# Internet Service in Developing Regions Through Network Coding

Mike P. Wittie, Kevin C. Almeroth, Elizabeth M. Belding
Department of Computer Science
UC Santa Barbara
Santa Barbara, CA 93106
{mwittie, almeroth, ebelding}@cs.ucsb.edu

Ivica Rimac, Volker Hilt
Bell Labs
Alcatel-Lucent
Holmdel, NJ 07733
{rimac, hilt}@bell-labs.com

***Abstract*—The availability of Internet services brings many benefits to developing regions, yet Internet deployment levels in these regions remain staggeringly low. In this work we investigate how existing cellular deployments, which have enjoyed more rapid and wider deployment than client Internet infrastructure, could be used to provide very low cost Internet services in under-developed rural areas. We propose a new service model in which traffic is delivered over multihop client-to-client connections that are coordinated by end-to-end control traffic exchanged over cellular infrastructure. To enable this scheme in low client density rural settings, we propose a novel data forwarding mechanism for opportunistic space-time paths. To explore multiple opportunistic paths, but without the high forwarding cost of replicating data on these paths, we use network coding and send only a fraction of the data on each path. Through extensive OPNET simulations we show that globally coordinated opportunistic forwarding enables service acceptable to most applications at only a fraction of cellular infrastructure load. We argue that the reduced load on the cellular infrastructure allows additional users to share services and cost of the network and has the potential to lower the per user price of data services in developing regions.**

## I. Introduction

The availability of Internet services brings many benefits to developing regions, yet Internet deployment levels in these regions remain staggeringly low [1], [2]. Existing Internet deployment efforts are isolated and serve relatively small communities [3]. On the other hand, cellular infrastructure in developing regions already serves much larger areas and continues to expand at a rapid pace [4]. However, most of these cellular deployments do not offer data services, and, while there are plans to provide Internet in Africa using Third Generation (3G) technologies, the costs of spectrum licenses and cellular equipment make this option viable only for a relatively small percentage of urban populations [5].

Studies of Internet usage in developing regions conclude that most data traffic is local to the community [6], [7]. These communications are centered around multimedia file exchange, requests of cached content like Wikipedia, or business and medical record traffic. These types of traffic can be supported by multihop client-to-client connections and infrastructure data transmissions can be used only for data not available locally. We posit that very low cost data services in developing regions can be easily deployed by leveraging existing cellular infrastructure to coordinate multihop data traffic between clients.

For some time now, researchers have considered augmenting cellular networks with client-to-client Wi-Fi connections [5], [8], [9]. In addition to cell tower to client communications, data in these these networks, dubbed Multihop Cellular Networks (MCNs), is forwarded on ad-hoc routes between the clients themselves using Wi-Fi radios already available on many handsets. The additional multihop connectivity can be used to increase a cell's capacity or improve service coverage.

In the developed world, the capacity of cellular networks is improved by the deployment of additional cellular towers to increase spatial reuse of the wireless spectrum. Figure 1 illustrates an MCN that can be constructed over such a dense cellular deployment. On the other hand, cellular deployments in developing regions are characterized by far lower user density and therefore require larger cell size to service enough users to make deployments viable. Figure 2 illustrates a sparse MCN possible in such larger cells.

There are two major differences between dense and sparse MCNs. First, the low client-to-tower ratio of dense MCNs means that, in most cases, clients can connect to the Internet directly through the infrastructure. While MCN solutions can improve the quality of the data service, motivating user cooperation adequately remains a challenge. On the other hand, in sparse MCNs communicating with distant users in the larger cells requires additional code space and so network capacity remains a scarce resource in spite of lower user density. Deployment of additional towers may not be financially viable, and to enable reliable service at all, some of the traffic may need to be exchanged between clients.

Second difference, users in a dense MCN can be expected to be close enough to each other to make possible the establishment of ad-hoc routes. On the other hand, in sparse MCNs a contemporaneous routing path may not exist and non-infrastructure client-to-client communications may only be possible over opportunistic time-space paths.

The delivery of data in spite of a lack of contemporaneous end-to-end paths has been the focus of research in Delay Tolerant Networks (DTNs). To effect delivery, aggregated data is flooded across opportunistic connections until some copy reaches the destination. Many solutions have been proposed

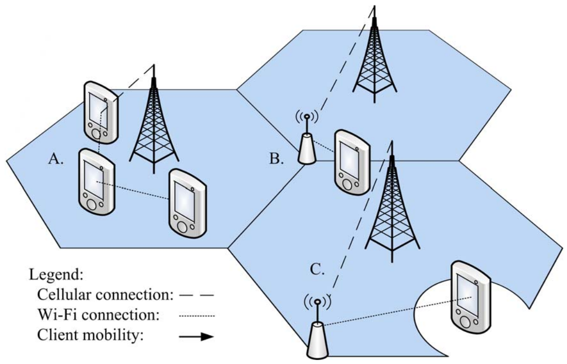


Fig. 1. Communications in dense MCN: A. multihop client relay; B. cell capacity sharing through ad-hoc relaying station; C. cell coverage extension.

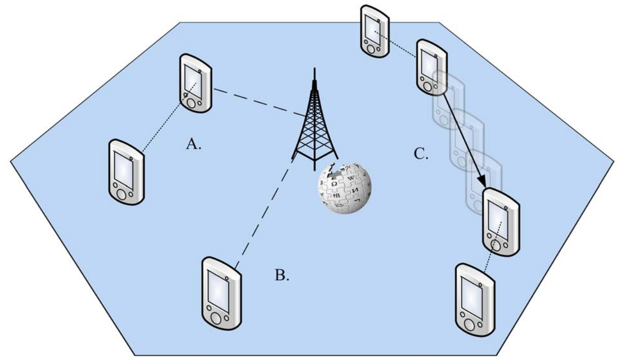


Fig. 2. Communications in sparse MCN; A. multihop client relay; B. localized content access; C. opportunistic client relay.

to reduce the scope of bundle floods, but because mobility is hard to predict in DTNs, data replication onto ultimately superfluous paths increases the cost of forwarding [10], [11].

