Reliable Topology Design in Time-Evolving Delay-Tolerant Networks with Unreliable Links

Fan Li, *Member, IEEE*, Siyuan Chen, Minsu Huang, Zhiyuan Yin, Chao Zhang, and Yu Wang, *Senior Member, IEEE*

Abstract—Delay tolerant networks (DTNs) recently have drawn much attention from researchers due to their wide applications in various challenging environments. Previous DTN research mainly concentrates on information propagation and packet delivery. However, with possible participation of a large number of mobile devices, how to maintain efficient and dynamic topology becomes crucial. In this paper, we study the topology design problem in a predictable DTN where the time-evolving topology is known a priori or can be predicted. We model such a time-evolving network as a weighted directed space-time graph which includes both spacial and temporal information. Links inside the space-time graph are *unreliable* due to either the dynamic nature of wireless communications or the rough prediction of underlying human/device mobility. The purpose of our reliable topology design problem is to build a sparse structure from the original space-time graph such that (1) for any pair of devices, there is a space-time path connecting them with a reliability higher than the required threshold; (2) the total cost of the structure is minimized. Such an optimization problem is NP-hard, thus we propose several heuristics which can significantly reduce the total cost of the topology while maintain the "reliable" connectivity over time. In this paper, we consider both unicast and broadcast reliability of a topology. Finally, extensive simulations are conducted on random DTNs, a synthetic space DTN, and a real-world DTN tracing data. Results demonstrate the efficiency of the proposed methods.

Index Terms—Topology design, reliability, greedy algorithm, space-time graph, delay tolerant networks

1 Introduction

Delay or disruption tolerant networks (DTNs) have been used for a wide range of applications to provide robust data communications in challenging environments, such as pocket switched networks [1], [2], vehicular ad hoc networks [3], [4], [5], mobile sensor networks [6], [7], mobile social networks [8], [9], disaster-relief networks [10], or space communication networks [11], [12], [13]. In DTNs, the lack of continuous connectivity, network partitioning, long delays, unreliable time-varying links, and dynamic topology pose new challenges in design of DTN network protocols. Recently, many new routing schemes [2], [14], [15], [16], [17], [18] have been proposed for DTNs to take the intermittent connectivity and time-varying topology into consideration. In addition, different mobility studies [19], [20], [21] and graph modeling [22], [23], [24] have been conducted for DTNs to understand the underlying social and temporal characteristics of the network participants. However, there is little research on how to maintain a cost-efficient and reliable topology of time-evolving DTNs.

Network topology is always a key functional issue to the design of any network system. For different network applications, network topology can be designed or controlled under various objectives (such as power efficiency, fault tolerance, and throughput maximization). Topology control protocols have been well studied in wireless networks, especially, wireless ad hoc and sensor networks [25], [26], [27]. The focus of previous research is mainly on how to construct a cost-efficient structure from a static and connected topology (an underlying communication graph includes all possible communication links). However, in DTNs, the underlying topology often lacks continuous connectivity which makes existing topology control algorithms useless. Therefore, how to maintain efficient and dynamic topology of DTNs becomes crucial, especially with the participation of a large number of mobile devices.

In this paper, we study the topology design problem in a *time-evolving* and *predictable* DTN. DTNs often evolve over time: changes of network topology can occur if nodes move around. The node mobility and the evolution of topology are heavily dependent on social and temporal characteristics of the network participants. For *certain* type of networks, the temporal characteristics of topology could be known a priori or can be predicted from historical tracing data. For example, it is easy to discover the temporal patterns of topology for a DTN formed by either public buses [3] or satellites [11], [12], [13] which have fixed tours and schedules, or a mobile social network [20] consisting of students who share fixed class schedules. A recent study [28] also shows that human mobility model can achieve a 93% potential predictability. For this kind of time-evolving and predictable DTNs, the *space-time graph* model [29], instead of the static graph model, can be used to capture both the space and time dimensions of the dynamic network topology and to enable the emulation of any "store-and-forward" DTN routing methods. Given such a space-time graph including all possible temporal and spacial links, the *topology design*

- *F. Li and Z. Yin are with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China. E-mail: fli@bit.edu.cn,helloyinzhiyuan@gmail.com.*
- *S. Chen, M. Huang, C. Zhang, and Y. Huang are with the Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223. E-mail: {gausschen; huangms}@gmail.com, {czhang25; yu.wang}@uncc.edu.*

Manuscript received 17 Oct. 2013; revised 23 July 2014; accepted 29 July 2014. Date of publication 4 Aug. 2014; date of current version 1 May 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2014.2345392

problem aims to find a subgraph which maintains the connectivity over time between any two devices while minimizing the cost. Such problem is crucial for DTNs with either expensive communication links or a large number of devices. For example, in a space DTN [11], [30], the communication between two satellites or spacecrafts could be costly due to the energy constraints and the environmental issues (such as sunlight shadowing). Therefore, it is important to carefully plan the topology of space DTNs. On the other hand, in large-scale low-duty-cycle sensor networks [31], [32] or energy-harvesting sensor networks [33], in order to prolong application lifetime, many sensors can be put into sleep and thus their associated links are removed from the time-varying network topology. Since the duty cycles or harvested energy amount can be known a priori, efficient topology design can be performed for such time-evolving and predictable sensor networks.

In our recent study [34], we have proposed several heuristics for the basic topology design problem in time-evolving and predictable DTNs. Those methods all assume that the prediction of future links is perfect and links are *reliable* for communications, i.e., the packet delivery over these spacial or temporal links is guaranteed and without errors. This is clearly too optimistic in practice since the wireless communications are unreliable due to the lossy nature of wireless channels. It is observed that the wireless link reliability depends on many factors such as the transmission power, the receiver's sensitivity, the channel condition, and so on. In addition, even though certain type of network mobility can be predicted based on user behavior or historical data, the link prediction can be inaccurate sometimes. Therefore, in this paper, we remove the strong assumptions on reliable links and perfect prediction. Instead, for each spacial or temporal link in the space-time graph, we assume it has a probability to reflect its "reliability", either based on link quality estimation or mobility prediction. Higher the reliable probability of a link, higher chance the DTN transmissions over that link to be successful. Therefore, given the weighted space-time graph, the new *reliable topology design problem* (RTDP) aims to find a subgraph in which the reliability of DTN routing between any two devices is guaranteed to be higher than a required threshold and the total cost of topology is minimized. Notice that in many time-evolving and predictable DTNs (such as the interplanetary space DTN [11], [13]), maintaining dense structure is too expensive.

In this paper, we study the RTDP in a time-evolving DTN with unreliable links/predictions. Our major contributions are summarized as follows:

- We formally define the reliable topology design problem which aims to build a sparser structure (also a space-time graph) from the original space-time graph such that (1) for any pair of devices, there is a space-time path connecting them with the reliability higher than a required threshold; (2) the total cost of the structure is minimized. We also show that this new topology design problem is a NP-hard problem.
- We propose five heuristics which can significantly reduce the total cost of network topology while maintaining the DTN reliability over time.
- We also discuss how to address the reliable topology design problem when a flooding-based DTN routing is used instead of the single-copy DTN routing.
- Extensive simulations have been conducted on random DTN networks, a synthetic space DTN, and a real-world DTN tracing data [35]. Results demonstrate the efficiency of all proposed methods.

The rest of this paper is organized as follows. Section 2 summarizes related works in delay tolerant networks and topology design of wireless networks. Section 3 introduces the space-time graph model and the reliability of DTN networks. Section 4 formally defines the reliable topology design problem and provides its NP-hardness proof. Our proposed heuristics are presented in Section 5 and Section 6 for single-copy DTN routing and flooding-based DTN routing, respectively. Section 7 presents our simulation results over wide-range of networks. Finally, Section 8 concludes the paper by discussing the limitations of proposed methods and pointing out some possible future directions. A preliminary version of this paper appeared in [36].

2 Related Works

2.1 Delay Tolerant Networks

Existing research in DTNs mainly focuses on routing [14], [15], [16], [17], [19], [18]. Most of the existing schemes adopt the "store and forward" strategy, in which nodes store the packets in their buffers if there is no opportunity for message forwarding and wait for future opportunities. Then the key problem is how to select appropriate relay nodes for message forwarding during encounters. Two types of solutions are used in DTN routing: *single-copy* or *flooding based*.

In single-copy DTN routing, there is only one copy of each message in the network at any time so that the resulting propagation path of a message is a single path from the source to the destination. To make the right routing decision (i.e., picking the right relay at each step), a good metric to measure the ability of nodes to deliver the message is essential. Existing methods use metrics obtained from historical encounter information [17], [62], [63], [64], mobility information [19], [65], or social properties [2], [18], [66], [67]. During any encounters, the node with higher metric becomes the relay node [68].

Allowing multiple path propagation by flooding multiple messages in DTNs can significantly improve the chances of successful delivery. The simplest flooding routing method is epidemic routing [15], [42], where a copy of the message is given to every encountered node. However, such approach suffers from large overheads. To overcome this issue, some flooding-based routing methods (such as Spray & Wait [16] or Select & Spray [69], [70]) limit the number of copies of a message to a certain constant or the possible relays to certain nodes. Recently, there are also studies on how to maximize information propagation in DTNs via flooding [71], [72] and how to deploy additional throwboxes to assist the message delivery in DTNs [73], [74].

2.2 Time-Evolving Networks

Modeling the time-evolving networks has been studied in both mobile ad hoc networks [29], [44], [60] and DTNs [24], [46]. Xuan et al. [44] first study routing problem in a fixed schedule dynamic network modeled by an evolving graph (i.e., an indexed sequence of subgraphs of a given graph). Then, [24], [60] also use evolving graphs to evaluate various ad hoc and DTN routing protocols. Shashidhar et al. [29] study the routing problem in a space-time graph. Liu and Wu [46] also model a cyclic mobispace as a probabilistic space-time graph in which an edge between two nodes contains a set of discretized probabilistic contacts. All of these works only focus on the routing problem in the dynamic networks modeled by either evolving graphs or space-time graphs, and they usually aim to deliver the messages to their destinations. In this paper, we investigate the topology design problem in these networks with a different focus on the cost efficiency of the network topologies.

2.3 Topology Design

Topology design (or topology control) has drawn a significant amount of research interests in wireless ad hoc and sensor networks [25], [26], [27]. Primary topology design algorithms aim to maintain network connectivity and conserve energy. All existing methods deal with topology changes by re-performing the construction algorithm. Fortunately, most of the algorithms are localized, thus the update cost is not expensive. However, all methods assume that the underlying communication graph is fully connected at any time and they do not consider the time domain knowledge of network evolution.

