Experiments on the Spatial Distribution of Network Code Diversity in Segmented DTNs

Proceedings of the ACM MobiCom Workshop on Challenged Networks (CHANTS). 2011

Brenton Walker[1], Calvin Ardi[1], Agoston Petz[2], Jung Ryu[2] and Christine Julien[2]

[1] Laboratory for Telecommunications Sciences College Park, MD, USA, {brenton, calvin} @ltsnet.net
[2] The University of Texas-Austin, Austin, TX, USA, {agoston, jung.ryu, c.julien}@utexas.edu

TR-ARiSE-2011-009

ARiSE
Advanced Research in
Software Engineering

Experiments on the Spatial Distribution of Network Code Diversity in Segmented DTNs [*]

Brenton Walker, Calvin Ardi
Laboratory for Telecommunications Sciences
College Park, MD, USA
{brenton, calvin}@ltsnet.net

Agoston Petz, Jung Ryu and
Christine Julien
University of Texas-Austin
Austin, TX, USA
{agoston, jung.ryu, c.julien}@utexas.edu

ABSTRACT

Biologists have observed that when a small colony breaks off from a larger population, the colony tends to have less genetic diversity. This phenomenon, called the "founder effect", has an analogy in delay-tolerant networks that employ network coded routing to disseminate large bundles of data. In this paper, we study the spread of information diversity through various experiments using a network coding DTN router. The scenarios we investigate include single communities with "mixing nodes", segmented communities with occasional travelers between sub-groups, and real encounter traces. Our experiments are carried out on the VirtualMeshTest testbed, which allows us to perform large trials with real implementations communicating using commodity wireless cards over emulated RF channels.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Store and forward networks

General Terms

Design, Experimentation, Performance

1. INTRODUCTION

Network coding is enjoying a burgeoning research interest as an alternative to traditional routing protocols due to its resilience to route failures, low overhead, its natural resistance to a variety of network attacks such as eavesdropping and replay-attacks [4, 9], and its information-theoretic properties [2]. Nodes employing network coded routing do not forward packets in the traditional sense but instead create and send linear combinations of the packets they are buffering. When a destination node receives a linearly independent set of such combinations, it is able to decode the entire set. Network coding allows for more efficient transfers of information when communicating over a broadcast medium such as wireless. It can also be employed to add controlled amounts of redundancy in ways that improve stream protocols such as TCP [15, 18].

We focus on network coding in delay-tolerant networks (DTNs) [8], which may lack end-to-end connectivity and must rely on intermediate nodes to store, carry, and forward messages. Applications of DTNs include remote villages, which can connect to the Internet only intermittently, military networks where infrastructure is not available or impractical, and deep space communications. In this paper we explicitly differentiate network coding (where all nodes can create new linear combinations) from erasure coding (where only the source can create linear combinations).

In DTNs, like in peer-to-peer networks, network and erasure coding provide benefits such as better efficiency and bandwidth utilization [20], and several ideas have been proposed for using DTNs for information dissemination, including podcasting [3] and localized data dissemination in both social and vehicular networks [14, 19]. Furthermore, Ho *et al.* [10] and Chou *et al.* [5] developed randomized network coding that makes the algorithm practical in distributed networks and robust to network dynamics. In [12] Lin *et al.* develop an analytical model of a simple DTN network coding protocol that is similar to what we have implemented. Chuah *et al.* [6] have also simulated several erasure code distribution schemes under a variety of mobility scenarios.

Our work is motivated by a desire to see if network and erasure coding in DTNs is practically achievable and to understand the design choices relevant to network coding in dynamic and opportunistic DTN environments. To that end we have built **SimpleNC**, a network coding router for the DTN2 Reference Implementation [7] and carried out a series of controlled mobile experiments on real wireless nodes. We treat the DTN as a distributed storage medium, where many nodes hold some set of coded fragments and one or more "harvester" nodes try to collect a full basis for the message. In this sense, we are focused on a broad use case in which there is not necessarily a single destination for a piece of information. We chose this approach as it is a generalization of a variety of use cases for a DTN, including as a unicast network (i.e., when there is only a single "harvester"), a multicast content distribution network, or even as a system for distributed storage among resource constrained nodes.

---

2. SYSTEM DESIGN

In this section, we explain the basics of network coding as we use it, clarify our terminology, and describe our router implementation.

Bundle: the fundamental data unit of the bundle protocol [17]. Every bundle has a globally unique identifier (*GUID*).