The client cooperation solutions proposed for MCNs suffer from a lack of adequate incentive schemes to make such cooperation attractive. Mobile clients are expected to buffer and forward traffic for one another in order to reduce the number of data transmissions over the cellular infrastructure. While client cooperation can improve service in an overloaded cell area and decrease deployment cost by allowing more clients to share a cell area, these benefits come at the cost of lower handset battery lifetime. Clients are reluctant to share their power-hungry Wi-Fi radios when the primary benefit is reduced infrastructure load. We believe that a more compelling argument for client cooperation is the creation of data service availability where no such service currently exists.

To create sparse MCNs suitable for developing regions that conserve hand-held resources, a new routing solution is required. We aim to reconcile the need for path redundancy in sparse, opportunistic networks, but without resolving to the high forwarding cost of redundancy by replication. We propose to forward a single copy of the data and spread that data over different paths with the use of *network coding*.

Network coding encompasses a number of techniques of encoding $n$-partitioned data into at least $n + k, k \ll n$ coded pieces or linearly independent combinations of the partitions. Using network coding, the original data can be decoded from any $n + k$ coded pieces with high probability of success. Network coding allows data on multiple paths to converge such that each successive coded piece increases the amount of decoded data. While network coding has already been used in a number of network scenarios [12], [13], [14], our proposed technique is unique in that it uses an end-to-end control channel to manage the coding and forwarding mechanisms.

To demonstrate the benefits of our proposed scheme, we evaluate a fully distributed implementation with extensive OPNET simulations. We show that the combination of network coding and opportunistic traffic management using an end-to-end control channel is an effective technique for achieving path redundancy that avoids the cost of forwarding multiple copies of the same data. We also show that our proposed schemes reduces cellular infrastructure load, which has the potential to lower the cost of data services in developing regions.

The remainder of this paper is organized as follows. In Section II we discuss related work in the areas of MCNs, DTN routing and network coding. Our proposed solution is introduced in Section III. In Section IV we evaluate our work and present the results. Finally, we conclude in Section V.

## II. Background and Related Work

The work presented in this paper spans research in MCNs, DTNs, and network coding. We aim to build on recent developments in these areas and propose novel techniques that address specific challenges in each area. To motivate our effort, we proceed with a summary of related work as well as existing challenges in the way of sparse MCNs.

### A. Multihop Cellular Networks

Introduced by Lin and Hsu, MCNs leverage cellular infrastructure and multihop ad-hoc paths between clients to deliver a unified network service [8]. MCNs can improve a cell's throughput or increase coverage. Existing techniques have focused primarily on the dense MCN scenario presented in Figure 1.

Figure 1-A shows the Unified Cellular and Ad-Hoc Network (UCAN) architecture proposed by Luo *et al.* [9]. UCAN eliminates transmissions to clients with poor tower connectivity and instead transmits to well-connected clients, which then deliver the data to the ultimate destination over high bandwidth Wi-Fi paths.

MCNs can also be used to extend network connectivity beyond the range of a cell. Wu *et al.* proposed to use strategically deployed Ad-Hoc Relaying Stations (ARSs) to use spare capacity of adjacent cells (Figure 1-B) and to extend connectivity to holes in cellular coverage (Figure 1-C) [15].

Finally, Manoj *et al.* use the cellular infrastructure not only to create an MCN, but also to improve ad-hoc routing performance. The authors proposed Base-Assisted Ad-Hoc Routing (BAAR), which reduces routing overhead by aggregating link quality measurements and calculating routes at base stations [5]. While the routing paths computed using cell-wide link state eliminate much of route discovery overhead, they suffer under high mobility scenarios where the precomputed route is no longer viable.

While these schemes realize MCNs and have the potential to improve client connectivity, their treatment of ad-hoc routing does not fully take advantage of the global knowledge provided by cellular control channels. With the use of global network information, multihop routing could take advantage of opportunistic paths as well. Opportunistic routing allows for multihop routing in sparse MCNs, where contemporaneous end-to-end paths may not exist.

### B. Delay Tolerant Networks

DTNs are networks in which a contemporaneous forwarding path between two nodes may not exist. Instead, delivery is made by grouping application data into *data bundles* that are exchanged during opportunistic contacts in hope of eventually reaching the destination. The DTN network architecture was introduced by Vahdat and Becker and later standardized by Cerf *et al.* [16], [17].

Vahdat and Becker proposed Epidemic Routing as a first stab a routing in DTNs [16]. Their protocol floods data during opportunistic contacts in hope that one of the bundle copies ultimately reaches the destination. In spite of achieving high delivery rate and low delay, Epidemic Routing exhibits high cost in terms of total number of bundle transmissions, which can cause network congestion when multiple flows are present.

Subsequent advances in DTN routing were aimed at reducing the cost of forwarding by limiting data hop count or node forwarding willingness, forwarding only to nodes better suited to carry traffic for a particular destination based on similarity of mobility patterns between the potential forwarder and the destination, and using explicit knowledge about node mobility and topology to schedule data forwarding with respect to expected contacts [10].

A novel approach to DTNs is the use of a thin end-to-end connections to aid opportunistic connectivity. We have previously proposed ParaNets, a network architecture where an end-to-end control channel and an opportunistic channel exist side by side [18]. The control channel may not be suitable for data communications due to bandwidth or cost limitations, but can exchange bundle delivery acknowledgments much more quickly than opportunistic propagation.

We take the idea of using a thin control channel further in Cloud Routing (CR) [11]. CR uses the control channel to disseminate node position information and bundle progress notifications. Using this information, CR forwards with highest priority only a small set, or *cloud*, of data copies that are closest to the destination with respect to other copies of the same flow. A major contribution of CR is that it explicitly manages congestion in DTNs with respect to global network state. The use of parallel channels in DTNs has been recently explored to some extent by other works as well [19], [20].

While CR greatly improves the efficiency of the opportunistic channel, it is not immediately suited for adaptation in developing regions. In an uncongested network, CR allows the traffic to be transmitted during all available contacts, which leads to high delivery cost and large buffer requirements. In rural settings, one needs to be judicious with the use of client battery power and minimize the number of cooperative Wi-Fi transmissions. To explore multiple paths, but without the cost of data replication, we turn to network coding.

