Ad Hoc Networks 46 (2016) 48–60

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Spanning tree based topology control for data collecting in predictable delay-tolerant networks

Hongsheng Chen[a,b], Ke Shi[a,*], Chunhui Wu[b]

[a] College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[b] College of Computer Science and Technology, Hubei University of Science and Technology, Xianning 437005, China

ARTICLE INFO

Article history:
Received 16 June 2015
Revised 6 January 2016
Accepted 7 March 2016
Available online 4 April 2016

Keywords:
Predictable delay tolerant networks
Data collection
Topology control
Spanning tree

ABSTRACT

In predictable delay tolerant networks (PDTNs), the network topology is known a priori or can be predicted over time such as vehicular networks based on public buses or trains and space planet network. Previous DTN research mainly focuses on routing and data access. However, data collecting are used widely in PDTNs and it is very important in practical application, how to maintain the dynamic topology of this type of PDTNs becomes crucial. In this paper, a spanning tree (ST) based topology control method for data collecting in PDTNs is proposed. The PDTN is modeled as layered space-time directed graph which includes spatial, temporal and energy cost information, which can be simplified as reduced aggregated directed graph. The topology control problem is defined as constructing a ST that the total energy cost of the ST is minimized and the time delay threshold is satisfied. We propose three heuristic algorithms based on layered space-time directed graph and reduced aggregated directed graph to solve the defined problem, and compare them in terms of energy cost and time delay. Extensive simulation experiments demonstrate that the proposed algorithms can guarantee data transmission effectively, reduce the network energy consumption significantly, and shorten the time delay of data transmission.

1. Introduction

Topology control has been studied widely in wireless ad hoc and sensor networks, which can maintain network connectivity while minimizing the energy consumption and reducing radio interference. A lot of methods have been proposed, for example, graph-based methods, hierarchical structure based methods, social network analysis based methods, probabilistic-based methods, and etc.

Recently, with the widely application of wireless devices in challenging environment, such as battlefield reconnaissance, search-and-rescue, and satellite communications, delay/disruption tolerant networks (DTNs) [1,2] emerge as a good complement to the traditional wireless ad hoc and sensor networks. Due to the harsh surrounding and the mobility of devices, DTNs are not always connected, and the data is delivered in a long delay. Therefore, how to ensure the data can be transmitted to the destination is very important. Many routing protocols for DTNs have been proposed to transmit data from the source to the destination by utilizing opportunistically existing and time varying routing paths. Most of them use some kinds of redundancy and exploit routing opportunity greedily, which may lead to unnecessary resource consumption.

In general, topology control can avoid extra resource consumption in data transmitting. It is also true in DTNs, especially for predictable DTNs [3], in which the network topology is known a priori or can be predicted over time. In PDTNs, a cyclic time Ct exists, and in each Ct, the network topology changes are identical. Pocket switched networks based on human mobility [4,5], vehicular networks based on public buses or taxi cabs [6,7], mobile social networks [8,9], disaster-relief networks, and space communication networks [10–12] are all PDTNs. Instead of trying to using all possible transmitting opportunities, topology control mechanism can select certain transmitting opportunities to delivery data in such network to satisfy the reliability/delay requirement with lower resource consumption, especially energy consumption.

However, using the existing topology control mechanisms to DTNs directly is not feasible because the topology of DTNs is time evolving and the routing paths intermittently exist. As far as we know, only Huang et al. [13–15,35] studied topology control in PDTNs. They took the temporal characteristic into consideration and model PDTN as a directed space-time graph which includes both spatial and temporal information. Two greedy-based methods were proposed to find a routing sub-graph from the directed

* Corresponding author. Tel.: +86 18986639920.
E-mail addresses: 305464116@qq.com, keshi@mail.hust.edu.cn, keshi@hust.edu.cn (K. Shi), chenhs1981@163.com (C. Wu).

http://dx.doi.org/10.1016/j.adhoc.2016.03.009
1570-8705/

space-time graph. The goal is to ensure each pair of nodes can communicate under certain reliability with the least energy consumption.

Data collecting are the dominant applications in most PDTNs, which means many to one are the most frequently used communication patterns. In this paper, we focus on the topology control issues in these kinds of network. A tree based topology control mechanism is proposed to satisfy the performance requirement while minimizing the energy consumption. Our major contributions are summarized as follows:

(1) The PDTNs for data collecting are modeled as layered space-time directed graph, and then re-formed as reduced aggregated directed graph. The topology control issue is defined as finding the spanning tree (ST) of such a graph.
(2) Heuristic algorithms are proposed to find the spanning tree.
(3) The constructed ST can minimize the energy consumption of data collection with the certain delay constraint.

