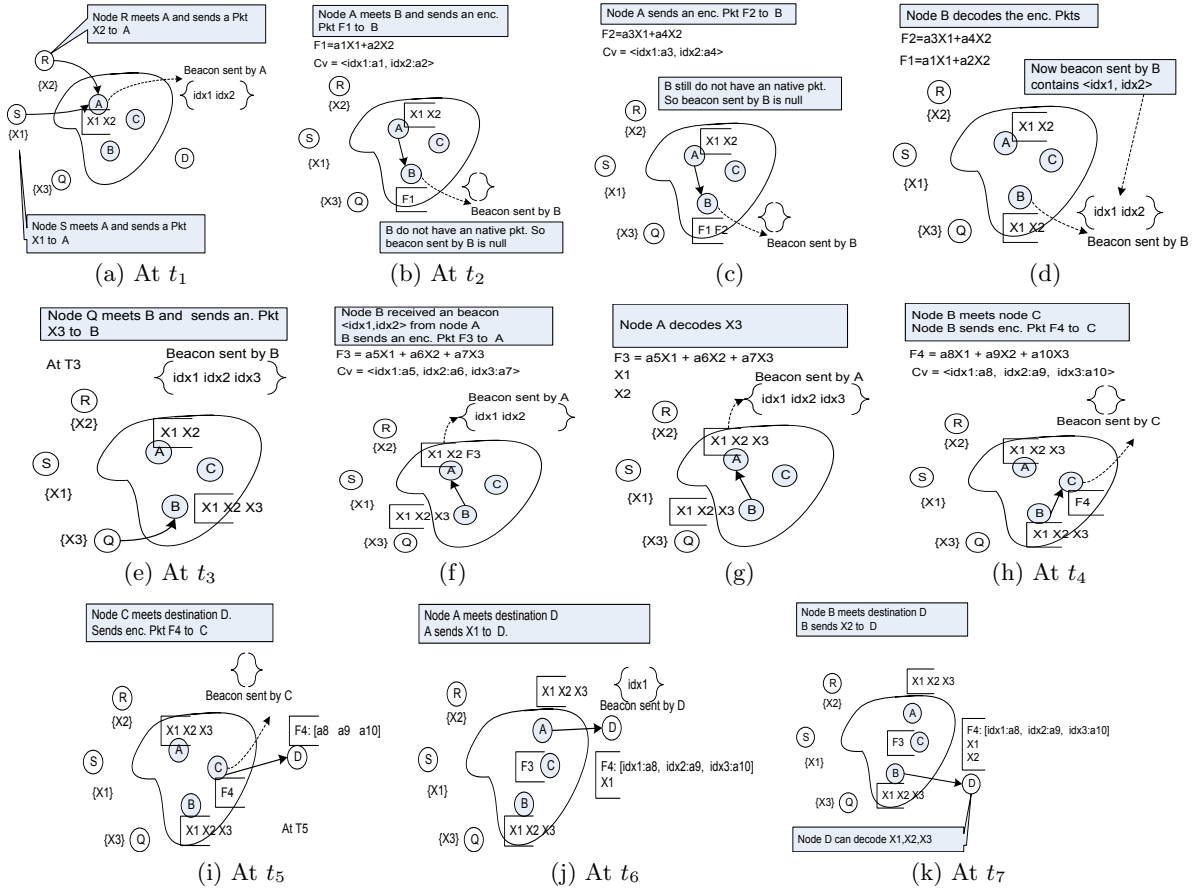


Figure 2: Example illustrating the operation of HubCodeV2

Table 1: Properties of Mobility Traces

Data Set	Intel[24]	Cambridge Comp Lab[24]	INFOCOM'05[14]	Seattle Buses[16]
Period	05-01-06 to 05-01-11	05-01-25 to 05-01-31	05-03-07 to 05-03-10	01-10-30 to 01-11-26
Device	iMote	iMote	iMote	None
Network	Bluetooth	Bluetooth	Bluetooth	WiFi (simulated)
Nodes	9	12	41	1163
Coverage	Top 2 covered 9	Top 2 covered 12	Top 4 covered 41	Top 100 covered 865

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 with several other DTN forwarding schemes. *Epidemic* (i.e. flooding) [26] is included since it achieves the highest delivery ratio in the ideal scenario. We choose *Spray and Wait* [25], 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 [25]. *RLC* [31] and *BubbleRap* [15] are included since they are closely related to our work (see Section II). We use the same methods as described in [15] 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. In

the last set of simulations, we study the impact of varying the number of hubs on the performance of our schemes.