### C. Network Coding

Network coding allows networks to increase their throughput by transmitting useful data on links that cannot be utilized by non-coded traffic for a given network topology. Pioneered by Ahlswede *et al.* for multicast applications, the technique was simplified Koetter and Médard, who showed linear codes were sufficient to reach network broadcast capacity [21], [22]. Subsequently, Ho *et al.* showed that random linear codes were sufficient to decode data at any given receiver with probability $1 - \delta$ for encoding field size $|E|/\delta$, where $E$ is the number of edges in the network [23]. Katti *et al.* applied these techniques in COPE, which performs *inter-flow* coding of packets overheard in neighboring transmissions to improve network throughput [12].

Chou *et al.* adapted network coding to networks characterized by random loss and delay [24]. The authors' technique concatenates portions of encoded data and corresponding codes inside *coded pieces*. New coded pieces are created at intermediate forwarding nodes from random linear combinations of previously received data. With high probability, the coded pieces arriving on multiple paths are linearly independent, or *innovative*. When enough innovative coded pieces are received, it is possible to reconstruct the global encoding from codes embedded in the coded pieces themselves and decode the original data. The network coding technique used in this paper is based on work by Chou *et al.* and we explain the technique and its implementation in more detail in Section III-A.

The technique of Chou *et al.* allows *intra-flow* coding, where coded pieces that belong to the same flow, but traverse different paths, are encoded together. In MORE, Chachulski *et al.* partition a continuous data flow into *generations*, which are then encoded together as they traverse a set of routing paths [13]. To reduce the cost of forwarding, CodeOR, by Lin *et al.*, allows for multiple generations to be in-flight simultaneously [14]. While MORE and CodeOR exploit multiple paths and increase network throughput, such end-to-end paths may not exist in sparse MCNs. Moreover, MORE and CodeOR cease forwarding when 'enough' data has been sent, which can be hard to determine a DTN, where data progress may be erratic and unreliable.

Network coding of unicast flows in DTNs has been investigated by Widmer and Le Boudec [25]. Their solution, Network Coding Probabilistic Routing (NCPR), forwards coded pieces with a network wide forwarding factor, $d$. Each node attempts to forward up to $\lfloor d \rfloor$ coded pieces and an additional piece with probability $d - \lfloor d \rfloor$. A node stops forwarding once $\lceil d \rceil$ coded pieces have been forwarded. Newly received innovative coded pieces reset the forwarded coded piece counter at the sender. The probabilistic forwarding factor, $d$, is configured with respect to node density. The fundamental tradeoff is that high values of $d$ result in high delivery rates but also high network load.

Indeed, this tradeoff is where MORE, CodeOR, and NCPR come up short. A large number of redundant pieces is costly to forward, while a small number may not provide enough redundancy and the destination may not be able to decode the data. Even if the number of redundant pieces is dynamically adjusted with respect to network conditions, generating all the redundancy at the source results in a higher end-to-end transmission cost.

Instead, we propose to forward data with a constant amount of redundancy and to construct the encoding such that lost information can be reconstructed from the coded pieces remaining in the network. Our mechanism can be configured to tolerate a specific number of simultaneous node failures, but an unlimited number of total failures as long as the global encoding can be regenerated quickly enough with respect to failure rate.

## III. SEMI-INNOVATIVE SET ROUTING

To enable Internet services in sparse MCNs, we propose Semi-Innovative Set Routing (SISR), a novel network coding forwarding mechanism. SISR reduces the cost of opportunistic forwarding to conserve hand-held resources and increase potential network throughput. Similarly to previous solutions, SISR avoids the cost of data replication and sends only a fraction of coded data on each path. SISR is novel in that it controls the global data encoding such that every forwarding coded piece of data is innovative, and in that SISR assures resiliency to losses by regenerating encoding redundancy during the forwarding process. Ultimately, we hope SISR enables data services in sparse MCNs, where data is forwarded during opportunistic client contacts and the cellular network disseminates a much smaller volume of control traffic.

### A. Network Coding Model

The network coding technique used in SISR is based on work by Chou et al. [24]. To facilitate the presentation of the coded flow management proposed in this paper, we briefly describe our adaptation of their scheme. For an overview of network coding techniques and applications see an excellent primer by Fragouli *et al.* [26].

Coding operations are performed using symbols defined by polynomials from some finite field $GF$ that is large enough with respect to the number of edges in the network. A Galois field $GF(8)$ has been shown to be large enough for most applications and $GF(16)$ sufficiently large that coding errors become negligible with respect to other sources of packet loss [24].

Assume a symbol size of 8 bits represents the finite field polynomial. Transmission data composed of $n$ bytes, or symbols is divided into an $n/p \times p$ data matrix $D$ where $p+n/p$ is the number of bytes that can be carried by a network packet. At the source node, the data matrix is prepended by an $p \times p$ encoding matrix $E$, initially the identity matrix.

To generate a *coded piece* the source node creates an *encoding vector* $\mathbf{v}$ of $p$ random symbols from $GF$. The coded piece is the concatenation of vectors obtained by $\mathbf{v}E$ and $\mathbf{v}D$ operations performed in $GF$. Coded pieces received at intermediate nodes, or the destination, build up the local $E$ and $D$ matrices row by row. When the prepended $E$ can be inverted by Gaussian elimination, $E$ becomes again the identity matrix and the associated $D$ is reduced to the original form at the source node and its decoded data can be read directly.

To generate coded pieces at intermediate nodes, a node creates a new $\mathbf{v}$ to encode the $E$ and $D$ matrices using the rows present at the node. The resulting coded piece is a random linear combination of rows present and if $E$ has full rank the coded piece is linearly independent from coded pieces generated at other nodes with high probability [26]. Thus coded pieces generated at the source node or intermediate nodes are linearly independent, and each such received coded piece is expected to increase the rank of the local encoding matrix.