The most relevant work with this study is our recent study [34] where we consider the basic topology design problem of reliable space-time graphs for either connectivity or spanner property. The proposed methods there assume that all links are reliable and the prediction of future links are prefect. In this paper, we remove such unrealistic assumption by considering the probability of link reliability. Notice that Liu et al. [61] have studied topology control over unreliable sensor networks. However, their study only consider the problem for a static sensor network without any time dimension dynamic. To our best knowledge, this paper is the first attempt to study topology design for time-evolving networks with unreliable links. We believe that topology can be controlled more wisely and efficiently if the network evolution over time is considered.

3 Models and Assumptions

3.1 Space-Time Graphs: Modeling Time-Evolving Networks

Assume that the time is divided into discrete and equal time slots, such as $\{1, \ldots, T\}$. Let $V = \{v_1, \ldots, v_n\}$ be the set of all individual nodes in the network (which represents the set of mobile devices). Since positions of individual nodes and the topology co-evolve over time, a sequence of static graphs can be defined over V to model the interactions among nodes in the time-evolving DTN during certain time slot. Fig. 1a illustrates such an example. Let $G^t = (V^t, E^t)$ be a directed graph representing the snapshot of the network at time slot t and a link $\overrightarrow{v_i^t v_j^t} \in E^t$ represents that node v_i can communicate to v_j at time t.

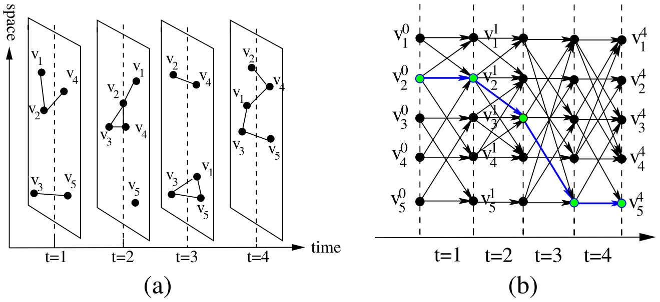


Fig. 1. A time-evolving DTN and its corresponding space-time graph: (a) the time-evolving topology of a DTN (a sequence of snapshots); (b) the corresponding space-time graph $\mathcal{G}$, where a space-time path from the source v_2 to the destination v_5 is highlighted in blue.

Then, the dynamic network is described by the union of all snapshots $\{G^t | t = 1 \cdots T\}$. However, such sequence of static graphs is not easy to use in the analysis or design of DTN routing protocols. For example, some of the snapshots may not be connected graphs at all (e.g., the first three snapshots in Fig. 1a), which makes routing tasks over them challenging.

In many existing DTN protocols, an aggregated graph over $\{G^t\}$ is used, where an edge exists between a pair of nodes if they have interacted at any point during the time period T. The weight of such a link is the probability of the occurrence (or the fraction of the occurrence over the total/historical period) of the link. For example, link $\overrightarrow{v_1 v_3}$ exists in two time slots out of total four slots, thus its weight is 0.5. Many DTN routing protocols [14], [17] estimate this contact probability based on past history and use it to select forwarding nodes. However, such aggregated view discards temporal information about the timing and order of interactions, which may cause failure of delivery or bad performances.

In this paper, we adopt the space-time graph [29] to model the time-evolving DTNs. We convert the sequence of static graphs $\{G^t\}$ into a space-time graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which is a directed graph defined in both spacial and temporal spaces. Fig. 1b illustrates the space-time graph of the same network. In the space-time graph $\mathcal{G}$, $T+1$ layers of nodes are defined and each layer has n nodes, thus the whole vertex set $\mathcal{V} = \{v_j^t | j = 1, \ldots, n \text{ and } t = 0, \ldots, T\}$ and there are $n(T+1)$ nodes in $\mathcal{G}$. Two kinds of links (spacial links and temporal links) are added between consecutive layers in $\mathcal{E}$. A temporal link $\overrightarrow{v_j^{t-1} v_j^t}$ (those horizontal links in Fig. 1b) connects the same node v_j across consecutive $(t-1)$th and tth layers, which represents the node carrying the message in the tth time slot. A spacial link $\overrightarrow{v_j^{t-1} v_k^t}$ represents forwarding a message from one node v_j to its neighbor v_k in the tth time slot (i.e., $\overrightarrow{v_j v_k} \in E^t$). By defining the space-time graph $\mathcal{G}$, any communication operation in the time-evolving network can be simulated on this directed graph. As the blue path shown in Fig. 1b, a space-time path from v_2^0 to v_5^4 shows a particular DTN routing strategy to deliver the packet from v_2 to v_5 in the network using four time slots: v_2 holds the packet for the first time slot, then passes it to v_3 at $t = 2$, etc.

We can now define the connectivity of a space-time graph, which is different from that of a static graph.

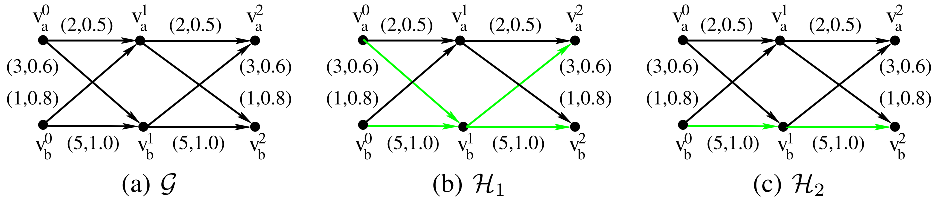


Fig. 2. *Examples of reliable topology design:* (a) the original space-time graph $\mathcal{G}$ with total cost 22, unicast reliability 0.36, and broadcast reliability 0.52; (b) a connected topology $\mathcal{H}_1$ with total cost 6, unicast reliability 0.25, and broadcast reliability 0.25; and (c) a connected topology $\mathcal{H}_2$ with total cost 12, unicast reliability 0.36, and broadcast reliability 0.4. Here, links are labeled by (cost, reliability), and green links are removed links from $\mathcal{G}$.

Definition 1. *A space-time graph $\mathcal{H}$ is* connected *over time period T if and only if there exists at least one directed path for each pair of nodes (v_i^0, v_j^T) (i and j in $[1, n]$).*

This guarantees that the packet can be delivered between any two nodes in the network over the period of T. Notice that a connected space-time graph does not require connectivity in each snapshot. Fig. 1 shows an example. Hereafter, we always assume that the original space-time graph $\mathcal{G}$ is connected over time period T. Note that to check whether a space-time graph is connected over T, the simplest way is running n rounds of breadth-first search (BFS), one from each v_i^0 to see whether they can reach all v_j^T for every j.

We further assume that for each link $e \in \mathcal{E}$ there is a *cost* $c(e)$, which is the energy cost associated with transiting a message on that link. The *total cost* of a space-time graph $c(\mathcal{H})$ is the summation of costs of all links in $\mathcal{H}$, i.e.,

$$c(\mathcal{H}) = \sum_{e \in \mathcal{H}} c(e). \tag{1}$$

Given the link costs, the *least cost path*$P_c^{\mathcal{H}}(u, v)$ is the path from u to v in $\mathcal{H}$ with the minimum total cost. The total cost of such path is denoted by $c^{\mathcal{H}}(u, v) = \sum_{e \in P_c^{\mathcal{H}}(u,v)} c(e)$, i.e., the summation of all link costs along the path.

3.2 Reliability of DTN Topology Over Unreliable Links

To consider the reliability of lossy wireless links or imperfect link predictions, we also define a *reliable probability*$r(e)$ for each link $e \in \mathcal{E}$, which represents the probability of a successful data transmission over link e. The unreliability of the spacial link could come from either link failure due to lossy wireless signal or poor prediction of future links. For each temporal link, it may also have a reliable probability for successfully holding the data over one time slot. Failures of a temporal link may come from buffer overflow or storage errors. However, such failures are seldom compared with those of wireless transmission. Thus, it is fine that all temporal links have perfect reliable probability (i.e., $r(e) = 1$). Here, we assume that the reliable probability of each link can be obtained through link estimation techniques at the link and physical layers [37] or mobility prediction techniques [38], [39]. Fig. 2a illustrates a simple space-time graph with just two devices. The label of each link e is a pair of cost and reliability, i.e., $(c(e), r(e))$. For example, in the first time slot, the cost of spacial link $\overrightarrow{v_a^0 v_b^1}$ is 3 and its reliability is 0.6. Given the reliability of each link, we can then define the reliability of a path P or a structure H. In this paper, two different types of reliabilities are considered for DTN topologies: one for single-copy DTN routing, and the other for flooding-based DTN routing.

In single-copy DTN routing, there is only one copy of each message in the network. When a node with a message meets other nodes, it just picks one node as the forwarding node to relay the message or keeps holding the message. Therefore, the resulting propagation path of a message is basically a single space-time path in $\mathcal{G}$. Given a path $P(u, v)$ connecting nodes u and v, the reliability of $P(u, v)$ is the production of reliability of all links in that path. For example, the path $v_a^0 \to v_a^1 \to v_b^2$ in Fig. 2a has reliability of $0.5 \times 0.8 = 0.4$. For a given routing topology $\mathcal{H}$, we can define the *most reliable path* $P_r^{\mathcal{H}}(u, v)$ as the path from u to v in $\mathcal{H}$ with the highest reliability. Let $r^{\mathcal{H}}(u, v) = \Pi_{e \in P_r^{\mathcal{H}}(u,v)} r(e)$ be the reliability of path $P_r^{\mathcal{H}}(u, v)$. For example, between v_a^0 and v_b^2 in Fig. 2a, $v_a^0 \to v_b^1 \to v_b^2$ is the most reliable path with reliability of $0.6 \times 1.0 = 0.6$. Then the *reliability* of the topology $\mathcal{H}$ is defined as follows.

Definition 2. *The* reliability *of a space-time graph $\mathcal{H}$ is the minimum reliability among all most reliable paths among each pair of nodes (v_i^0, v_j^T) (i and j in $[1, n]$), i.e.,*

$$r(\mathcal{H}) = \min_{1 \le i,j \le n} r^{\mathcal{H}}(v_i^0, v_j^T). \tag{2}$$

The reliabilities of structures shown in Figs. 2a, 2b and 2c are 0.36, 0.25 and 0.36 respectively. Notice that it is easy to calculate $r(\mathcal{H})$ given $\mathcal{H}$ by using any shortest path algorithms. Hereafter, we call this type of reliability *unicast reliability*.