4.3 Ideal Scenario

In this scenario, we assume that the network has infinite bandwidth. In other words, when two nodes encounter each other, they can exchange all required messages. As a result, the exchange of auxiliary information in the beacons has no impact on the actual data exchange. Figs. 3 and 4 plot the evolution of the delivery ratio and delivery cost as a function of time. Note that, the latter graph uses a log scale for the y-axis. Recall that, messages are injected between 3000 to 4000 seconds. As expected, Epidemic achieves the highest delivery ratio (85%), since it resorts to flooding. However, the delivery cost is quite significant. Both versions of HubCode come a close second in terms of the delivery ratio, indicating the efficacy of this approach. Recall that, the primary difference between the two versions of HubCode is in the amount of auxiliary information included in the beacons. Since, this has no impact on the actual data transfer in the ideal scenario, both schemes exhibit similar performance. Fig. 4 illustrates that the delivery cost of these schemes is a fraction of that incurred in Epidemic (a reduction by an order of 3). This is expected since, in our schemes only the hubs, which form 10% of the total nodes, are involved in forwarding messages. Further, the use of network coding results in additional savings. Also observe that HubCodeV1 achieves a 10% reduction in the delivery cost as compared to that of HubCodeV2. In HubCodeV1, the hubs exchange their complete coefficient matrices in the beacons whereas in HubCodeV2, hubs only exchange a list of the native decoded messages (ignoring any encoded messages). As a result, in the former, hubs are able to make more informed decisions about forwarding of the required messages, thus avoid some of the redundant messages that are exchanged in the latter.

The performance of our proposed schemes are comparable to that of RLC in terms of the delivery ratio. However, the delivery cost of our schemes are less than that of RLC by an order of magnitude (Fig. 4). This is attributed to the fact that unnecessary message spreading of RLC only increases delivery cost but do not yield high delivery ratio.

BubbleRap and Spray and Wait exhibit similar performance in terms of the delivery ratio and are outperformed by all schemes. Since, BubbleRap is a gradient forwarding scheme, its delivery ratio primarily depends on the quality of the forwarders - more specifically, the quality of the local and global ranking schemes. A closer inspection on the simulator's log revealed that BubbleRap often chose inferior forwarders because of incorrect ranking of nodes. We have used one of the methods recommended in [15] to obtain local and global rankings of nodes. However, as mentioned by the authors, there are several alternate methods, which can be used to generate these rankings, which may affect the performance. For similar reasons, the delivery cost of this scheme is about twice of that of the HubCode schemes. The poor performance of Spray and Wait can be attributed to the large population of the network (over 1000 nodes) and the large diameter (7). As a result, this scheme is unable to find suitable forwarders, which can deliver the messages to the destination. Further, the delivery cost incurred by Spray and Wait is almost an order greater than HubCode. This can be attributed to the fact that this scheme employs a restricted version of flooding.

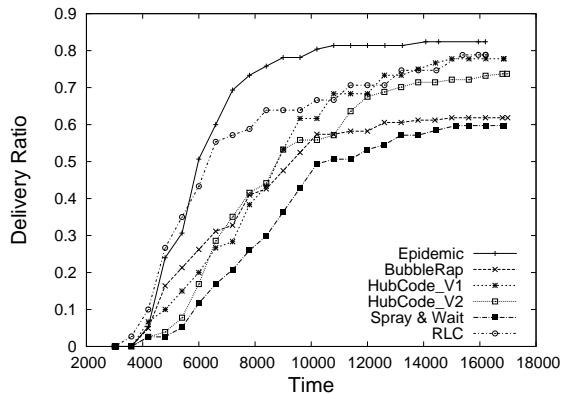


Figure 3: Delivery Ratio (ideal scenario)