### B. Bundle Code Management

In sparse MCNs, nodes may be too far apart to form contemporaneous routing paths, which necessitates opportunistic forwarding. Because forwarding multiple copies of the same data is wasteful of network resources, we propose to send only a fraction of data on each path and use network coding to assure that all data reaching the destination is innovative.

In a sparse network, all space-time forwarding paths may not reach the destination in a timely manner. It therefore becomes necessary to send more coded data than strictly required for decoding. Nodes carrying a portion of the data may be lost due to churn, or simply fail to come in contact with the destination as predicted by the routing metric. To avoid the long delay and additional cost of sending replacement data from the source, redundancy should be included in the original transmission. However, losses in the network could be more numerous than expected and redundancy built into the flow might no longer guarantee delivery. Conversely, transmitting too much redundant data increases the cost of forwarding and wastes network resources. In sparse MCNs, where it is important to minimize the cost of cooperative forwarding, we propose to recover redundant data from the remaining innovative encoding right in the network.

Assume a number of coded pieces $b = |\mathbf{v}|$ that together constitute the amount of data minimally required to decode the original bundle. Also forwarded are $r = \frac{b}{4}$ coded pieces of redundant data. This scenario is illustrated in Figure 3-A. The forwarded data is spread across multiple nodes such that no more than a *bundle fraction* $f = r$ is carried by each node.

When some node leaves the network, up to $f$ coded pieces are lost from the global encoding. The remaining $b$ coded pieces are linearly independent with high probability if every coded piece is a random linear combination of the bundle data. The lost redundant data can be regenerated using new random encoding of the remaining linearly independent coded pieces.

The solution to implementing redundancy regeneration in practice is to ensure that any coded pieces that make up $b$ out of $b+r$ of coded data are linearly independent. Linear independence can only be checked up to the full rank of the encoding

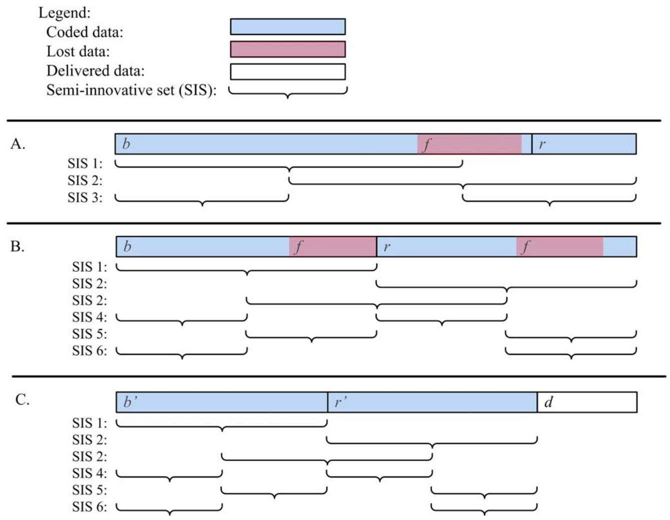


Fig. 3. Semi-innovative set examples.

matrix, however checking linear independence for every subset of coded pieces of size $b$ is computationally impractical.

SISR solves this problem with the use of *semi-innovative sets* (SISs). We define an SIS as a set of coded pieces whose combined rank equals the size of the set. Notice that it is not required for an SIS to have full rank. Given the SIS definition we offer the following theorem:

*Theorem 1:* Given a set of coded pieces $C, |C| = b+r, b = |\mathbf{v}|, r \geq 1$, $\binom{\lceil \frac{2|C|}{b} \rceil}{2}$ SISs can be constructed over $C$, such that any $c \subset C, |c| = b$ has full rank if every SIS has rank $2\frac{|C|}{\lceil 2\frac{|C|}{b} \rceil}$.

Theorem 1 implies that a successful construction of SISs over $C$ assures the full rank of any subset of size $b$. Because the loss of redundant data does not invalidate the global encoding, SISR can use the remaining data to recreate the redundancy.

Figure 3-A shows a construction of three SISs that satisfy Theorem 1. Coded pieces in $C$ are equally divided among $\lceil \frac{2|C|}{b} \rceil$ disjoint subsets $s_1, s_2, ..., s_{\lceil 2|C|/b \rceil}$. SISs are constructed as every union $s_i \cup s_j \mid \forall i, j \in [1, \lceil \frac{2|C|}{b} \rceil], i \neq j$. Because rank of each SIS has to be less than or equal to full rank and the size of each SIS equal to its rank, the size of each $s_i$ has to be less than or equal to $\frac{b}{2}$. In turn the size of each $s_i$ dictates the number of $s_i$'s as $\lceil \frac{|C|}{b/2} \rceil = \lceil \frac{2|C|}{b} \rceil$, the number of SISs as $\binom{\lceil \frac{2|C|}{b} \rceil}{2}$, and the rank of each SIS as $2\frac{|C|}{\lceil 2\frac{|C|}{b} \rceil}$.

We briefly justify the correctness of Theorem 1 as follows. Assume that Theorem 1 initially holds for $C$ in Figure 3-A. When a set of coded pieces of size $f$ is lost from the global encoding, the construction of SISs ensures that the remaining data has full rank. SIS1 contains linearly independent coded pieces by the definition of semi-innovative sets. The remaining data in SIS2-SIS1 is linearly independent with respect to SIS1-SIS2, because it is linearly independent with respect to SIS1∩SIS3, by the construction of SIS3. Similarly, the remaining data in SIS2-SIS1 is linearly independent with respect to SIS1-SIS3, by the construction of SIS2.

SISs can be constructed to tolerate any number of losses $l = \lfloor r/f \rfloor, f \leq r$. For example Figure 3-B shows the construction of six SISs for $|C| = 2b$ and $l = 3$. Increasing a flow's resiliency to losses requires that either more redundant data be transmitted, a larger $r$, or that the data be more finely spread across the nodes, a smaller $f$. While the number of SISs grows exponentially as $\binom{\lceil 2(|\mathbf{v}|+r)/|\mathbf{v}| \rceil}{2}$, the SISR forwarding mechanism requires that each node maintain only $\lceil 2(|\mathbf{v}|+r)/b \rceil - 1$ SISs and so the complexity is linear in practice.