Fragment x_i: a bundle is split into M (non-encoded) fragments of k bits, such that $x_i \in \mathsf{GF}(2)^k$. Each fragment is associated with its parent bundle's *GUID*.

Coefficient vector c: a vector, $c = \langle c_1, c_2, \ldots, c_M \rangle$, $c_i \in \mathsf{GF}(2)$ that indicates the fragments that were xor'ed together to create a coded fragment.

Coded fragment / Codeword w_c: an bundle containing some linear combination of fragments such that $w_c = \sum_{i=1}^{M} c_i x_i$ for some coefficient vector c.

(Re)encoding vector d: on a node with r coded fragments, a (re)encoding vector $d = \langle d_1, d_2, \ldots, d_r \rangle$ $d_i \in \mathsf{GF}(2)$ can be chosen to create a new coded fragment $w_{c'} = \sum_{i=1}^{r} d_i w_i$. (Re)encoding is only allowed between coded fragments associated with the same original bundle (i.e., with the same *GUID*).

Each coded fragment c corresponds to a row with elements $\langle c_1, c_2, \ldots, c_M \rangle$ in an $M \times M$ decoding matrix. When the destination receives enough coded fragments to construct a full-rank decoding matrix, the original bundle can be recovered by matrix inversion. Our router does network coding over $\mathsf{GF}(2)$, however other finite fields could be used.

By **Network Coding (NC)** we mean that the re-encoding weight is variable on all nodes. That is, all nodes may create and send new linear combinations of existing coded fragments. By **Erasure Coding (EC)** we refer to the special case of network coding where only the source node is configured to create new linear combinations, and non-source nodes may only forward coded fragments they already have.

2.1 SimpleNC Router Overview

Our network coding router, called SimpleNC, was developed as an integrated routing module in the DTN2 reference implementation [7]. The way the router conveys coded fragment metadata, such as coefficient vectors in extension blocks, and sets the administrative fields of the coded fragment bundles is based on a preliminary Internet-Draft by Caro and Zinky [1]. The full details of the implementation are beyond the scope of this paper, however we provide a summary here.

In SimpleNC, when an **original bundle** is received from the network, through the API, or loaded from the database, the node checks to see if the received bundle is already a coded fragment. If not, and the bundle is larger than some threshold, the node splits it into fixed-size fragments. Each fragment is tagged with its corresponding coefficient vector, c, and the GUID of the original bundle for record keeping.

Internally, the SimpleNC router manipulates **NCBundle** objects. NCBundle is a wrapper class for the DTN2 Bundle class, which contains the Bundle itself and metadata such as the coefficient vector and GUID of the original bundle. The NCBundles are grouped into **NCBundleCollection** objects that contain all of a node's NCBundles associated with a particular GUID. The NCBundleCollection is the central object handled by the SimpleNC router, as each one corresponds to an original bundle whose coded fragments are
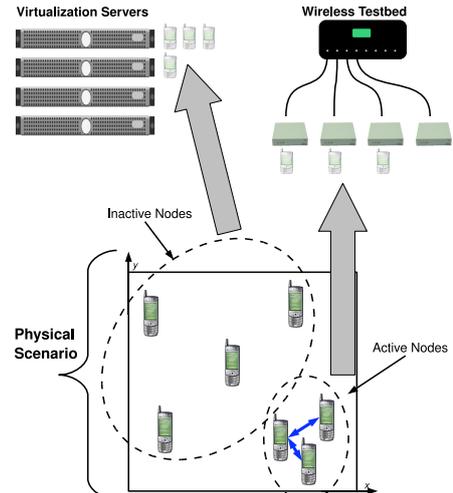

Figure 1: The premise behind the VMT system.

being distributed through the network. Each NCBundleCollection tracks the space spanned by its NCBundles and drops any received coded fragments that are not *innovative*. It maintains a partially reduced matrix whose rows span the same space as its current NCBundles. When a new NCBundle is received, its coefficient vector is appended to this matrix and its rank tested. SimpleNC uses the m4ri [13] binary linear algebra library to do these matrix manipulations.

When transmitting, SimpleNC randomly chooses a NCBundleCollection and some subset of the NCBundles in that collection. It then creates and sends a new bundle whose payload is the xor of the payloads of the selected NCBundles. The Hamming weight, H, of the re-encoding vector is configurable but defaults to $H \leq \log_2(\texttt{rank})$, where `rank` is the rank of the NCBundleCollection's inventory.