Empirical and analytical studies [40], [41] have shown that allowing multiple copies propagation in DTN routing can significantly improve the chances of successful delivery. The simplest multi-copies DTN routing is flooding routing or epidemic routing [15], [16], [42], where a node with a message will relay it to every node it encounters. This type of flooding-based routing protocols propagate the copies of the message via multiple paths which leads to higher reliability. Therefore, we define a new type of reliability, *broadcast reliability*, as follows. For a given source-destination pair u, v in $\mathcal{H}$, the $r_B^{\mathcal{H}}(u, v)$ is the probability that a packet sent from node u over the routing topology $\mathcal{H}$ reaches node v under flooding-based DTN routing. To efficiently calculate the pair-wise broadcast reliability is not an easy job. Actually, it is known that the computation of such reliability over general graphs is a problem of NP-hard [43]. Fortunately, the space-time graph in our model is a very special directed acyclic graph where all paths from the sources to the destinations are T hops and there is not any loop. This property allows us to compute the reliability $r_B^{\mathcal{H}}(u, v)$ efficiently by using dynamic programming. We will present such an algorithm in Section 6. With $r_B^{\mathcal{H}}(u,v)$, the definition of the reliability $r_B(\mathcal{H})$ of $\mathcal{H}$ is straightforward and same as equation (2). The broadcast reliabilities of structures shown in Figs. 2a, 2b and 2c are 0.52, 0.25 and 0.4 respectively, which are higher than or equal to their unicast reliabilities.

4 Reliable Topology Design Problem

4.1 The Problem

We now can define the *reliable topology design problem* (RTDP) on weighted space-time graphs as follows.

Definition 3. *Given a connected space-time graph $\mathcal{G}$, the aim of RTDP is to construct a sparse space-time graph $\mathcal{H}$, which is a subgraph of the original space-time graph $\mathcal{G}$, such that (1) $\mathcal{H}$ is still connected over the time period T; (2) the reliability (unicast or broadcast reliability) is higher than or equal to a predefined threshold γ; and (3) the total cost of $\mathcal{H}$ is minimized.*

Notice that generally speaking denser space-time topology leads to better connectivity and reliability but more expensive in term of total cost. Therefore, the topology design aims to find a topology with certain reliability guarantee while minimizing the cost. If the threshold γ is a constant, the reliability requirement itself covers the basic connectivity, since it is obviously a much stronger requirement than basic connectivity. Once again, we assume that the original space-time graph $\mathcal{G}$ is connected and has reliability higher than γ. Figs. 2b and 2c show two sub-topologies of Fig. 2a with different costs and reliability. Note that these topologies are still connected over time, i.e., every node can find a space-time path to any other node.

4.2 Hardness and Discussions

The newly defined topology design problem is different with the standard space-time routing [29], [44], which only aims to find the most cost-efficient space-time path for a pair of source and destination. The topology design problem aims to maintain both cost-efficient and connected space-time routing topology for supporting reliable DTN transmissions between all pairs of nodes.

The topology design problem studied in our earlier work [34] is a special case of RTDP, in which reliability is not considered and all links are assumed to be perfectly reliable. Such unrealistic assumption makes the generated topologies by those methods unreliable in practice. For example, consider the unreliable DTN shown in Fig. 2 and the required reliability γ is 0.3 . The structure constructed by methods in [34] (as shown in Fig. 2b) has a smaller cost than the one shown in Fig. 2c, but it fails achieving the desired reliability.

The topology design problem over space-time graph is generally harder than the one over a static graph. For a static graph, a minimum spanning tree can achieve the goal of keeping connectivity with minimal cost. However, simply applying the spanning tree in each snapshot or over the entire space-time graph is not a valid solution anymore. In [34], we have already proved that the topology design problem over space-time graph without reliability requirement is NP-hard by a reduction from the *directed Steiner tree* (DST) problem [45]. Since such topology design problem is a special case of RTDP when $\gamma = 0$. Our RTDP is also NP-hard.

Theorem 1. *The newly defined* reliable topology design problem *on space-time graphs is NP-hard.*

In RTDP, we only consider the connectivity from the first time slot 0 to the last time slot T, i.e., packets are generated at time 0. In reality, if a packet arrives in the middle of the period T, it may not be able to reach the destination in the end of T via the constructed topology. However, in many time-evolving DTNs (such as the interplanetary space DTN [11], [12], [13], the cyclic mobispace [46] or periodic sensor networks [47]), the mobility process of network is periodic and the routing is repeated. Thus, the delivery of packets is still guaranteed in such cases. In addition, the reliable topology design problem studied here can be directly used for *low-duty-cycle sensor networks* [48], [49], [50], since putting sensor nodes into sleep periodically creates a time-evolving and predictable topology.

5 Heuristics of Reliable Topology Design for Single-Copy DTN Protocols

Since RTDP over space-time graphs is NP-hard, we now propose five different heuristics to construct a sparse structure that fulfills the connectivity and reliability requirements over time for single-copy DTN routing protocols. The first three heuristics are based on minimum cost reliable routing, while the other two are based on greedy algorithms. For all algorithms, the inputs are the original graph $\mathcal{G}$ and reliability requirement $\gamma \leq 1$, and the output is a subgraph $\mathcal{H}$ of $\mathcal{G}$.

5.1 Heuristics Based on Min Cost Reliable Path

One natural idea to construct a low-cost reliable subgraph, which guarantees the route reliability between any two nodes over the time period T, is keeping all the minimum cost reliable paths from v_i^0 to v_j^T for $i, j = 1, \ldots, n$ in $\mathcal{G}$. Here the *minimum cost reliable path* connecting u and v, denoted by $P_{cr}^{\mathcal{G}}(u,v)$, is the path with the least cost among all paths between u and v which have reliability higher than or equal to γ. Obviously, if we can find such paths for every pair of source and destination, the union of all of them satisfies the reliability requirement of our topology design problem. However, to find the shortest path with additional constrains in a graph itself is also a well known NP-hard problem, *restricted shortest path* problem [51], [52], [53]. Therefore, in this paper, we use one of the existing heuristics for restricted shortest path, backward-forward method (BFM) by Reeves and Salama [54].

The basic idea of backward-forward method is quite simple. Assume we want to find a reliable path from s to t. BFM first determines the *least-cost path* (LCP) and the *most reliable path* (MRP) from every node u to t. It then starts from s and explores the graph by concatenating two segments: (1) the so-far explored path from s to an intermediate node u, and (2) the LCP or the MRP from node u to t. BFM simply uses Dijkstra's algorithm with the following modification in the relaxation procedure: a link uv is relaxed if it reduces the total cost from s to v while its approximated end-to-end reliability obeys the reliability constraint. The computational complexity of the BFM is basically three times that of Dijkstra's algorithm. Let $P_{BFM}^{\mathcal{G}}(u,v)$ be the resulting path from u to v based on BFM. Notice that $P^{\mathcal{G}}_{BFM}(u, v)$ is not optimal for restricted shortest path, i.e., not the minimum cost reliable path connecting u and v in $\mathcal{G}$. However, based on simulation study by Kuipers et al. [55], BFM is one of the most efficient methods among all existing methods, it has small execution time and often generates good quality path compared with the optimal.

With backward-forward method, our first heuristic approach is to find a "minimum" cost reliable path for each pair of source and destination and take the union of them to form the subgraph which guarantees the overall reliability. Algorithm 1 shows the detailed algorithm. Its time complexity is $O(n^2T(\log(nT) + n))$ since we only need to compute n^2 minimum cost reliable paths of $\mathcal{G}$ (with $O(nT)$ nodes and at most $O(n^2T)$ links). This can be easily achieved by running $3n$ times of Dijkstra's algorithm whose complexity is $O(nT(\log(nT) + n))$. Hereafter, we refer this method as *union of minimum cost reliable path algorithm* (UMCRP).

Algorithm 1. Union of Min Cost Reliable Path (UMCRP)

$\mathcal{H} \leftarrow \phi$; $X = \{(v_i^0, v_j^T)\}$ for all integer $1 \le i, j \le n$.
for all pairs $(v_i^0, v_j^T) \in X$ **do**
  Find the "minimum" cost reliable path $P^{\mathcal{G}}_{BFM}(v_i^0, v_j^T)$ in $\mathcal{G}$ using Backward-Forward method.
  if $e \in P^{\mathcal{G}}_{BFM}(v_i^0, v_j^T)$ and $e \notin \mathcal{H}$ **then**
    $\mathcal{H} = \mathcal{H} \cup \{e\}$.
  end if
end for
return $\mathcal{H}$

Algorithm 2. Greedy Algorithm with Min Cost Reliable Path (GMCRP)

$\mathcal{H} \leftarrow \phi$; $X = \{(v_i^0, v_j^T)\}$ for all integer $1 \le i, j \le n$.
while $X \neq \phi$ **do**
  Find the minimum cost reliable paths for every pair nodes in X using BFM in $\mathcal{G}$, and assume $P_{BFM}(v_i^0, v_j^T)$ has the least cost among these paths.
  if $e \in P_{BFM}(v_i^0, v_j^T)$ **then**
    $\mathcal{H} \leftarrow e$; $c(e) \leftarrow 0$.
  end if
    $X \leftarrow X - (v_i^0, v_j^T)$.
end while
return $\mathcal{H}$

However, the structure built by the above method may contain more links than necessary. Therefore, we propose a new greedy algorithm (as shown in Algorithm 2) to further improve the performance. The basic idea is quite simple and as follows. Initially, we need to connect n^2 pair of nodes in X. In each round we pick the minimum cost reliable path between a pair nodes in X which is the minimum among all minimum cost reliable paths connecting any pair of nodes in X. Then we add all links in this path into $\mathcal{H}$, clear the costs of these links to zeros, and remove this pair from X. This procedure is repeated until all pair nodes (v_i^0, v_j^T) are guaranteed to be connected by reliable paths in $\mathcal{H}$. It is clear that the output of this method is much sparser than the one of UMCRP method. We refer this method as GMCRP. The time complexity of this algorithm is $O(Tn^5 + Tn^4\log(Tn))$ since in each round $3n$ times of Dijkstra's algorithm are running on the space-time graph and there are n^2 rounds.

The third method basically combines the least cost path and the minimum cost reliable path. It includes two steps. In the first step, we connect n^2 pair of nodes in $X = \{(v_i^0, v_j^T)\}$ for all integer $1 \le i, j \le n$ by adding all of the least cost paths between them. After this step, we have a structure $\mathcal{H}$ which connects all pairs of (v_i^0, v_j^T) but may not satisfy the reliability requirements yet. In the second step, for each pair (v_i^0, v_j^T), we calculate its reliability in $\mathcal{H}$. If the cost of $r^{\mathcal{H}}(v_i^0, v_j^T)$ is less than γ, we directly add the links in the minimum cost reliable path between this two nodes in $\mathcal{G}$ into $\mathcal{H}$. See Algorithm 3 for detail. Hereafter, we denote this method as LCP/MCRP. Both steps of LCP/MCRP need $n^2 \cdot O(n^2T(\log(nT) + n))$ time since they both run computation of shortest paths for $O(n^2)$ rounds. Therefore, the total time complexity is $O(n^4T(\log(nT) + n))$, same with the one of GMCRP.