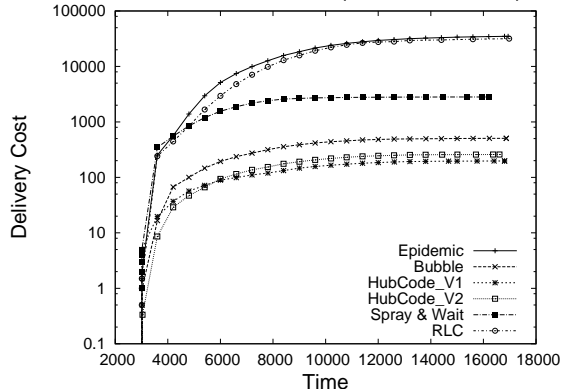


Figure 4: Delivery Cost (ideal scenario)

4.4 Realistic Scenario

Recall that, in the previous set of simulations, we had assumed that nodes can exchange unrestricted data during each contact duration. However, in reality, contact durations are finite and in most cases, short lived. 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 [14, 9]. Hence, in this scenario, we adopt a pragmatic approach and relax the infinite bandwidth assumption. 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,

$$D = (T_c - T_a) \times 4Mbps \quad (5)$$

Empirical experiments have shown that in 802.11b, the typical goodput (accounting for overheads) at the highest data rate of 11Mbps is around 4Mbps [17]. 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. The rest of the simulation parameters are similar to the previous experiments.

Figs. 5 and 6 plot the delivery ratio and delivery cost, respectively for all the forwarding schemes. It is evident that under realistic assumptions, the delivery ratio for all schemes reduce as compared to the ideal case. However,

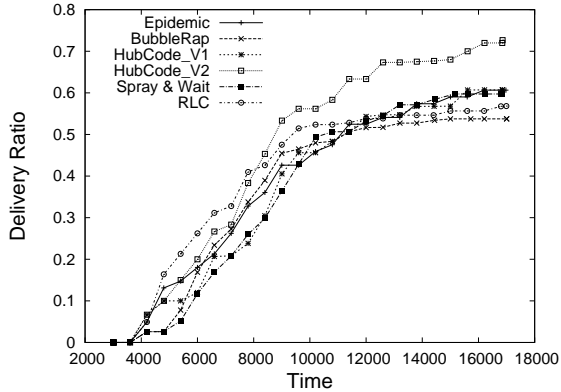


Figure 5: Delivery Ratio (Realistic Scenario)

the impact on different schemes varies significantly. The delivery ratio for HubCodeV2, drops from 76% in the ideal scenario to about 72%. Further, observe that this scheme outperforms all other schemes by about 15% – 20%. On the contrary the delivery ratio for HubCodeV1 drops from 78% to 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. 6 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 drops significantly (from 85% to 60%) in the realistic scenario. This is because a node may not be able to transfer all the messages it carries to other nodes during an encounter. In a realistic environment, 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.

4.5 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 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 the realistic scenario in Section 4.4. Figs. 7 illustrates the impact of increasing the percentage of nodes that constitute

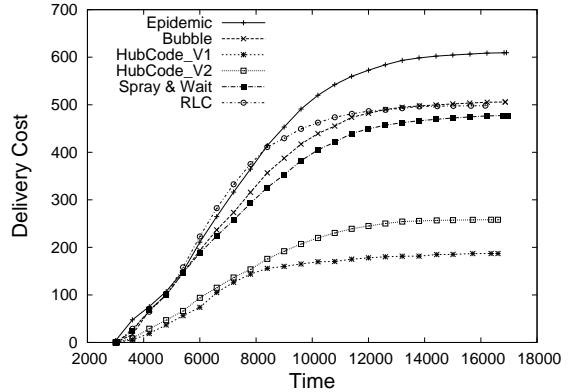


Figure 6: Delivery Cost (Realistic Scenario)

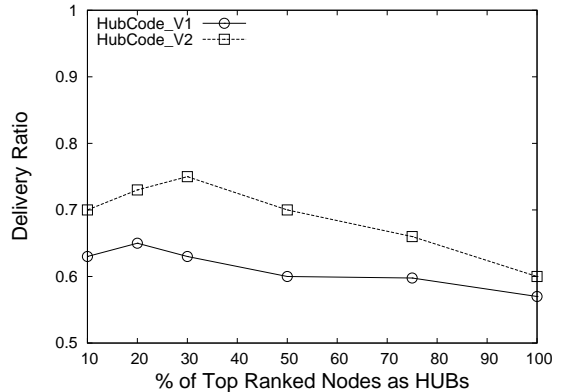


Figure 7: Delivery ratio as a function of the percentage of hubs

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 (20% for V1 and 30% for V2), 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. Note that, when 100% of the nodes are regarded as the hubs, HubCode degenerates into RLC. The delivery cost (not presented in this paper due to space constraints) continues to increase with an increase in the number of hubs. This is expected, since, greater the number of hubs, the more the traffic exchanged between them, which increases the overall cost of the scheme. These results suggest that only a small percentage of the nodes (20% – 30%), which are highly connected should be designated as hubs.