3. EXPERIMENTAL PLATFORM

Our experiments were carried out on the VirtualMeshTest (VMT) mobile wireless testbed [11]. VMT allows us to subject real wireless nodes running real DTN stacks to emulated mobile environments. VMT improves on the older MeshTest system by running the nodes inside virtual machines, allowing users to carry out much larger DTN experiments.

The premise behind the VMT system is shown in Figure 1. Nodes that are in wireless range of other nodes in the underlying physical scenario will have their images migrated to the wireless hosts. Nodes that are isolated will be run on the virtualization servers. The wireless testbed is effectively a many-to-many analog channel emulator based on an array of programmable attenuators. Given a desired physical arrangement of nodes, VMT computes what the path loss between nodes should be and programs the attenuators to achieve those path losses. By updating the attenuations every second, VMT can emulate the dynamic connections of a mobile wireless environment for real wireless nodes.

The current VMT system has two 8-port RF switches and two offline virtualization servers. The wireless physical nodes have Broadcom BCM4321 802.11 WiFi cards, which we run in 802.11b mode with effective maximum throughput of about 5Mb/s.
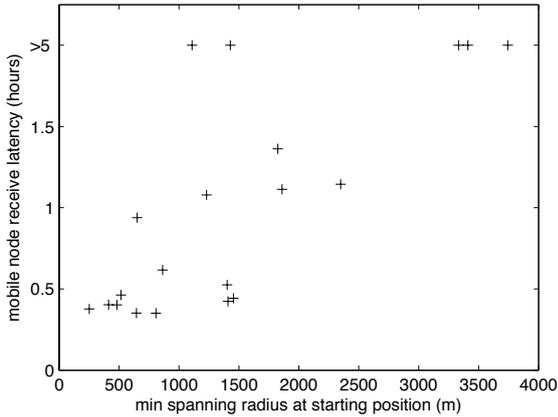
Figure 2: Minimum spanning radius vs latency for a mobile "harvester" node.

Figure 3: Contour plot of the minimum spanning radius in a 5km×5km area at the end of a 10-hour 50-node experiment.

4. EXPERIMENTAL METRICS

The specific metrics we study here are focused on understanding and quantifying the founder effect phenomenon. We define three metrics: the r-local rank, minimum spanning radius, and group rank.

The effective distribution of network code diversity has an effect on the speed and reliability with which a "harvester" node can collect a full basis of coded fragments from its peers. Consider a geographic area containing mobile and stationary nodes in which each node holds some (possibly empty) collection of coded fragments generated from some original bundle. We may be interested in how much time and effort it would take a mobile harvester node to collect enough coded fragments to recover the original bundle. Note that simply adding up the ranks of the coefficient vectors of a set of nodes is not a good metric for assessing a geographic area's diversity, as the coded fragments in the nodes' collections may be linearly dependent, even if they are all unique.

For any point (x, y) in an experimental area, we consider a circle, $B_r(x, y)$, of radius r centered at that point and the collection of nodes enclosed by that circle. Then for any triple (x, y, r), we can compute the rank of the collection of all coefficient vectors of nodes inside the circle. We call this the **r-local rank** at location (x, y) and time t and denote it by $R_r^t(x, y)$.

Given this function, we can compute the distribution of r-local ranks over some sample set of points in the space. Let $S \subset \mathbb{R}^2$ be a set of sample points in the experimental area. Then the average r-local rank is

$$\overline{R_r^t(S)} = \frac{1}{|S|} \sum_{s \in S} R_r^t(s) \qquad (1)$$

A second more concise metric is based on taking the smallest circle around any point in the space that encloses a full basis for a particular bundle. Then for any point (x, y) and time t, we can define the **minimum spanning radius**

$$R_{\text{span}}^t(x, y) = \arg\min_r \left[R_r^t(x, y) = M \right] \qquad (2)$$

where M is the length of the coefficient vector (*i.e.*, full rank). For this metric, small values indicate a high concen-