Algorithm 3. Least Cost Path or Min Cost Reliable Path (LCP/MCRP)

$\mathcal{H} \leftarrow \phi$; $X = \{(v_i^0, v_j^T)\}$ for all integer $1 \le i, j \le n$.
for all pairs $(v_i^0, v_j^T) \in X$ **do**
  Find the least cost path $P_c^{\mathcal{G}}(v_i^0, v_j^T)$ in $\mathcal{G}$.
  if $e \in P_c^{\mathcal{G}}(v_i^0, v_j^T)$ and $e \notin \mathcal{H}$ **then**
  $\mathcal{H} = \mathcal{H} \cup \{e\}$.
  end if
end for
for all pairs $(v_i^0, v_j^T) \in X$ **do**
  if $r^{\mathcal{H}}(v_i^0, v_j^T) < \gamma$ **then**
  Add all links $e \in P^{\mathcal{G}}_{BFM}(v_i^0, v_j^T)$ into $\mathcal{H}$ if $e \notin \mathcal{H}$.
  end if
end for
return $\mathcal{H}$

5.2 Greedy-Based Heuristics

Both the fourth and fifth algorithms are based on the same greedy principle, deleting or adding edges in order and checking whether the reliability is achieved or void.

The basic idea of the fourth method is deleting edges from input graph (either the original graph $\mathcal{G}$ or the output graph from the first two algorithms) in a decreasing order based on link cost. Link $v_i^t v_j^{t+1}$ is removed if without this link the subgraph $\mathcal{H}$ can still achieve reliability of γ. Algorithm 4 shows the detail, we denote this algorithm as GDL hereafter. The time complexity analysis is straightforward. The sorting could be done in $O(n^2T\log(n^2T))$ with $O(n^2T)$ links. The *for*-loop includes n^2T rounds of computations of shortest paths. Thus, the time complexity of this part is $O(n^2T \cdot n^2T(\log(nT) + n)) = O(n^4T^2(\log(nT) + n))$. Therefore, total time complexity of GDL is also in order of $O(n^4T^2(\log(nT) + n))$, which is much higher than those of methods based on minimum cost reliable path. Notice that since the outputs of our first two algorithms are much sparser than the original graph, if we use them as the input of GDL it will save certain computation.

Algorithm 4. Greedy Algorithm to Delete Links (GDL)

$\mathcal{H} \leftarrow \mathcal{G}$.
Sort all links in link set $\mathcal{E}$ of $\mathcal{G}$ based on their costs.
for all $e \in \mathcal{E}$ (processed in decreasing order of costs) **do**
  if $r(\mathcal{H} - \{e\}) \geq \gamma$ **then**
    $\mathcal{H} = \mathcal{H} - \{e\}$.
  end if
end for
return $\mathcal{H}$

The fifth algorithm is also a greedy algorithm but processes links in the reverse order compared with the fourth algorithm. It starts from building a connected sparse structure and then adds more links into it to satisfy the reliability requirement. It also includes two steps. In the first step, again we take the union of all least cost paths between (v_i^0, v_j^T) for all integer $1 \leq i, j \leq n$ as $\mathcal{H}$. In the second step, remaining links are processed in the increasing order of cost. Link $v_i^t v_j^{t+1}$ is added to $\mathcal{H}$ if the current subgraph $\mathcal{H}$ cannot achieve reliability of γ yet. See Algorithm 5 for detail. Hereafter, we denote this method as GAL. The complexity of GAL is the same as the one of GDL ($O(n^4T^2(\log(nT) + n))$) since the second step is almost the same with GDL and the first step cost less time.

Algorithm 5. Greedy Algorithm to Add Links (GAL)

$\mathcal{H} \leftarrow \phi$; $X = \{(v_i^0, v_j^T)\}$ for all integer $1 \leq i, j \leq n$.
for all pairs $(v_i^0, v_j^T) \in X$ **do**
  Find the least cost path $P_c^{\mathcal{G}}(v_i^0, v_j^T)$ in $\mathcal{G}$.
  if $e \in P_c^{\mathcal{G}}(v_i^0, v_j^T)$ and $e \notin \mathcal{H}$ **then**
    $\mathcal{H} = \mathcal{H} \cup \{e\}$.
  end if
end for
for all $e \in \mathcal{E}$ but $\notin \mathcal{H}$ (processed in increasing order of costs) **do**
  if $r(\mathcal{H}) < \gamma$ **then**
    $\mathcal{H} = \mathcal{H} + \{e\}$.
  end if
end for
return $\mathcal{H}$

6 Heuristics of Reliable Topology Design for Flooding-Based DTN Protocols

In the above RTDP heuristics, we consider the unicast reliability of a topology for single-copy DTN routing protocols. If multiple copies are allowed during the DTN propagation, the reliability of the topology (the probability of success DTN delivery over the topology) will increase. Therefore, in this section, we study the RTDP heuristics under broadcast reliability for flooding-based DTN protocols.

We first introduce a dynamic programming algorithm (DP) to compute the broadcast reliability $r_B^{\mathcal{H}}(u, v)$ of a structure $\mathcal{H}$ for a given source-destination pair u, v. Even though such problem is NP-hard in general graphs [43], the nice loop-free property of our space-time graph model allows us to compute the reliability very efficiently with a dynamic programming algorithm. Basically, for any node v_i^t in $\mathcal{H}$, its broadcast reliability from a source node s can be calculated as follows:

$$r_B^{\mathcal{H}}(s, v_i^t) = 1 - \prod_{\overrightarrow{v_j^{t-1} v_i^t} \in \mathcal{H}} (1 - r_B^{\mathcal{H}}(s, v_j^{t-1}) r(\overrightarrow{v_j^{t-1} v_i^t})).$$

For example, in Fig. 2c, $r_B^{\mathcal{H}_2}(v_a^0, v_a^2) = 1 - (1 - 0.5 \times 0.5)(1 - 0.6 \times 0.6) = 0.52$. Given the structure $\mathcal{H}$, starting from a source node, the dynamic programming algorithm can compute the broadcast reliability of all other nodes within time of $O(nT(\log(nT) + n))$. Notice that the time complexity of DP algorithm is the same with that of Dijkstra's algorithm. We use this DP as a building block in our two RTDP heuristics.

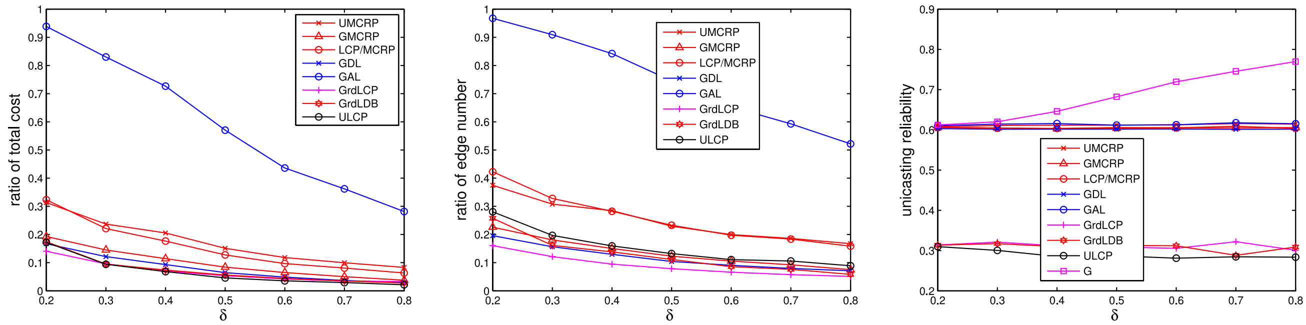
Notice that with broadcast reliability, the union of minimum cost reliable paths or the union of most reliable paths based on unicast reliability between all source-destination pairs may not achieve the required broadcast reliability. In addition, there is no effect algorithm to generate a multipath structure which can minimize the cost while guarantee the reliability for a given source-destination. Therefore, the only possible heuristics we have are two greedy based algorithms (Algorithm 4 and Algorithm 5). Both algorithms can be used for broadcast reliability. The only difference is using the DP algorithm to check the reliability of a topology.

7 Simulations

We have conducted extensive simulations to evaluate our proposed algorithms devoted to the RTDP problem over random networks, a synthetic space DTN, and pocket switched networks extracted from real tracing data. We implement and test five proposed algorithms (UMCRP (Algorithm 1), GMCRP (Algorithm 2), LCP/MCRP (Algorithm 3), GDL (v 4), and GAL (Algorithm 5)), and three existing topology algorithms (ULCP, GrdLCP, and GrdLDB) from [34], which maintain the connectivity over time but cannot guarantee the reliability. Here, ULCP is the union of least cost paths from v_i^0 to v_j^T for $i, j = 1, \ldots, n$, while GrdLCP and GrdLDB are greedy algorithms which add either the least-cost path or the least-density bunch in each round to connect nodes from v_i^0 to v_j^T until all pairs are connected. We use them as the references to our proposed methods since they are the only existing topology solutions for time-evolving networks. In all simulations, we take three metrics as the performance measurement for any algorithm devoted to *reliable topology design problem*:

- *Total Cost.* The total cost of the constructed topology $\mathcal{H}$ (output of the algorithm), i.e., $c(\mathcal{H}) = \sum_{e \in \mathcal{H}} c(e)$.
- *Total Number of Edges.* The total number of edges in $\mathcal{H}$, i.e., $|\mathcal{H}|$. Here, $|G|$ denotes total edge number of G.
- *Reliability.* $r(\mathcal{H}) = \min_{1 \leq i, j \leq n} r^{\mathcal{H}}(v_i^0, v_j^T)$, which could be either unicast or broadcast reliability.

The objective of our topology design methods is to efficiently construct topology structures with small total cost, small edge number and higher reliability.



(a) $c(\mathcal{H})/c(\mathcal{G})$ (b) $|\mathcal{H}|/|\mathcal{G}|$ (c) $r(\mathcal{H})$

Fig. 3. Simulation results on random networks with different density and $\gamma = 0.6$. (a)-(b): The cost (or edge number) of $\mathcal{H}$ is divided by the cost (or edge number) of $\mathcal{G}$, which illustrates how much saving achieved by the topology design, compared with the original network. (c): The reliability $r(\mathcal{H})$ shows the unicast reliability of $\mathcal{H}$.