The rest of this paper is organized as follows. In Section 2, we summarize related works in topology control and PDTNs. In Section 3, tree based topology control problem and model are defined. Section 4 describes the details of the heuristic algorithms proposed in this paper. Section 5 presents the simulation results of the proposed algorithms. Finally, Section 6 concludes the paper and points out future research directions.

2. Related works

2.1. Topology control in ad hoc and sensor networks

Topology control is very important in ad hoc and sensor networks, because it can save energy and reduce the interference of the network. It can effectively prolong the life time of the network and improve the success rate of data transmission. Among many methods proposed for topology control, constructing Minimum Spanning Tree (MST) to reduce the edges of original topology is a simple and effective method. Many researchers have drawn a significant amount of research interests on it recently.

In literature [16,17,20,21,23], they are all explore MST method for topology control. Sun et al. [16] and Li et al. [20] proposed the algorithms that each node builds its local minimum spanning tree (MST) respectively based on the energy-aware weighted graph and keeps on-tree nodes that are one-hop away as its neighbors in case the network topology adjusted. Chee-Wei and Chen-Khong [17] presented an algorithm, iMST, attempting to maximize average channel utilization by reducing interference. This algorithm not only generates k-edge-connected networks, but also guarantees minimum link bandwidth. Li et al. [21] proposed a family of structures, namely, k-localized minimum spanning tree (LMSTk) for topology control and broadcasting in wireless ad hoc networks. They give an efficient localized method to construct LMSTk using only O(n) messages under the local-broadcast communication model, i.e., the signal sent by each node will be received by all nodes within the node's transmission range. Chen et al. [23] proposed an improved variable-range transmission power control algorithm based on minimum spanning tree algorithm (MST) for mobile ad hoc network, and it support node's mobility and solve asymmetric graph problem.

In addition to the above, there is also has some other methods for topology control in terms of other aspects. For example, Xing et al. [18] proposed a new topology control formulation for lossy WSNs. Their formulation captures the stochastic nature of lossy links and quantifies the worst-case path quality in a network. They develop a novel localized scheme called configurable topology control (CTC). The key feature of CTC is its capability of flexibly configuring the topology of a lossy WSN to achieve desired path quality bounds in a localized fashion. Furthermore, CTC can incorporate different control strategies (per-node/ per-link) and optimization criteria. Qing and Jie [19] proposed three different algorithms, binary search, Prim's MST and its extension, to solve power conservation issue for ad hoc wireless networks, which can find the minimum uniform transmission power of an ad hoc wireless network where each node uses the same transmission power. And network connectivity is maintained. Zhang et al. [22] proposed DSPT, a more efficient topology control algorithm than traditional MST-based algorithms for WSN. A single destination shortest path tree rooted from sink node was build to minimize the power consumption. Tapiwa et al. [32] present a new distributed topology control technique that enhances energy efficiency and reduces radio interference in wireless sensor networks. Each node in the network makes local decisions about its transmission power and the culmination of these local decisions produces a network topology that preserves global connectivity. Central to this topology control technique is the novel Smart Boundary YaoGabriel Graph (SBYaoGG) and optimizations to ensure that all links in the network are symmetric and energy efficient. Fadoua et al. [36] explore the energy-aware topology control in wireless ad hoc networks by formulating and solving the corresponding optimization problem. They propose an ILP formulation that minimizes the total transmission power needed by nodes to construct a topology that can meet Quality of Service (QoS) requirements between source and destination node pairs with less computational effort. Hakki et al. [38] present a distributed fault-tolerant topology control algorithm, called the Disjoint Path Vector (DPV), for heterogeneous wireless sensor networks composed of a large number of sensor nodes with limited energy and computing capability and several super nodes with unlimited energy resource.

All of these topology control methods deal with topology in wireless sensor networks or ad hoc networks. For the topology changes in ad hoc networks, most algorithms explore re-perform algorithm method. Fortunately, many construction algorithms are localized or distributed algorithms, therefore the update cost is not expensive. However, all of the methods assume that the underlying communication graph is fully connected and they do not consider the time delay and the network changes with time evolving.