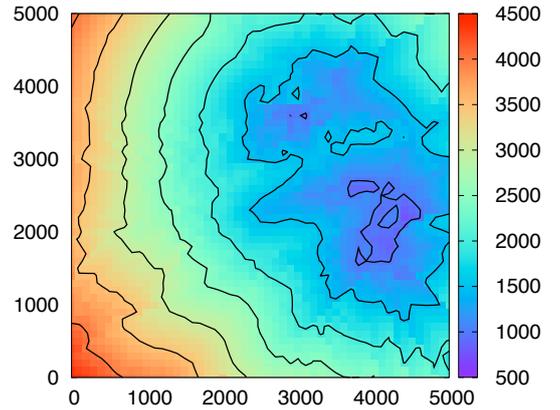tration of code diversity. A contour plot of the minimum spanning radius for one bundle at the end of an experiment is shown in Figure 3. The relationship between minimum spanning radius and the time it takes a harvester node to collect a full basis for a bundle is shown in Figure 2. As one would expect, the harvester tends to collect a full basis more quickly when it starts out in areas of higher code diversity.

In some networks it is natural to group the nodes by some non-geographic criteria: nodes can belong to different communities or play different roles in a network. It may not be appropriate to apply the geographically-oriented minimum spanning radius metric in such networks. Additionally, we would at times like to track the rank of the collection of all coefficient vectors in a particular, arbitrarily defined, set of nodes. This can be done using a metric very similar to the r-local rank: let I be some set of nodes and define the **group rank**, R_I^t to be the rank of the matrix of coefficient vectors held on nodes in the set I at time t.

5. EXPERIMENTS AND RESULTS

Our experiments focus on understanding the founder effect phenomenon and quantifying the spread of network code diversity. Our main comparison is between Erasure Coding (EC, all linear combinations generated at the source) and Network Coding (NC, linear combinations generated at all nodes). One expects that network coding will be more effective at distributing linearly independent fragments, but it is not clear how much more effective it will be, or how computationally expensive dynamically generating new coded fragments and rank-checking will be in practice. To realistically evaluate these trade-offs, it is essential to perform the experiments using a real implementation on real hardware.

5.1 Mobile Source and Mixing Nodes

A simple scenario that allows us to experiment with the spread of code diversity in a non-segmented network is the **mobile source with mixing nodes** scenario. In this experiment 48 stationary **relay nodes** are placed uniformly randomly in a 5km × 5km area, and a single mobile **source node** travels through the area with an average speed of 7 ± 3m/s. The source node performs a Manhattan-grid ran-
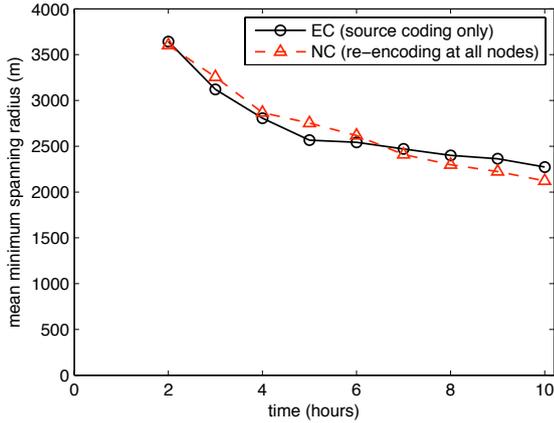
Figure 4: The mean minimum spanning radius for one bundle in two of the experiments described in section 5.1.

Figure 5: Three groups of nodes with three mobile nodes traveling between them.

dom walk for 5 hours, distributing coded fragments to relays it encounters. After 5 hours the source is removed, and a mobile **mixing node**, initially with no data, does a random walk through the area for 5 hours. This could be followed or accompanied by more mobile mixing nodes, but in these experiments we only use one.

In these experiments the source starts with ten 100MB bundles, each fragmented into 1000 100kB standard basis elements. When the source node meets a relay node, it generates and sends new linear combinations of its fragments, with a maximum hamming weight of $\log_2(1000) \approx 10$. When relay nodes meet a mixing node, they can either generate new linear combinations from their store or send their existing ones. The mixing node further distributes coded fragments to the relays it meets.

5.1.1 Analysis

In the NC experiments, the mixing node collected a full basis for all ten 100MB bundles after 180-200 minutes. Coded fragments from each original bundle are sent with equal probability, so the ten bundles tend to complete at about the same time. Figure 4 shows the minimum spanning radius vs. time for one 100MB bundle in EC and NC experiments. We see that in a two-hop non-segmented network such as this, NC and EC have similar performance. This is expected since the relays only communicate with the mobile nodes, and source nodes generate new coded fragments under both NC and EC configurations. In these experiments the NC mixing node decreases the minimum spanning radius slightly faster, but the difference is negligible.