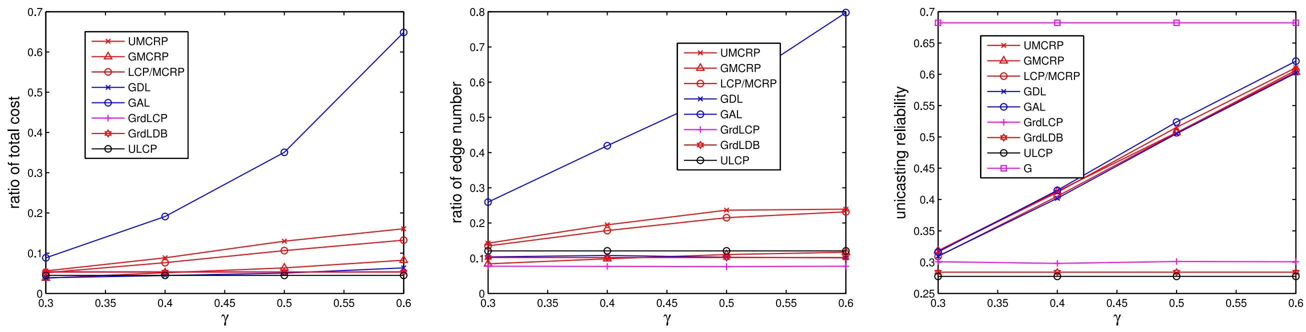
7.1 Simulations on Random Networks

The underlying time-evolving networks are randomly generated from random graph model. We first generate a sequence of static random graphs $\{G^t\}$ with 10 nodes ($n = 10$), spreading over 10 time slots ($T = 10$). For each time slot t, the static graph G^t is generated using the classical random graph generator. For each pair of nodes v_i and v_j, we insert edge $\overrightarrow{v_i v_j}$ with a fixed probability δ into G^t. Small value of δ leads to a sparse DTN, and $\delta = 1.0$ implies that the topology in each time slot is a complete graph. After generating $\{G^t\}$, we convert it into its corresponding space-time graph $\mathcal{G}$ with $n(T+1)$ nodes. We check the connectivity of $\mathcal{G}$ and only use those connected graphs for our simulations. The cost and reliability of each inserted edge are randomly chosen from 1 to 50 and from 0.8 to 1, respectively. Then we perform proposed topology design algorithms on $\mathcal{G}$. For each setting, we generate 100 random networks and report the average performance of topology design algorithms among them.

Topology design with unicast reliability. For the first set of simulations, we increase the network density by rising δ from 0.2 to 0.8, and keep reliability requirement γ at 0.6. Figs. 3a and 3b show the ratios between the total cost/number-edges of the generated graph $\mathcal{H}$ and that of the original graph $\mathcal{G}$ when δ increases. This ratio implies how much saving achieved by the topology design algorithm, compared with the original network. From the results, all algorithms can significantly reduce the cost of maintaining the connectivity. Even with the least density ($\delta = 0.2$), all algorithms except for GAL can save more than 70 percent cost and around 65 percent edges. When the network is denser, the saving of topology design is larger. For $\delta = 0.8$, more than 85 percent cost is saved. Comparing all methods, we can find the following observations. (1) ULCP, GrdLCP, and GrdLDB have the least costs since they only guarantee the connectivity while other proposed algorithms guarantee the reliability as well; (2) GAL has the largest cost due to simply adding many low-cost links; (3) GMCRP achieves smaller cost than UMCRP since it reuse some links in previous rounds; (4) LCP/MCRP's cost is between ULCP's and UMCRP's since it combines both LCP and MCRP; (5) Overall, GDL and GMCRP have the best costs among the proposed five methods. Fig. 3c shows the reliability of each structure generated by all methods. Hereafter, we also include the reliability of the original graph $\mathcal{G}$ (without any topology design algorithms) with the label of G in all of our results. Clearly, ULCP, GrdLCP, and GrdLDB are the only algorithms that cannot satisfy the reliability requirement $\gamma = 0.6$. Recall that they only maintain the connectivity over time. All other five algorithms can always achieve the desired reliability, and they have very close performance in term of reliability. But considering their costs, GDL, GMCRP, LCP/MCRP are better choices. When $\mathcal{G}$ is sparse (e.g. $\delta = 0.2$), its reliability is just over 0.6. In that case, our proposed algorithms remove around 70 percent edges but still keep the reliability at the desired level. This clearly demonstrates the power of topology design.

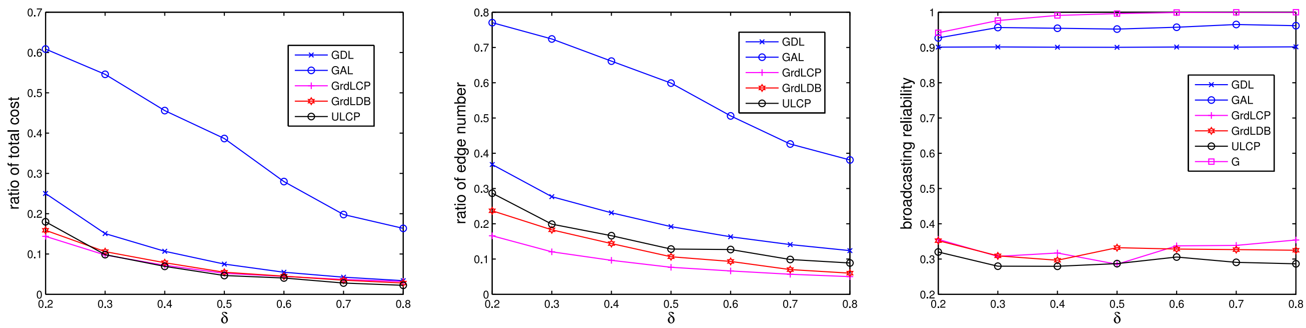
In the second set of simulations, we fixed the network density $\delta = 0.5$ and run our algorithms with different values of reliability requirement γ increasing from 0.3 to 0.6. From results shown in Fig. 4, we can observe that tighter reliability requirement results in higher cost and larger number of edge-usage of our topology algorithms. When the reliability requirement becomes looser, the restriction on removing links becomes weaker. In the extreme case with $\gamma = 0$, our RTDP problem converges to the basic topology design problem [34] which only preserves the connectivity. Notice that the performances of ULCP/GrdLCP/GrdLDB are not affected by the reliability requirement. In addition, GAL performs better when the reliability requirement is weak.

Topology design for broadcast reliability. We also implement the dynamic programming algorithm to compute the broadcast reliability, and test the performances of GDL (Algorithm 4), GAL (Algorithm 5), ULCP, GrdLCP, and GrdBDP on random networks. Similarly to previous experiments, we perform two sets of simulations. Fig. 5 shows the results when we increase δ from 0.2 to 0.8 and keep reliability requirement γ at 0.9. Fig. 6 shows the results when we increase broadcast reliability γ from 0.3 to 0.9 and keep δ at 0.5. From these results, it is obvious that both GDL and GAL can guarantee the reliability, but GDL is more efficient in term of cost. Compared with unicast reliability, fewer links and lower cost are needed to achieve the same level of reliability. Since ULCP, GrdLCP, and GrdLDB have similar performances over all experiments, hereafter, we will only report result from ULCP as the reference for our proposed methods.



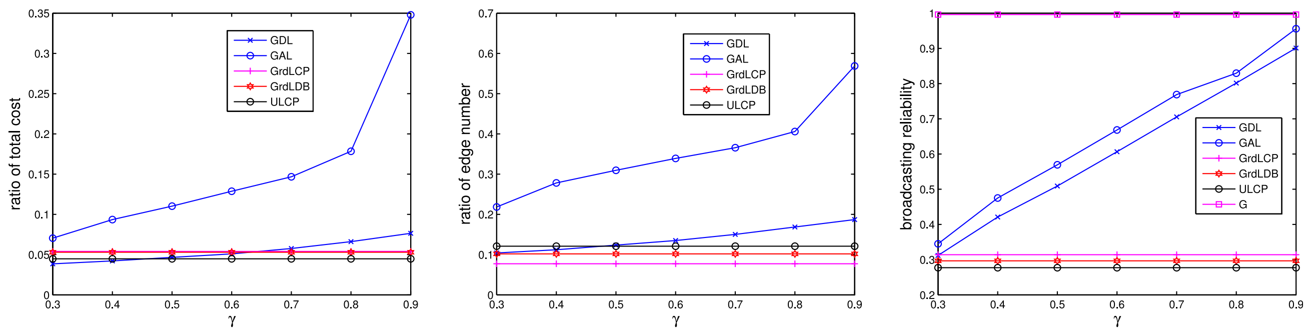
(a) $c(\mathcal{H})/c(\mathcal{G})$ (b) $|\mathcal{H}|/|\mathcal{G}|$ (c) $r(\mathcal{H})$

Fig. 4. Simulation results on random networks with fixed network density $\delta = 0.5$ and varying reliability requirement γ.



(a) $c(\mathcal{H})/c(\mathcal{G})$ (b) $|\mathcal{H}|/|\mathcal{G}|$ (c) $r(\mathcal{H})$

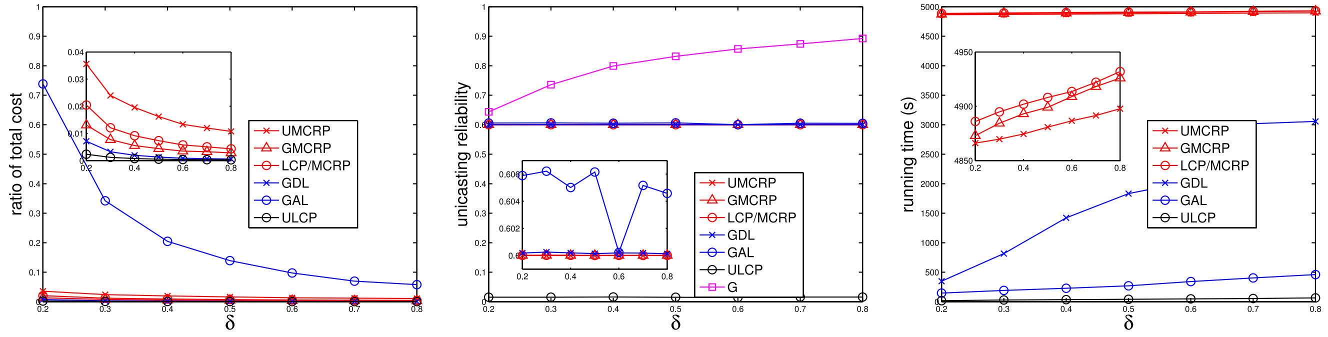
Fig. 5. Simulation results based on broadcast reliability on random networks with different density and $\gamma = 0.9$.