The implementation of SISR requires global knowledge of traffic and encoding data. SISR disseminates this information over the end-to-end control channel instantiated by the MCN cellular network. Upon receiving data opportunistically, each node sends on the control channel the encoding vector for its linearly independent encoding rows. Also on the control channel, each node periodically announces its position information. From these two pieces of information, each forwarding node can calculate the amount of linearly independent data located closer to the destination and decide whether forwarding its data can aid in bundle delivery. In this paper we treat distance to destination as the routing metric, though we have shown in previous work that the control channel can be used for other routing metrics as well [11].

To calculate the amount of innovative data in the network closer to the destination, each node only needs to calculate the amount of data in its SISs that is already closer to the destination. Since each node carries up to $f$ coded pieces, we can assign each node's fraction of bundle data to a specific part of the encoding, $s_i$, where $i$ is obtained from a uniform hash of node id. Because a node's data resides in only one $s_i$, the innovative state of the node's data only needs to be checked against $\lceil 2(|\mathbf{v}| + r)/b \rceil - 1$ SISs that include $s_i$. A forwarding node iterates through nodes closer in the network and adds innovative codes to appropriate SISs.

Data at nodes that hash to the same $s_i$ as the forwarding node is counted towards all SISs maintained by the forwarding node – if the hash values are different, the nodes will only have one SIS in common. For example in Figure 3-B, if the forwarding node's hash places its data in the left half of SIS1 and a closer node's encoding position is the right half of SIS2, then the data at the closer node overlaps only in SIS6. Thus if the forwarding node and the closer node carry the same codes, the forwarding node will have no innovative contribution toward SIS6. Based on the closer contribution in each SIS, the forwarding node creates new coded pieces from random linear combinations of its innovative data. Because these coded pieces are innovative with respect to the global encoding closer to the destination, data forwarded at multiple nodes can rebuild the lost redundancy as long as forwarding rate exceeds the loss rate.

To conserve buffer resources and limit the number of flows kept track of by every node, coded pieces are deleted when they are no longer innovative with respect to network data closer to the destination. If the node has no more innovative data for any of its SISs, either because data progressed toward, or the node has turned away from the destination, then the node drops the locally held data and announces this to the network.

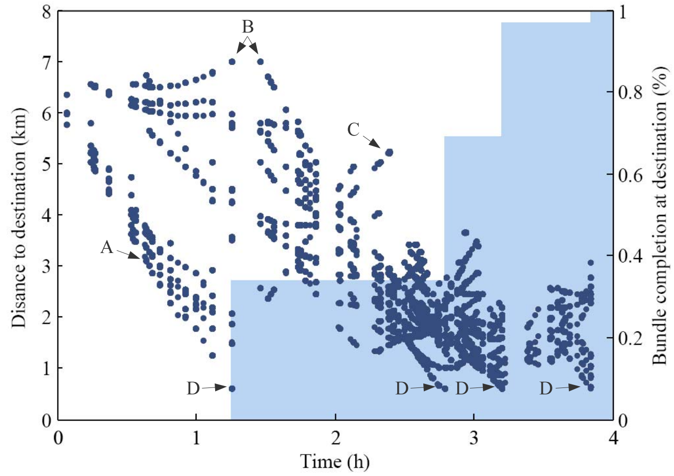


Fig. 4. Progress of coded data in the network (points) and of data decoding at the destination (solid area).

Another reason a node may no longer have innovative data is if that data it carries has been delivered to the destination.

When data is delivered, the SISs at each node are recomputed to exclude the delivered data. Because the destination announces new innovative rows it receives, each node can stop forwarding the non-innovative codes. Figure 3-C shows the amount of data still needed at the destination $b' = b - d$, where $d$ is the amount of data decoded at the destination. Bundle fraction is scaled down accordingly as $f' = max(fb', \frac{1}{|\mathbf{v}|})$.

While in this paper we assume 100% coverage and reliable delivery on the control channel, we have previously shown how network state inconsistencies due to missed control messages can be restored opportunistically [11].

### C. Example of Cloud Progress

To further clarify the exposition of SISR, we present a brief example of a bundle's progress in Figure 4. The horizontal axis marks simulation time, the left vertical axis marks distance to destination of coded pieces carried by individual nodes, while the right vertical axis the amount of data decoded at the destination. Each of the points represents the distance from the destination of a node that is part of some SIS, or in other words, contains some amount of innovative data for the destination recorded during progress notification processing.

As time passes, data is forwarded during opportunistic contacts and makes progress towards the destination, as in A. Nodes that no longer approach the destination either hand off their data to better forwarders, as in B, or delete their content, as in C, when they no longer contain data innovative for any SIS. As nodes come to within the communication distance of the destination, as in D, data received at the destination increases the amount of data that can be decoded. Finally, the destination decodes the entire bundle at time 3:50, and sends a notification instructing all cloud members to drop their data.

## IV. Evaluation

Our evaluation goals are twofold. First we aim to investigate the performance advantage of SISR in a sparse MCN, as compared to the previously proposed solutions of Network Coding Probabilistic Routing (NCPR), which floods network coded data, and Cloud Routing (CR), which replicates whole bundles onto a small number of paths. Second, we aim to show the effects of parameter settings on SISR behavior and identify areas for future research. We begin by detailing our experimental setup.

TABLE I
Simulation configuration.

| | Parameter | Value |
|---|---|---|
| Network | Network area | 8 by 8 km |
| | Network size | 100 nodes |
| | Transmission range | 600 meters |
| | PHY data rate | 54 Mbps |
| | Node connection protocol | TCP Reno |
| | Simulation time | 8 h |
| | Number of random seeds | 40 |
| Traffic Model | Bundle range | 1 - 6 km |
| | Bundle size | 1 MB |
| | Bundle time-to-live | Unlimited |
| | Node buffer size | Unlimited |
| Mobility | Mobility model | Random Mobility |
| | Speed | 1 - 1.5 m/s |
| | Pause time | 20 s |
| Network coding | Symbol size | 8 bits |
| | Encoding vector length | 16 symbols |
| Routing | Neighbor beacon interval | 1 s |
| | Position update interval | 60 s |
| | Bundle buffer size | Unlimited |
| | NCPR forwarding factor ($d$) | 0.625 (10/16 rows) |
| | CR cloud size | 3 bundles |
| | SISR cloud size | 2 bundles |
| | SISR node bundle fraction ($f$) | 0.33 |

### A. Experimental Setup

To understand the ability of each routing solution to forward flows through the network, we focus our evaluation on the performance of a single flow between fixed source and destination nodes. We configure the opportunistic forwarding environment between these two nodes to reflect the characteristics of sparse MCNs. Table I details our environment setup.