5.2 Island-Hopping

To observe the effects of network coding in segmented DTNs, we experiment with scenarios in which nodes are geographically separated into isolated communities, or **islands**, and only a small number of mobile **traveler nodes** occasionally move between islands, as pictured in Figure 5. We refer to this scenario as **island hopping**.

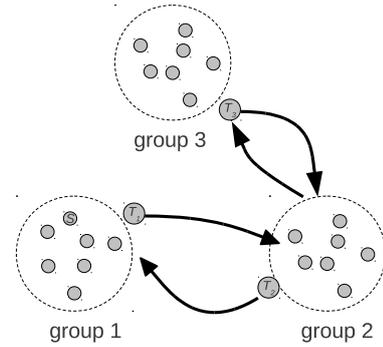We placed 47 stationary nodes, grouped into three islands of sizes 15, 20, and 12 nodes, into a 15km×15km area. The nodes are arranged uniformly randomly in the island regions, which range from 3km×3km to 4km×4km and are separated from each other by at least 10km.

There are three mobile nodes. Each goes through several travel cycles during an experiment. In each travel cycle, the mobile node does a Manhattan random walk around its home island for 60 minutes, spends 2 minutes traveling to another island, does a random walk on the visited island for 15 minutes, and spends 2 minutes traveling home. The travel time is unrealistically fast, but irrelevant since the nodes are in transit at the same time and no communication takes place while the nodes are in transit.

	Home island	Visited island
Mobile node 1	1	2
Mobile node 2	2	1
Mobile node 3	3	2

In our experiments four of the stationary nodes in island 1 are source nodes. Each one sends a 50MB bundle at the start of the experiment, which is divided into 500 100kB fragments and distributed using our SimpleNC router. We investigate the extent to which the bundles are transferred to islands two and three.

5.2.1 Analysis

If a single node from the source group visits group 2, it will distribute coded fragments spanning some subspace of the original message. After the visiting node returns to group 1, no matter how much the nodes in group 2 exchange and recombine those coded fragments, they will never span a space larger than what the traveler dropped off. This is analogous to the biological founder effect in that the diversity of the vectors in the second group will be a subset of the diversity of the vectors of the source group. Furthermore, when the mobile node from group 3 visits group 2 and returns home, the space ultimately spanned by the coefficient vectors in group 3 will be a subspace of that spanned by the coefficient vectors in group 2. While the founder effect in such a situation is a simple mathematical fact, not all methods of managing vectors in the second group perform equally well. As we see below, NC propagates innovative coded fragments to group 3 much more effectively.

We use the **group rank** metric to track the rank of the collection of all coefficient vectors in each island. Figure 6
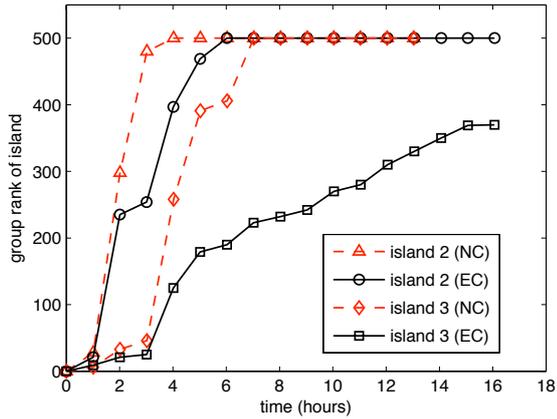
Figure 6: Island-hopping scenario: the group rank of the coded fragments of a bundle in islands 2 and 3 vs time, for both NC and EC.

Figure 7: The average group rank for random subsets of nodes in the haggle-based experiments.

shows the group rank for one bundle in islands 2 and 3 during an experiment. Using NC, the bundle is completely transferred to island 2 after 4 hours and to island 3 after 7 hours. Using EC, the bundle takes an additional 2 hours to completely transfer to island 2 and is still far from being completely transferred to island 3 after 16 hours. In general we have observed that the growth in an island's group rank is considerably faster when using network coding.