(a) $c(\mathcal{H})/c(\mathcal{G})$ (b) $|\mathcal{H}|/|\mathcal{G}|$ (c) $r(\mathcal{H})$

Fig. 6. Simulation results on broadcast reliability on random networks with network density $\delta = 0.5$ and varying reliability requirement γ.

Results and running time on larger networks. So far, we only try our algorithms with a small scale network with 10 nodes and over 10 time slots. With the increase of the number of nodes n and the length of time period T, the proposed algorithms need much longer time to find solutions. Hereafter, we denote the running time of each algorithm as the total time to generate the output topology $\mathcal{H}$. Due to the hardness of the proposed RTDP and the complexity of space-time graph itself (with $O(nT)$ nodes and possible $O(n^2T)$ links), it is challenging to find efficient algorithms to achieve good performance over large-scale networks. To accelerate the running time of proposed methods, we further optimize the implementations of these algorithms. The basic ideas include (1) caching the results from previous calculation of shortest paths to avoid redundant running of Dijkstra's algorithm; (2) caching the relationships among links and reliable paths so that greedy-based heuristics can check only a subset of links or paths; (3) adding and deleting links in batches. Figs. 7 and 8 show the results of our algorithms over larger random networks with 50 nodes and 50 time slots (i.e., around 2,500 nodes in the $\mathcal{G}$) for unicast reliability and broadcast reliability, respectively. All other parameters are kept the same as previous experiments. Here, we do not include the ratios of edge numbers since they have the same trends with the ratios of total costs. It is clear that all conclusions from the experiments over small random networks are still valid. In term of running time, GAL and ULCP run most fast. It is interesting to see that GDL uses significantly longer time than GAL and ULCP especially for denser networks, even though the time complexities of GDL and GAL are at the same level. This is mainly because it iterates almost all links in $\mathcal{G}$ in decreasing order of costs until the



(a) $c(\mathcal{H})/c(\mathcal{G})$ (b) $r(\mathcal{H})$ (c) running time

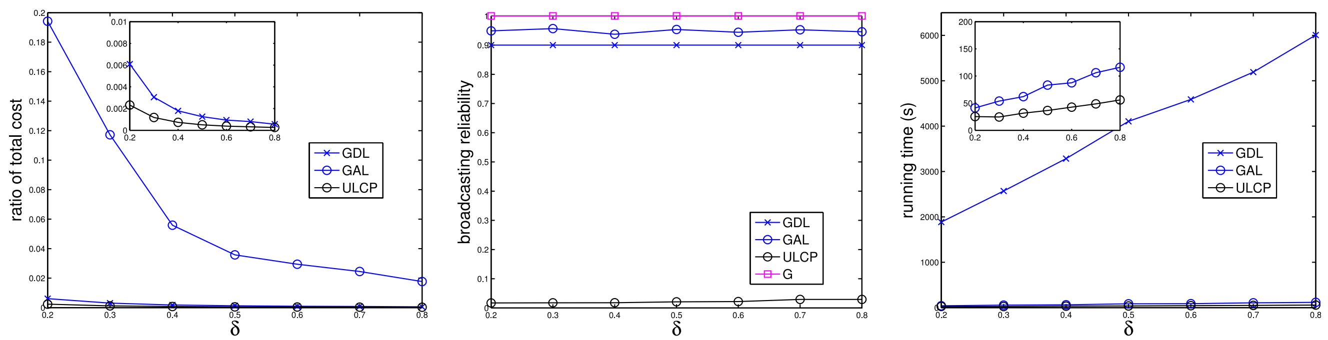
Fig. 7. Simulation results on unicast reliability on larger random networks ($n = 50$ and $T = 50$) with different density and $\gamma = 0.6$.

reliability requirement cannot be satisfied. All MCRP related methods need the most time since solving the minimum cost reliable path problem itself is a very challenging task. We also perform simulations on networks with different settings (various reliability, density, size and time length). The results and conclusions are similar, thus they are omitted here due to space limit.

7.2 Simulations on a Space DTN for Mars Exploration

Space DTNs (also called interplanetary Internet) [11], [13], [56] have been widely accepted among research community and space agencies, worldwide. Several space DTN testbeds and platforms have been implemented and tested recently, such as NASA JPL's Deep Impact Networking Experiment [57] or ESA's DTN testbed deployment project [58]. In space DTNs, the network has intermittent connectivity among multiple satellites, landed rovers, or spacecrafts but all of their trajectories and orbits are predetermined, thus communication opportunities are predictable and time-evolving topology is known a priori. On the other hand, the energy sources of satellites or landed rovers entail solar energy, which becomes very limited considering such a system travels farther away from the Sun or is shadowed (sunlight is blocked) for long periods for planet orbiting platforms. The level of power availability is a critical factor on the performance of space DTNs. To save energy, it is important and efficient to carefully plan the topology of the DTNs by limiting the communication links.

In this experiment, we consider a synthetic space DTN for Mars exploration. Space DTNs for Mars exploration have been studied in [12], [59]. It is well-known that relay communication via Mars-orbiting satellites (spacecrafts) offers significant advantages relative to conventional direct-to-Earth communications. Several NASA's or ESA's Mars orbiters (such as Mars Reconnaissance Orbiter (MRO), Mars Odyssey Orbiter, and Mars Express (MEX)) have been used to support the landed Mars rovers (such as Spirit and Opportunity rovers). Fig. 9a shows the Mars DTN relay network, which has been used [12], [56]. Here, we have two landed rovers which rely on three Mars orbiters to relay data packets to three deep space network (DSN) ground stations (Canberra, Goldstone, and Madrid). As in [12], [56], we assume that the landed rovers do not directly communicate to Earth and the orbiters do not have cross-links. Therefore, all possible communication links in this network is shown in Fig. 9b, which form the network topology among all nodes. Since all Earth ground stations can be connected to each other via Internet, we can use a virtual node to replace them, as in Fig. 9c. We directly use the parameters from [12], [56], [59]. For example, Odyssey and MRO (v_3 and v_4) need 2 hours to complete a total orbit around Mars, while MEX (v_5) uses 6 hours. Thus, rovers v_1 and v_2 can have a spacial link with v_3 and v_4 every two time slots (each time slot with one hour length), while have a link with v_5 every 6 hours. In Fig. 9c, we show a space-time graph with $T = 6$. Since the three DSN ground stations are located approximately 120 degrees apart on Earth, they can provide at least one link for each time slot. We assume that the links between rovers and orbiters (red links in the figure) have reliability of 0.999 and cost of 1; the links between orbiters and Earth stations (blue links in



(a) $c(\mathcal{H})/c(\mathcal{G})$ (b) $r(\mathcal{H})$ (c) running time

Fig. 8. Simulation results on broadcast reliability on larger random networks ($n = 50$ and $T = 50$) with different density and $\gamma = 0.9$.

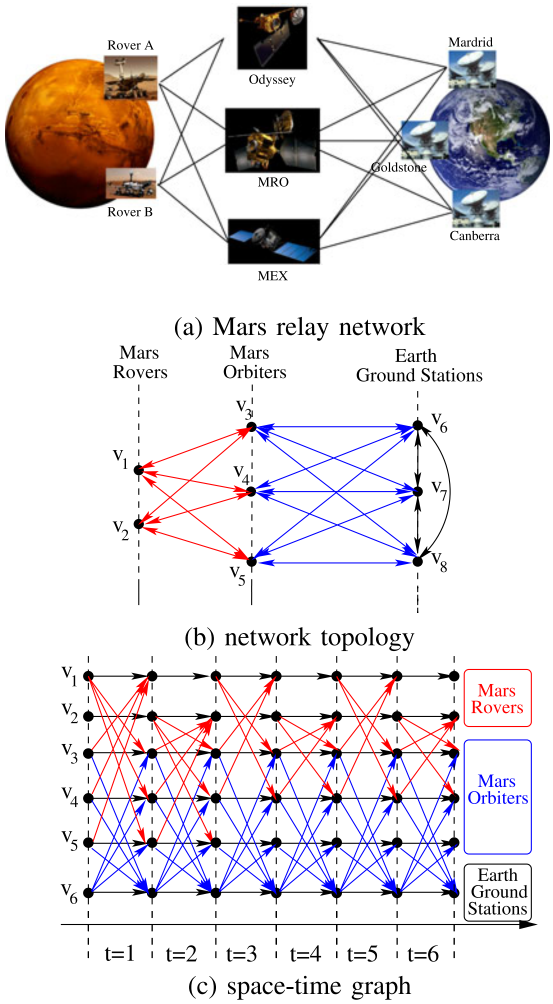


Fig. 9. A time-evolving space DTN for Mars exploration: (a) the possible relay links of the two Mars rovers to the ground stations; (b) the corresponding network topology with time-varying links; (c) the corresponding space-time graph $\mathcal{G}$, where three ground stations are merged to a single node. Here we use red links to denote the links between the rovers and relay orbiters, while use blue links to denote the links between relay orbiters and the ground stations.

the figure) have reliability of either 0.95 or 0.99 (randomly picked) and cost of 2; the remaining temporal links and links among ground stations (black links) have perfect reliability and zero cost. Once again the reliability settings are from [12], [56], [59].

Figs. 10a and 10b show the simulation results over this space DTN for unicast and broadcast reliability, respectively. For the unicast case, clearly GMCRP and GDL can significantly reduce the total cost of the DTN, even less than the ULCP. All proposed method can satisfy the reliability requirement, except for ULCP. For the broadcast case, GDL can achieve the best ratio of total cost (less than 30 percent), while both GDL and GAL can satisfy the reliability requirement. All these conclusions are consistent with those in the experiments over random networks. This experiment also confirms the efficiency of our proposed algorithms in a real-world application.

7.3 Simulations on Real-World DTN Tracing Data

Last, we test the proposed methods over a real-world wireless tracing data. Particularly, we select a data set from the Cambridge Haggle data set [35]. In this data set, connections among 78 mobile iMote Bluetooth nodes carried by researchers and additional 20 stationary nodes are recorded over 4 days during IEEE Infocom 2006. In our simulations, we only consider the 78 mobile nodes and the first half of the period in the tracing data. We divide the time period of the tracing data into 20 time slots. For each time slot t, if there is a contact trace which is overlapping with this slot, we add a spacial link between the two corresponding nodes in G^t. For each round of simulation, we extract a slice of the network which contains 10 mobile nodes. Once again, we check and make sure that the generated space-time graph $\mathcal{G}$ is connected over the period of 20 time slots. Costs and reliability probabilities are randomly generated from 1 to 50 and 0.8 to 1.0 respectively for both spacial and temporal links. Then our topology design algorithms are performed over these 10-node DTNs. In our simulation, 14 rounds of simulations are conducted and average measurements are plotted in Fig. 11. And the reliability requirement γ is set to 0.15 and 0.4 for unicast and broadcast cases respectively. The same conclusions can be drawn from these results: (1) all algorithms can reduce the cost remarkably (more than 75 percent) except for GAL; (2) ULCP uses the least cost among all methods but cannot satisfy the reliability requirement; (3) all the other methods can satisfy the reliability requirement; (4) topologies with similar amount of cost or links can achieve higher reliability by using flooding-based methods than single-copy methods.