We configure the simulated network to represent the area a typical rural community of around a hundred users might occupy. In previous studies, commodity Wi-Fi hardware with basic omni-directional antennas has been shown to maintain links of up to 800 meters in rural settings [6]. We expect the communication devices participating in sparse MCNs to have integrated antennas and so we restrict the range of each node to 600 meters. While the size of the community, user density, and communication ranges might vary between deployments, we believe that our setting accurately represents a sparse scenario. Users are able to communicate with one or two neighbors at any given time, but are not able to establish contemporaneous routing paths with every other node. Thus end-to-end connectivity requires opportunistic forwarding.

To simulate the number and duration of opportunistic contacts in rural areas, we model node mobility with Random Waypoint Mobility configured to walking speeds. We believe that random mobility is a good approximation for the opportunistic forwarding substrate.

The flows themselves are one megabyte bundles and represent both small multimedia files as well as aggregations of email or other data. One megabyte bundles are small with

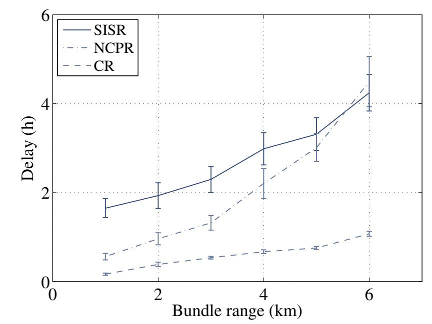


Fig. 5. Bundle delivery delay vs. bundle range.

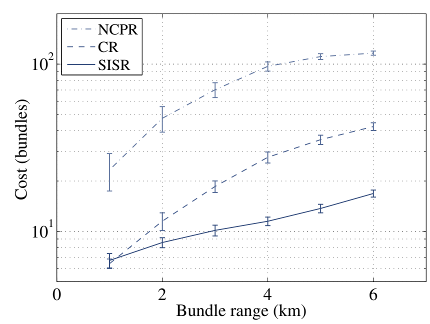


Fig. 6. Bundle delivery cost vs. bundle range.

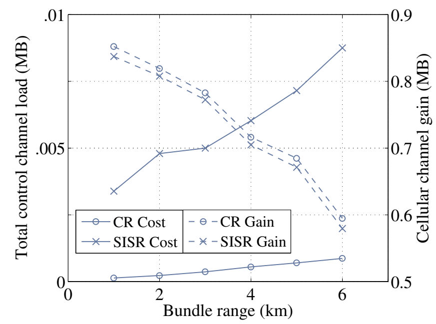


Fig. 7. Control traffic and gain vs. bundle range.

respect to opportunistic contact throughput, which allows them to be forwarded in their entirety by the CR mechanism. The network coding solutions partition bundles into coded pieces and are able to make forward progress even when an entire bundle cannot be transmitted during a short node contact.

Data is transferred between nodes using the IEEE 802.11g standard at data rates up to 54 Mbps. We use TCP Reno to accurately simulate opportunistic connection throughput in the face of medium contention and disconnections. We simulate the transmission of each flow for up to eight hours. While in practice the delivery times are much shorter, longer delays might still be acceptable for some types of data, like email or multimedia files.

Finally, we configure the routing solutions as follows. The NCPR forwarding factor $d$ is configured to 0.625, or 10 out of 16 rows of the encoding matrix. The cost of NCPR increases at larger forwarding factor values; the value we selected represents the lowest forwarding cost that achieves 100% delivery at 6 km bundle range for all forty simulation seeds.

We configure the CR cloud size to three bundles. Forwarding priority of flows in CR is determined by distance to destination. In multiflow scenarios, copies of long-range bundles can becomes ‘stuck’ behind short-range traffic [11]. Forwarding three to six copies simultaneously increases the probability of forward progress, reduces delay, and increases network throughput, as delivered bundles are removed from node buffers and allow for forwarding of other traffic. We configure CR cloud size to three bundles to minimize forwarding cost and retain the low forwarding delay of larger cloud sizes.

Similarly, we configure SISR to its lowest delay at 6 km with cloud size of two and bundle fraction of one third. Bundle fraction of one third allows a bundle’s worth of innovative data forwarded with SISR to explore the same number of paths as CR. SISR cloud size of two is needed to achieve 100% delivery success rate for all forty simulation seeds. While our configuration of SISR actively forwards one less bundle than CR, as we explain in the next section, the differences in forwarding cost are due to specificities of the mechanisms and not their configuration.

### B. Results

*1) End-to-End Delay:* End-to-end delay is recorded as the time between bundle creation at source and the reception of the entire bundle at the destination. Because CR forwards bundles in their entirety, the end-to-end delay is recorded at the arrival of the first bundle copy. For the network coding solutions, we record the delay when enough innovative coded pieces have been received to decode the entire bundle.

Figure 5 shows the comparative end-to-end delay and 90% confidence intervals of the SISR, CR, and NCPR solutions. The horizontal axis represents bundle range, or the distance between source and destination nodes. The vertical axis marks the end-to-end bundle delay. In general the end-to-end delay for all solutions increases with bundle range.

The SISR solution has the highest delay at the lower bundle ranges, but not the highest rate of delay increase with respect to bundle range. Because only a fraction of the data is carried by each node, the rare contacts with the destination result in delivery of only a fraction of bundle data. To decode the entire bundle, the destination node needs to wait for coded pieces carried by multiple nodes. We explore the relationship between bundle fraction and delay more fully later in this section. The delay of SISR is acceptable for applications like email or multimedia exchange. Applications that require quicker round trip times can resort to transmitting their data directly over the more costly cellular infrastructure.