5. 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 implementations of HubCode to address the important trade-off between routing overhead and computa-

tional 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.

6. REFERENCES

- [1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung. Network information flow. In *IEEE Transaction on Information Theory*, volume 46, July 2000.
- [2] M. Balazinsky and P. Castro. Characterizing mobility and network usage in a corporate wireless local-area networks. In *Proceedings of MobiSys*, May 2003.
- [3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption tolerant networks. In *Proceedings of IEEE INFOCOM*, Barcelona, Spain, April 2006.
- [4] 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 *Proceedings of IEEE INFOCOM*, May 2007.
- [5] 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*, September 2006.
- [6] 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.
- [7] S. Deb and M. Medard. Algebraic gossip: A network coding approach to optimal multiple rumor mongering. In *Proceedings of Allerton*, September 2004.
- [8] Q. Dong, J. Wu, W. Hu, and J. Crowcroft. Practical network coding in wireless networks. In *Proceedings of ACM MobiCom*, September 2007.
- [9] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Feasibility study of bluetooth-based content distribution stations on public transit systems. In *Proceedings of the 1st international workshop on decentralized resource sharing in mobile computing and networking, MobiShare*, September 2006.
- [10] B. H. et al. Cartel: A distributed mobile sensor computing platform. In *Proceedings of ACM SenSys*, October 2006.
- [11] L. Freeman. A set of measuring centrality based on betweenness. In *Sociometry*, pages 35–41, 1977.
- [12] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless networks. In *Proceedings of ACM MobiCom*, September 2004.
- [13] W. J. Hsu and A. Helmy. On nodal encounter patterns in wireless lan traces. In *Proceedings of the second IEEE International Workshop on Wireless Network Measurement (WiNMee)*, April 2006.
- [14] P. Hui, A. Chaintreau, J. Scott, R. Gass, C. Diot, and J. Crowcroft. Pocket switched networks and human mobility in conference environments. In *Proceedings of SIGCOMM*, August 2005.
- [15] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay tolerant networks. In *Proceedings of MobiHoc*, May 2008.
- [16] 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*, October 2003.
- [17] J. Jun, P. Peddabachagari, and M. L. Sichitiu. Theoretical maximum throughput of IEEE 802.11 and its application. In *Proceedings of Network Computing and Applications*, April 2003.
- [18] 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.
- [19] J. Lebrun and C. N. Chuah. Delegation forwarding. In *Proceedings of MobiHoc*, May 2008.
- [20] Z. Li and B. Li. Network coding: the case of multiple unicast sessions. In *Proceedings of the 42nd Allerton Annual Conference on Communication, Control, and Computing*, September 2004.
- [21] Y. Lin, B. Li, and B. Liang. Stochastic analysis of network coding in epidemic routing. In *Proceedings of ACM MobiOpp*, June 2007.
- [22] D. S. Lun, M. Medard, and R. Koetter. Network coding for efficient wireless unicast. In *Proceedings of International Zurich Seminar on Communications*, February 2006.
- [23] L. Rudolf and N. Harald. In *Finite Fields (2nd ed.)*. Cambridge University Press, 1997.
- [24] 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>.
- [25] 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.
- [26] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. In *Technical Report CS-200006*, Duke University, 2000.
- [27] 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.
- [28] Y. Xi and E. M. Yeh. Distributed algorithms for minimum cost multicast with network coding in wireless networks. In *Proceedings of the 4th International symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, April 2006.
- [29] E. Yoneki, P. Hui, and J. Crowcroft. Distinct types of hubs in human dynamic networks. In *SocialNets*, September 2008.
- [30] 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 *Proceedings of MobiCom*, September 2007.
- [31] X. Zhang, G. Neglia, J. Kursoe, and D. Towsley. On the benefits of random linear coding for unicast applications in disruption tolerant networks. In *Proceedings of the 2nd Workshop on Network Coding, theory and applications (NetCod)*, April 2006.