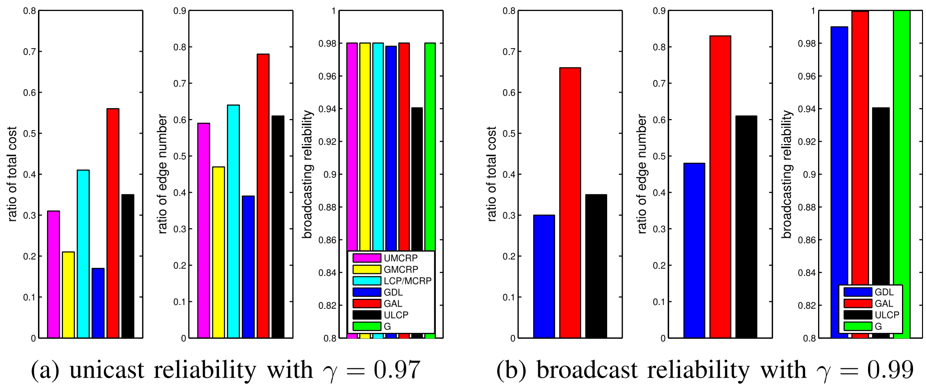
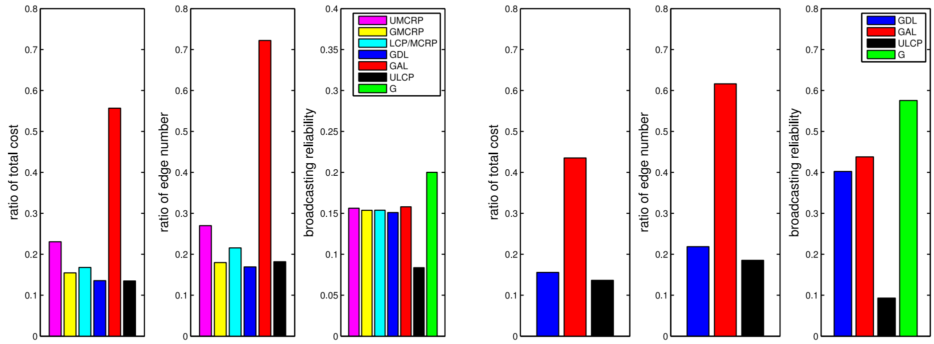


Fig. 10. Simulation results on the space DTN network.



(a) unicast reliability with $\gamma = 0.15$ (b) broadcast reliability with $\gamma = 0.4$

Fig. 11. Simulation results on networks from Cambridge Haggle [35] tracing data. Results are averages over 14 small networks.

8 Conclusion

We study reliable topology design problem in a predictable time-evolving DTN with unreliable links modeled by a probabilistic space-time graph. We first show that it is NP-hard, then propose a set of heuristics which can significantly reduce the cost of topology while maintain the connectivity and reliability of paths over time. Simulation results from random networks, a synthetic space DTN, and real-world tracing data demonstrate the efficiency of our methods. We believe that this paper presents the first step in exploiting topology design for time-evolving DTNs with unreliable links.

The topology design problem defined in this paper and our proposed algorithms have several limitations and weaknesses. (1) in our problem, the connectivity and reliability are only considered for a fixed time period T; (2) we still assume that the predictions of future links and their reliabilities are feasible, which limit the application of this problem to certain DTNs; (3) here we mainly consider the connectivity and reliability of the constructed topology, however, removing links may hurt the performance of communication protocols (such as routing). Thus it is interesting to conduct a study of the tradeoff between the cost saving for our design and the reduction of network performance. We leave such study as one of our future works. Other possible future works include: (1) design more efficient algorithms with lower complexity to achieve reliability over space-time graphs; (2) investigate how to adapt the constructed structure to unexpected changes in the network such as node failures.

Acknowledgments

The work of F. Li and Z. Yin was partially supported by the National Natural Science Foundation of China under Grant No. 60903151 and 61370192, and the Beijing Natural Science Foundation under Grant No. 4122070. The work of S. Chen, M. Huang, C. Zhang, and Y. Wang was supported in part by the US National Science Foundation under Grant No. CNS-1050398, CNS-1319915, and CNS-1343355. F. Li, S. Chen, and M. Huang contributed equally to this work. Y. Wang is the corresponding author.

References

[1] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proc. ACM WDTN*, 2005, pp. 244–251.

[2] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *Proc. ACM Mobihoc*, 2008, pp. 241–250.

[3] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang, "Study of a bus-based disruption-tolerant network: Mobility modeling and impact on routing," in *Proc. ACM MobiCom*, 2007, pp. 195–206.

[4] F. Li, L. Zhao, X. Fan, and Y. Wang, "Hybrid position-based and DTN forwarding for vehicular sensor networks," *Int. J. Distrib. Sen. Netw.*, vol. 2012, p. 10, 2012.

[5] Q. Fu, B. Krishnamachari, and L. Zhang, "DAWN: A density adaptive routing for deadline-based data collection in vehicular delay tolerant networks," *Tsinghua Sci. Technol.*, vol. 18, no. 3, pp. 230–241, 2013.

[6] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with Zebranet," *ACM SIGOPS Oper. Syst. Rev.*, vol. 36, no. 5, pp. 96–107, 2002.

[7] Y. Wang, H. Dang, and H. Wu, "A survey on analytic studies of delay-tolerant mobile sensor networks: Research articles," *Wirel. Commun. Mob. Comput.*, vol. 7, no. 10, pp. 1197–1208, 2007.

[8] A. Chaintreau, P. Fraigniaud, and E. Lebhar, "Opportunistic spatial gossip over mobile social networks," in *Proc. ACM 1st Workshop Online Social Netw.*, 2008, pp. 73–78.

[9] Z.-B. Dong, G.-J. Song, K.-Q. Xie, and J.-Y. Wang, "An experimental study of large-scale mobile social network," in *Proc. WWW*, 2009, pp. 1175–1176.

[10] M. Y. S. Uddin, D. M. Nicol, T. F. Abdelzaher, and R. H. Kravets, "A post-disaster mobility model for delay tolerant networking," in *Proc. Winter Simulation Conf.*, 2009, pp. 2785–2796.

[11] S. Burleigh and A. Hooke, "Delay-tolerant networking: An approach to interplanetary internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, pp. 128–136, 2003.

[12] C. V. Samaras and V. Tsaoussidis "Design of delay-tolerant transport protocol (DTTP) and its evaluation for Mars," *Acta Astronautica*, vol. 67, pp. 863–880, 2010.

[13] J. Mukherjee and B. Ramamurthy, "Communication technologies and architectures for space network and interplanetary Internet," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 2, pp. 881–897, May 2013.

[14] Q. Yuan, I. Cardei, and J. Wu, "Predict and relay: An efficient routing in disruption-tolerant networks," in *Proc. ACM MobiHoc*, 2009.

[15] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke Univ., Durham, NC, USA, Tech. Rep. CS-200006, 2006.

[16] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM WDTN*, 2005, pp. 252–259.

[17] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM Mob. Comp. Comm. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.

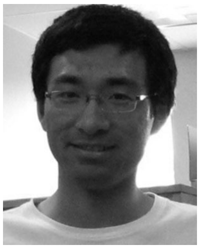
[18] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: Positive and negative social effects," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 1, pp. 87–401, Feb. 2013.

[19] J. Leguay, T. Friedman, and V. Conan, "Evaluating mobility pattern space routing for DTNs," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1–10.