This difference in performance between NC and EC is expected in a segmented scenario such as this. As a simple example, consider a situation where we allow only source-encoding (EC), and after one cycle mobile nodes 1 and 2 manage to transfer 100 basis vectors to nodes in island 2. During the next home-island phase, node 2 can circulate copies of those basis vectors amongst nodes in island 2, but because no new linear combinations can be formed, the transfer of coded fragments to mobile node 3 on its next visit will be no better than random flooding of fragments without any coding. Especially once node 3 has collected most of the coded fragments in island 2, the probability that it acquires the remaining coded fragments in any transfer falls rapidly, as $1 - m/n$, where m is the number of coded fragments collected by node 3, and n is the total number of coded fragments held in island 2.

5.3   Real Encounter Traces

The Haggle Infocom encounter trace [16] consists of a log of sightings between Bluetooth devices carried by participants at an IEEE conference. Since no location data was recorded, we synthesized node locations to match the grouping of the nodes observed by the traces. We grouped the nodes at every time step by single-linkage clustering and placed all nodes in a group in close proximity on the testbed. This has the drawback that each group is fully connected, which may not have actually been the case in an actual physical arrangement of the nodes.

We selected subsets of the data to build a mobility scenario that is usable on VMT. The data set contains 241 devices, 41 of which were Haggle nodes carried by participants. To work within the limitations of the testbed [11], a random subset
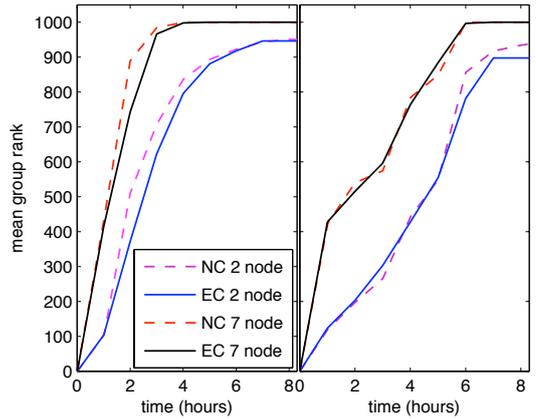
of 18 nodes was selected to generate a scenario. We selected a time interval of approximately 30,000 seconds (8.33 hours) to capture the encounter activity of a particular day.

Nodes	7, 8, 11, 12, 13, 18, 20, 22, 23, 25, 26, 27, 31, 36, 37, 38, 39, 41
Source Nodes	7, 8, 20, 31
Time Interval	70,048 − 100,373

In these experiments, four nodes were selected as source nodes, each sending a 100MB bundle that is divided into 1000 100kB fragments and distributed using our SimpleNC router.

5.3.1   Analysis

We ran experiments based on Haggle traces using the router in both NC and EC modes. For each hour in the experiment, and for $n \in \{2, 7\}$ we chose 1000 random subsets of size n and averaged the group ranks. The average group rank for n nodes represents the rank a harvester node would expect to achieve after meeting n peers.

Results for bundles sent from nodes 7 and 20 are shown in Figure 7. As expected, the mean group rank increases with time. For smaller group sizes the average never reaches 1000 because some nodes never picked up any coded fragments for the bundles in question, and they are included in the average. We observe that for the bundle sent from node 7, NC consistently leads EC by 30-60 minutes, while for the bundle sent from node 20, their group ranks are similar throughout the experiment. Despite the similarity of their group ranks, we see that NC tends to do a better job, on average, of delivering complete bases to nodes, especially towards the end of the day, as shown in figure 8.

5.4   Latency Results

Figure 8 shows the results for the delivery rate and latency for the SimpleNC router in our three types of scenarios. To measure these quantities, one must designate one or more *intended destinations*. In the mobile source and mixing nodes experiment we treat the first mixing node as the destination and plot the percentage of the ten 100MB bundles collected. For the island hopping experiment we consider the whole of
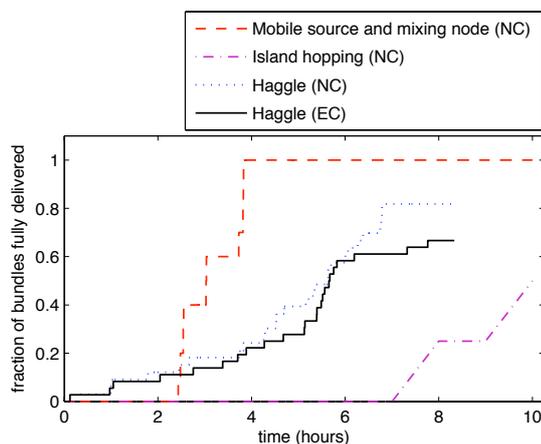
Figure 8: Delivery percentage vs time.

island 3 to be the destination, and plot the percentage of four original 50MB bundles for which the group rank of island 3 has reached 100% (any particular node in island 3 may or may not hold a full basis). For the Haggle experiments we consider all active non-source nodes to be destinations, and plot the percentage of the three 100MB bundles delivered, averaged over all destinations.