NCPR also exhibits long end-to-end delays, especially for long range bundles. As bundle data progresses through the network, nodes use up their forwarding allowance, $d$, and nodes farther from the source receive decreasing amounts of data from their predecessors. As a result, NCPR destination nodes need to wait for data arriving on multiple paths before an entire bundle can be decoded, as was the case in SISR. This effect is exacerbated at longer range and results in a higher rate of increase in delay for NCPR than SISR.

Finally, CR achieves the lowest delay. As soon as the first bundle copy reaches the destination, the entire bundle is delivered. The delay of CR is the delay of the first path connecting with the destination, or best explored path. In contrast, the delay for SISR is the delay of the path that delivers the last innovative coded piece. While this may indicate the routing a single copy in CR is sufficient, the relative proximity to destination of multiple copies can change during forwarding and so predicting which copy would arrive first is difficult.

*2) Opportunistic Forwarding Cost:* To quantify the efficiency of each forwarding scheme we analyze the forwarding cost, counted as the total amount of data received by nodes in the network until the entire bundle is delivered at the destination. Lower bundle forwarding cost allows more flows to coexist, and so, translates to higher potential network throughput.

For CR, forwarding cost is incremented by one whenever a node receives a copy of the bundle, while for network coding solutions cost is increased by the size of each received coded piece, including the encoding vector and data. Coded pieces are counted toward the forwarding cost even if they are not innovative with respect to data at the receiving node.

Figure 6 shows the comparative forwarding cost and 90% confidence intervals of SISR, CR, and NCPR. The horizontal axis represents bundle range, while the vertical axis the forwarding cost. Longer range flows require that bundles traverse more opportunistic hops, and so in general the forwarding cost for all solutions increases with bundle range.

The progress of data in NCPR is limited to coded pieces innovative for the next hop, but not necessarily with respect to data closer to the destination. As a result, longer delivery times of long range bundles increase opportunities for transmissions far from the destination and drive up NCPR forwarding cost.

At longer ranges the cost of NCPR exceeds one hundred, or the number of nodes in the network. This can seem counterintuitive. Even if every node in the network received the entire copy of the bundle, the cost would be no more than one hundred. In practice, nodes with two neighbors can request the same data simultaneously from both. Because data at two forwarding nodes may not be independent, for example if one received its data from the other, the data arriving at the requesting node is not independent, but is counted twice towards the forwarding cost. To achieve delivery at ranges similar to network size, NCPR effectively spreads data to every other node in the network, and together with the non-innovative receptions the cost exceeds one hundred bundle copies.

The CR solution achieves the next lowest cost. CR explores three path alternatives with three bundle copies, at almost three times the data than strictly needed for delivery. While data replication is an expensive mechanism for exploring multiple paths, the low delay of CR allows it to retire bundles early and thus actively forward data only for a short amount of time.

Finally, SISR achieves the lowest forwarding cost. Multiple paths are explored by sending only a fraction of the data on each path. As data progresses, some coded pieces are no longer innovative with respect to data closer to the destination and are no longer forwarded. Each fraction of delivered data reduces the amount of data forwarded by each SIS, so the delivery of $\frac{b}{x}$ of data allows $\frac{2b}{x}$ to be discarded from the network. SISR forwards only data innovative with respect to the destination and achieves a factor of two improvement for bundle ranges over 4 km. We expect the low forwarding cost of SISR to promote higher network throughput when multiple flows compete during limited opportunistic node contacts, a principle we previously demonstrated for CR [11].

*3) Control Channel Cost:* Control channel load is recorded as the total amount of control traffic generated by opportunistic forwarding. For CR this includes bundle progress notifications.

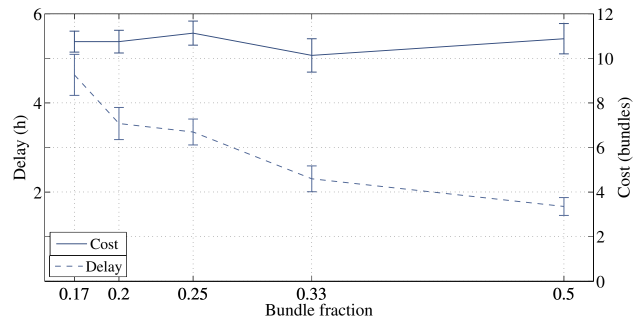


Fig. 8. SISR delay and cost vs. bundle fraction.

SISR includes data encoding vectors in progress notifications and also transmits data drop notifications when a bundle has no more contributing data for any SIS. Cellular channel gain is the amount of data saved on the cellular channel by transmitting data opportunistically instead of using the cellular infrastructure. Cellular channel gain is calculated as bundle size minus total control traffic load, which includes forwarding control traffic position updates at the 0.21 Kbps rate required for routing metric calculation of CR and SISR.

Figure 7 shows the comparative control channel load and cellular channel gain of SISR and CR. The horizontal axis represents bundle range; the left vertical axis, control channel load; and the right vertical axis, cellular channel gain.

The amount of forwarding control traffic in general increases with bundle range and is higher for SISR than CR. SISR includes data codes sent in progress notifications, which makes these messages larger. However, bundle progress notifications are infrequent, and the difference between SISR and CR routing traffic is not large in absolute terms. Because of this small difference, when combined with the constant background node position update traffic, the difference in forwarding traffic has little effect on the gain difference between SISR and CR.

The cellular channel gain is larger for CR, but SISR has a much lower opportunistic forwarding cost, which makes the solution a more attractive alternative for promoting user cooperation. The gain for SISR and CR would increase exponentially if multiple bundles shared the cost of sending node position updates. Overall, SISR is an attractive solution for reducing cellular infrastructure load, while providing equivalent throughput to the user. NCPR offers even greater savings due to lack of routing traffic, but the forwarding cost of the solution is prohibitive for cooperative user relays.

Another way of quantifying SISR control channel overhead is in terms of the number of control messages. Each node processes a position update from every node every 60 seconds. Position updates require only a simple hash insert and their processing is not computationally intensive. The forwarding messages are much more rare and their processing also requires only the storage of arriving codes.