[20] V. Srinivasan, M. Motani, and W.T. Ooi, "Analysis and implications of student contact patterns derived from campus schedules," in *Proc. ACM Mobicom*, 2006, pp. 86–97.
[21] K. Lee, S. Hong, S.J. Kim, I. Rhee, and S. Chong, "SLAW: A new mobility model for human walks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2009, pp. 858–863.
[22] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proc. ACM MobiHoc*, 2007, pp. 32–40.
[23] T. Hossmann, T. Spyropoulos, and F. Legendre, "Know thy neighbor: Towards optimal mapping of contacts to social graphs for DTN routing," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2009, pp. 1–9.
[24] L. Arantes, A. Goldman, and M. V. dos Santos, "Using evolving graphs to evaluate DTN routing protocols," in *Proc. ExtremeCom*, 2009.
[25] R. Ramanathan and R. Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2000, pp. 404–413.
[26] R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang, "Distributed topology control for wireless multihop ad-hoc networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2001, pp. 1388–1397.
[27] Y. Wang and X.Y. Li, "Localized construction of bounded degree and planar spanner for wireless ad hoc networks," *Mob. Netw. Appl.*, vol. 11, no. 2, pp. 161–175, 2006.
[28] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi, "Limits of predictability in human mobility," *Science*, vol. 327, pp. 1018–1021, 2010.
[29] S. Merugu, M. Ammar, and E. Zegura, "Routing in space and time in networks with predictable mobility," GIT, Tech. Rep. CC-04-07, 2004.
[30] C. Caini, P. Cornice, R. Firrincieli, and D. Lacamera, "A DTN approach to satellite communications," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 820–827, May 2008.
[31] Y. Gu and T. He, "Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 12, pp. 1741–1754, Oct. 2011.
[32] L. Chen, Y. Gu, S. Guo, T. He, Y. Shu, F. Zhang, and J. Chen, "Group-based discovery in low-duty-cycle mobile sensor networks," in *Proc. IEEE Commun. Soc. Conf.Sensor Mesh Ad Hoc Commun. Netw.*, 2012, pp. 542–550.
[33] F. Li, S. Chen, S. Tang, X. He, and Y. Wang, "Efficient topology design in time-evolving and energy-harvesting wireless sensor networks," in *Proc. IEEE Mob. Ad Hoc Sensor Syst.*, 2013, pp. 1–9.
[34] M. Huang, S. Chen, Y. Zhu, and Y. Wang, "Topology control for time-evolving and predictable delay-tolerant networks," *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2308–2321, Sep. 2013.
[35] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. (2006, Sep.). CRAWDAD data set cambridge/haggle (v. 2006-09-15). [Online]. Available: http://crawdad.cs.dartmouth.edu/cambridge/haggle
[36] M. Huang, S. Chen, F. Li, and Y. Wang, "Topology design in time-evolving delay-tolerant networks with unreliable links," in *Proc. IEEE Global Commun. Conf.*, 2012, pp. 5296–5301.
[37] N. Baccour, A. Koubaa, M. Ben Jamaa, H. Youssef, M. Zuniga, and M. Alves, "A comparative simulation study of link quality estimators in wireless sensor networks," in *Proc. IEEE Modeling Anal. Simul. Comput. Telecommun. Syst.*, 2009, pp. 1–10.
[38] W. Su, S. Lee, and M. Gerla, "Mobility prediction in wireless networks," in *Proc. IEEE 21st Century Military Commun.*, 2000, pp. 491–495.
[39] W.-S. Soh and H. S. Kim, "Dynamic bandwidth reservation in cellular networks using road topology based mobility predictions," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2004, pp. 2766–2777.
[40] B. Gallagher, D. Jensen, B. N. Levine, "Explaining routing performance in disruption tolerant networks," Univ. Massachussetts, Amherst, MA, USA, Tech. Rep. 05-57, 2005.
[41] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, Feb. 2008.
[42] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Comput. Netw.*, vol. 51, pp. 2867–2891, 2007.
[43] A. Agrawal and R. E. Barlow, "A survey of network reliability and domination theory," *Oper. Res.*, vol. 32, pp. 478–492, 1984.
[44] B. B. Xuan, A. Ferreira, and A. Jarry, "Computing shortest, fastest, and foremost journeys in dynamic networks," *Int. J. Found. Comput. Sci.*, vol. 14, no. 2, pp. 267–285, 2003.
[45] L. Zosin and S. Khuller, "On directed Steiner trees," in *Proc. ACM SODA*, 2002, pp. 59–63.
[46] C. Liu and J. Wu, "Routing in a cyclic mobispace," in *Proc. ACM MobiHoc*, 2008, pp. 369–382.
[47] T. Al-Khdour and U. Baroudi, "An energy-efficient distributed schedule-based communication protocol for periodic wireless sensor networks," *Arab. J. Sci. Eng.*, vol. 35, no. 2B, pp. 155–168, 2010.
[48] Y. Gu and T. He, "Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 12, pp. 1741–1754, Oct. 2011.
[49] Z. Li, Y. Peng, D. Qiao, and W. Zhang, "LBA: Lifetime balanced data aggregation in low duty cycle sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 1844–1852.
[50] Y. Cao, S. Guo, and T. He, "Robust multi-pipeline scheduling in low-duty-cycle wireless sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 361–369.
[51] H. C. Joksch, "The shortest route problem with constraints," *J. Math. Anal. Appl.*, vol. 14, pp. 191–197, 1966.
[52] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 7, pp. 1228–1234, Sep. 1996.
[53] F. A. Kuipers, "An overview of constraint-based path selection algorithms for QoS routing," *IEEE Commun. Mag.*, vol. 40, no. 12, pp. 50–56, Dec. 2002.
[54] D. S. Reeves and H. F. Salama, "A distributed algorithm for delay-constrained unicast routing," *IEEE/ACM Trans. Netw.*, vol. 8, pp. 239–250, Apr. 2000.
[55] F. Kuipers, T. Korkmaz, M. Krunz, and P. V. Mieghem, "Performance evaluation of constraint-based path selection algorithms," *IEEE Netw.*, vol. 18, no. 5, pp. 16–23, Sep./Oct. 2004.
[56] J. S. Segui, E. H. Jennings, and S.C. Burleigh, "Enhancing contact graph routing for delay tolerant space networking," in *Proc. IEEE Global Telecommun. Conf.*, 2011, pp. 1–6.
[57] A. Jenkins, S. Kuzminsky, K. K. Gifford, R. L. Pitts, and K. Nichols, "Delay/disruption-tolerant networking: flight test results from the international space station," in *Proc. IEEE Aerospace Conf.*, 2010, pp. 1–8.
[58] C. V. Samaras, I. Komnios, S. Diamantopoulos, E. Koutsogiannis, V. Tsaoussidis, G. Papastergiou, and N. Peccia, "Extending Internet into space - ESA DTN testbed implementation and evaluation," in *Proc. Int. Conf. Mobile Lightweight Wireless Syst.*, 2009, pp. 397–404.
[59] C. D. Edwards, Jr., B. Arnold, R. Depaula, G. Kazz, C. Lee, and G. Noreen, "Relay communications strategies for Mars exploration through 2020," *Acta Astronautica*, vol. 59, pp. 310–318, 2006.
[60] J. Monteiro, A. Goldman, and A. Ferreira, "Performance evaluation of dynamic networks using an evolving graph combinatorial model," in *Proc. IEEE Int. Conf. Wireless Mobile Comput. Netw. Commun.*, 2006, pp. 1–8.
[61] Y. Liu, Q. Zhang, and L. Ni, "Opportunity-based topology control in wireless sensor networks," in *Proc. IEEE 28th Int. Conf. Distrib. Comput. Syst.*, 2008, pp. 421–428.
[62] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1–11.
[63] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot, "Diversity of forwarding paths in pocket switched networks," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, 2007, pp. 161–174.
[64] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella, "Hibop: A history based routing protocol for opportunistic networks," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, 2007, pp. 1–12.
[65] J. Wu, M. Xiao, and L. Huang, "Homing spread: Community home-based multi-copy routing in mobile social networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2013, pp. 2319–2327.
[66] A. Mtibaa and K. Harras, "CAF: Community aware framework for large scale mobile opportunistic networks," *Comput. Commun.*, vol. 36, no. 2, p. 180190, 2013
[67] A. Mtibaa and K. Harras, "Social forwarding in large scale networks: Insights based on real trace analysis," in *Proc. IEEE Int. Conf. Comput. Commun. Netw.*, 2011, pp. 1–8.
[68] V. Erranmilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *Proc. ACM MobiHoc*, 2008, pp. 251–260.

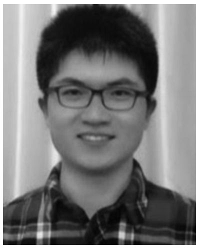
[69] A. Mtibaa and K. Harras, “Select&Spray: Towards deployable opportunistic communication in large scale networks,” in *Proc. ACM MobiWac*, 2013, pp. 1–8.
[70] A. Mtibaa and K. Harras, “Exploiting space syntax for deployable mobile opportunistic networking,” in *Proc. IEEE 10th Int. Conf. Mobile Ad Hoc Sensor Syst.*, 2013, pp. 533–541.
[71] W. Rao, K. Zhao, Y. Zhang, P. Hui, and S. Tarkoma, “Maximizing timely content advertising in delay tolerant networks,” in *Proc. IEEE 9th Annu. Commun. Soc. Conf. Sensor Mesh Ad Hoc Commun. Netw.*, 2012, pp. 254–262.
[72] S. Tang, J. Yuan, X.-Y. Li, Y. Wang, C. Wang, and X. Liu, “MINT: Maximizing information propagation in predictable delay-tolerant network,” in *Proc. ACM MobiHoc*, 2013, pp. 253–256.
[73] Y. Zhu, C. Zhang, and Y. Wang, “Social based throwbox placement in large-scale throwbox-assisted delay tolerant networks,” in *Proc. IEEE ICC*, 2014.
[74] W. Zhao, Y. Chen, M. Ammar, M. Corner, B. Levine, and E. Zegura, “Capacity enhancement using throwboxes in DTNs,” in *Proc. IEEE Int. Conf. Mobile Ad Hoc Sensor Syst.*, 2006, pp. 31–40.

Fan Li received the MEng degree in electrical engineering from the University of Delaware in 2004, the PhD degree in computer science from the University of North Carolina at Charlotte in 2008, and the BEng and MEng degrees in communications and information system from Huazhong University of Science and Technology, China. She is currently an associate professor in the School of Computer Science at Beijing Institute of Technology, China. Her current research focuses on wireless networks, ad hoc and sensor networks, and mobile computing. She is a member of the IEEE and ACM.

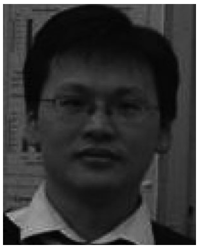


Siyuan Chen received the BS degree from Peking University, China, in 2006, and the PhD degree in computer science from the University of North Carolina at Charlotte in 2012. His research interests include wireless networks, ad hoc and sensor networks, delay tolerant networks, and algorithm design.

Minsu Huang received the BEng degree in mechanical engineering from Central South University, China, in 2003, the MS degree in computer science from Tsinghua University, China, in 2006, and the PhD degree in computer science from the University of North Carolina at Charlotte in 2012. His research interests include wireless networks, ad hoc and sensor networks, delay tolerant networks, and algorithm design.



Zhiyuan Yin received the BE degree from the Department of Computer Science, Harbin University of Science and Technology, Harbin, China, in 2012. He is currently working toward the master’s degree in the School of Computer Science, Beijing Institute of Technology, Beijing, China. His research interests include sensor networking, algorithm design, and mobile computing.



Chao Zhang received the PhD degree in civil engineering from the University of Alabama in 2010. He is currently working toward the PhD degree in computer science at the University of North Carolina at Charlotte from 2012. His current research interests include algorithm design and analysis of three-dimensional wireless sensor networks and network routing protocols of mobile social networks.

Yu Wang received the BEng and MEng degrees in computer science from Tsinghua University, Tsinghua, China, in 1998 and 2000, respectively, and the PhD degree in computer science from Illinois Institute of Technology, in 2004. He is currently an associate professor of computer science at the University of North Carolina at Charlotte. His current research interests include wireless networks, ad hoc and sensor networks, delay tolerant networks, mobile computing, complex network, and algorithm design. He has published over 100 refereed papers. He has served as general chair, program chair, program committee member, etc. for several international conferences (such as the IEEE International Performance Computing and Communications Conference, the IEEE Conference on Computer Communications, the ACM International Symposium on Mobile Ad Hoc Networking and Computing, the IEEE Global Telecommunications Conference, the IEEE International Conference on Communications, the IEEE International Conference on Mobile Ad hoc and Sensor Systems, etc.). He received the Ralph E. Powe Junior Faculty Enhancement Awards from Oak Ridge Associated Universities in 2006 and the Outstanding Faculty Research Award from College of Computing and Informatics at University of North Carolina at Charlotte in 2008. He received Best Paper Awards from multiple conferences, such as IEEE HICSS-35, the IEEE International Conference on Mobile Ad hoc and Sensor Systems 2013, and the IEEE International Performance Computing and Communications Conference 2013. He is a senior member of the ACM, IEEE, and IEEE Communications Society and currently chairs the Communication Chapter at the IEEE Charlotte Section.