6. CONCLUSIONS AND FUTURE WORK

DTNs often involve isolated nodes and network partitions, and must rely on node mobility to carry bundles between disconnected groups. This segmentation creates bottlenecks which have an effect analogous to the founder effect on network code distribution. Conventional metrics for evaluating DTN protocols include latency and delivery ratio between a source and destination. These metrics are essential measures of network performance, however they do not give us much insight into the behavior of a code distribution scheme or the founder effect phenomenon. We have defined and experimented with metrics for quantifying the effectiveness of code distribution schemes, and have demonstrated a relationship between minimum spanning radius and latency.

Experimentally, we have found that in non-segmented two-hop DTNs, network and erasure coding perform comparably, though network coding mixing nodes seem to be somewhat more effective at distributing innovative coded fragments. In the highly segmented island-hopping scenario, we found that network coding enjoys a drastic performance advantage over erasure coding. In our Haggle-driven experiment, we found that distributing content using network coding may increase a node's chances of collecting a full basis through encounters with a certain number of peers.

In the process we have demonstrated the practical feasibility of using random linear network coding to disseminate up to 1GB of data using 5Mbps wireless links in mobile scenarios, and gained useful experience with the design issues associated with a network coding DTN router.

Our current router development paves the way for an abundance of possibilities for future research, including exploring directed routing based on network coding, ensuring fairness when handling multiple bundles, and using code diversity as a feedback mechanism.

7. REFERENCES

[1] A. Caro and J. Zinky. DTN erasure coding extension block (preliminary notes). To appear as Internet Draft.

[2] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[3] R. Beraldi and H. Alnuweiri. Delay tolerant podcasting with network coding. In *Proceedings of IFIP Wireless Days*, 2010.

[4] N. Cai and R. Yeung. Secure network coding. In *Proceedings of the 2002 IEEE International Symposium on Information Theory*, 2002.

[5] P. Chou, Y. Wu, and K. Jain. Practical network coding. In *Proceedings of Allerton Conference on Communication, Control, and Computing*, 2003.

[6] M. C. Chuah, P. Yang, and Y. Xi. How mobility models affect the design of network coding schemes for disruption tolerant networks. In *ICDCS Workshops*, 2009.

[7] Delay Tolerant Networking Research Group. DTN bundle protocol reference implementation. http://www.dtnrg.org/wiki/Code.

[8] K. R. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of ACM SIGCOMM*, 2003.

[9] C. Gkantsidis and P. R. Rodriguez. Cooperative security for network coding file distribution. 2006.

[10] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *Proceedings of 2003 IEEE International Symposium on Information Theory*, 2003.

[11] Y. Kim, K. Taylor, C. Dunbar, B. Walker, and P. Mundur. Reality vs emulation: Running real mobility traces on a mobile wireless testbed. In *HotPlanet 2011*, 2011.

[12] Y. Lin, B. Li, and B. Liang. Stochastic analysis of network coding in epidemic routing. *IEEE JSAC*, 26(5):794–808, 2008.

[13] M4RI(e). http://m4ri.sagemath.org/.

[14] J. Ott and D. Kutscher. A disconnection-tolerant transport for drive-thru internet environments. In *Proceedings of IEEE Infocom*, 2005.

[15] J. Ryu, S. Bhadra, and S. Shakkottai. TCP with feed-forward source coding for wireless downlink networks. In *Proceedings of Allerton Conference*, 2009.

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

[17] K. Scott and S. Burleigh. Bundle Protocol Specification. RFC 5050 (Experimental), Nov. 2007.

[18] J. K. Sundararajan, D. Shah, M. Medard, M. Mitzenmacher, and J. Barros. Network coding meets TCP. In *In Proceedings of IEEE Infocom*, 2009.

[19] N. Thompson, R. Crepaldi, and R. Kravets. Locus: A location-based data overlay for disruption-tolerant networks. In *Proceedings of ACM CHANTS*, 2006.

[20] M. Yang and Y. Yang. Peer-to-peer file sharing based on network coding. In *Proceedings of IEEE ICDCS*, 2008.