*4) Details of SISR:* Figure 8 shows the effects of varying node bundle fraction on delivery delay and cost in SISR. The horizontal axis represents bundle range; the left vertical axis, bundle delay; and the right vertical axis, bundle cost. The results in Figure 8 represent performance measurements for an SISR 6 km bundle range.

Partitioning the same amount of data using different bundle fraction does not have a pronounced effect on forwarding cost. While larger bundle fractions achieve quicker deliveries and are able to retire data from the network earliest, these larger fractions also induce more cost during an opportunistic contact, and thus the effect of early retirement does not drastically effect the overall cost of forwarding.

The end-to-end bundle delay in Figure 8 decreases for large bundle fractions. When nodes are allowed to carry more data, fewer contacts with the destination are needed to deliver the full bundle. However, large bundle fraction sizes make the traffic cloud more susceptible to losses due to node churn or routing errors, as a loss of each node represents a larger loss to the global encoding. Larger bundle fraction also requires more redundant data to tolerate same number of losses.

Finally, we address the computational cost of the network coding in SISR. The decoding of data requires the inversion of a $16 \times 16$ encoding matrix, which is not prohibitive on modern hand-held hardware. The real complexity of SISR is the calculation of the amount of innovative data located closer to the destination. The closer data encoding is aggregated in $\lceil 2(|\mathbf{v}|+r)/b \rceil - 1$ SISs, which need to be inverted to reduce the accumulated data to innovative rows. This operation is only performed when nodes queue data for forwarding and, due to the sparsity of nodes, is relatively rare. Nevertheless we would like to evaluate the computational impact of SISR on larger traffic scenarios and real-hardware as part of future research.

## V. Conclusions

In this paper we have proposed SISR, a new routing mechanism for sparse MCNs. SISR leverages MCN infrastructure to exchange protocol control traffic to enable end-to-end management of network coding and opportunistic forwarding. Global network state knowledge allows SISR to forward only innovative data. SISR also maintains encoding redundancy that can tolerate losses on any number of nodes as long as the opportunistic forwarding rate exceeds the loss rate. These novel mechanisms allow SISR to reduce the cost of opportunistic routing by a factor of two over previous solutions. The integration of SISR in cellular deployments is expected to significanly reduce infrastructure load of data services. Reduced loads allow for more clients and a lower service price point, which can open currently unaffordable Internet services to more clients in developing regions.

## References


[1] R. Gerster and S. Zimmermann, "Information and communication technologies (ICTs) and poverty reduction in sub sharan Africa: A learning study (synthesis)," Building Digital Opportunities Program, Tech. Rep., 2003.

[2] L. Centre and M. Jensen, "Fair access to Internet report," Open Society Initiative and International Development Research Centre (IDRC), Tech. Rep., 2004.

[3] R. Chandra and P. Bahl, "MultiNet: Connecting to multiple IEEE 802.11 networks using a single wireless card," in *IEEE Infocom*, March 2004.

[4] K. S. Sridhar and V. Sridhar, "Telecommunications infrastructure and economic growth: Evidence from developing countries," *Applied Econometrics and International Development*, vol. 7, no. 2, pp. 37–56, July 2007.

[5] B. S. Manoj, R. Ananthapadmanabha, and C. S. R. Murthy, "Multi-hop cellular networks: architecture and protocols for best-effort and real-time communication," *Journal of Parallel and Distributed Computing*, vol. 65, no. 6, pp. 767–791, June 2005.

[6] D. L. Johnson, "Evaluation of a single radio mesh network in South Africa," in *International Conference on Information and Communication Technologies and Development*, December 2007.

[7] S. Surana, R. Patra, and E. Brewer, "Simplifying fault diagnosis in locally managed rural WiFi networks," in *ACM Workshop on Networked Systems for Developing Regions (NSDR)*, August 2007.

[8] Y. Lin and Y. Hsu, "Multihop cellular: a new architecture for wireless communications," in *IEEE Infocom*, March 2000.

[9] H. Luo, X. Meng, R. Ramjee, P. Sinha, and L. Li, "The design and evaluation of unified cellular and ad-hoc networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 9, pp. 1060–1074, September 2007.

[10] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 1, pp. 24–37, January 2006.

[11] M. P. Wittie, K. Harras, K. C. Almeroth, and E. M. Belding, "On the implications of routing metric staleness in delay tolerant networks," *Computer Communications*, 2009, to appear.

[12] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," in *ACM SIGCOMM*, September 2006.

[13] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *ACM SIGCOMM*, August 2007.

[14] Y. Lin, B. Li, and B. Liang, "CodeOR: Opportunistic routing in wireless mesh networks with segmented network coding," in *IEEE International Conference on Network Protocols (ICNP)*, October 2008.

[15] H. Wu, C. Qiao, S. De, and O. Tonguz, "Integrated cellular and ad hoc relaying systems: iCAR," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2105–2115, October 2001.

[16] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep., 2000.

[17] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," IRTF RFC 4838 (Informational), April 2007.

[18] K. A. Harras, M. P. Wittie, K. C. Almeroth, and E. M. Belding, "ParaNets: A parallel network architecture for challenged networks," in *IEEE Workshop on Mobile Computing Systems and Applications (HotMobile)*, March 2007.

[19] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *ACM SIGCOMM*, August-September 2004.

[20] W. H. Yuen and H. Schulzrinne, "Performance evaluation of time-based and hop-based TTL schemes in partially connected ad hoc networks," in *IEEE International Conference on Communications*, June 2006.

[21] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[22] R. Koetter and M. Medard, "An algebraic approach to network coding," in *IEEE International Symposium on Information Theory*, June 2001.

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

[24] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Allerton Conference on Communication, Control, and Computing*, September 2003.

[25] J. Widmer and J. Le Boudec, "Network coding for efficient communication in extreme networks," in *SIGCOMM Workshop on Delay-Tolerant Networking (WDTN)*, August 2005.

[26] C. Fragouli, J. Le Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Computer Communications Review*, vol. 36, no. 1, pp. 63–68, January 2006.