2.2. Routing in DTNs

Recently, many researches focus on DTN routing, most of the routing protocols belong to two categories: social-based and prediction-based. In social-based routing protocols, they usually use the social properties as the determinants for routing. Zhu et al. [24] summarized the social properties in DTNs, and provide a survey of recent social-based DTN routing approaches. And they classified social-based DTN routing approaches as positive and negative routing according to the social characteristics. Cao et al. [25] proposed a multi-dimensional routing protocol (MDimension) for the human associated delay-tolerant networks which uses local information derived from multiple dimensions to identify a mobile node more accurately.

Prediction-based routing protocol is based on the prediction of network according to the historical or other known information. Specifically, in PDTNs, this method is used widely. In literature [4,5], they use human mobility for forwarding in pocket switched networks. Cao et al. [26] proposed geographic routing in DTNs, which estimates the movement range of the destination using its historical geographic information, promotes message replication reaching the edge of this range using a Reach Phase and spreading within this range using a Spread Phase. Tournoux et al. [27] proposed an adaptive variant of the spray-and-wait algorithm DA-SW (Density-Aware Spray-and-Wait) to exploit the accordion phenomenon and tune the dissemination effort to respond adequately

to the density variations. More specifically, the source sends more copies of a bundle when the topology is sparse and fewer copies when the topology becomes denser. Merugu et al. [10] proposed a new space-time routing framework for PDTNs leveraging the predictability of node motion. Specifically, they construct space-time routing tables where the next hop node is selected from the current as well as the future neighbors. They devise an algorithm to compute these space time routing tables to minimize the end-to-end message delivery delay. Their routing algorithm is based on a space-time graph model derived from the mobility of nodes. Cong and Jie [28] proposed an EMD-based probabilistic routing protocol, called routing in cyclic MobiSpace (RCM) using the expected minimum delay (EMD) as a new delivery probability metric in DTNs with repetitive but non-deterministic mobility. Li et al. [29] present LocalCom, a community-based epidemic forwarding scheme that efficiently detects the community structure using limited local information and improves the forwarding efficiency based on the community structure in DTNs. Cao et al. [30] proposed an active congestion control based routing algorithm that pushes the selected message before the congestion happens. In order to predict the future congestion situation, a corresponding estimation function is designed and their proposed algorithm works based on two asynchronous routing functions, which are scheduled according to the decision of estimation function. Giorgos et al. [31] present DS-TP's basic design principles and they evaluate its performance both theoretically and experimentally. They verify that practice conforms with theory and observe great performance boost, in terms of file delivery time between DTN-nodes, in case of DS-TP. Recently, Yue and Zhili [34] also proposed a Taxonomy, Survey and Challenges for routing in DTNs, in which they review the existing unicasting, multicasting and issue any casting issues of DTNs. Different from the perspective of the taxonomy which classifies the routing algorithms depending on the underlying mobility model, they classify the routing algorithms according to their design characteristics. They also identify their main challenges and open issues followed by an evaluation framework proposed for routing in DTNs. Zakhary et al. [37] proposes a novel fully distributed and collaborative k-anonymity protocol (LPAF) to protect users' location information and ensure better privacy while forwarding queries/replies to/from untrusted location-based service (LBS) over opportunistic mobile networks (OppMNets).

All these routing researches are focusing on delivering data from source to destination; topology control is not their main concerning. However, some methods finding better transmission chance may be helpful to topology control.

2.3. Topology control for PDTNs

At the time of writing this paper, only Huang et al. [13–15,35] studied topology control in PDTNs. In literature [14], a cost efficient topology design (CETD) for PDTNs is presented. They model such time-evolving network as a weighted space-time graph which includes both spatial and temporal information. The topology control problem in PDTNs is defined as constructing a sparse network that satisfies two goals, each pair nodes can connected and the total energy cost is minimized [13]. In literature [15], the unreliable links is considered. Through simulation, the efficiency of the proposed topology control methods is demonstrated. The sparse network they constructed to guarantee each pair of nodes can communicate may not suitable for data collecting applications used widely in PDTNs. In this paper we will investigate the topology control problem for data collection.

To our best knowledge, there are no previous results on topology control in data collecting environment of PDTNs. This paper is the first attempt to study topology control for PDTNs based on spanning tree (ST). We believe that topology can be controlled more wisely and efficiently if the network evolution over time is known.

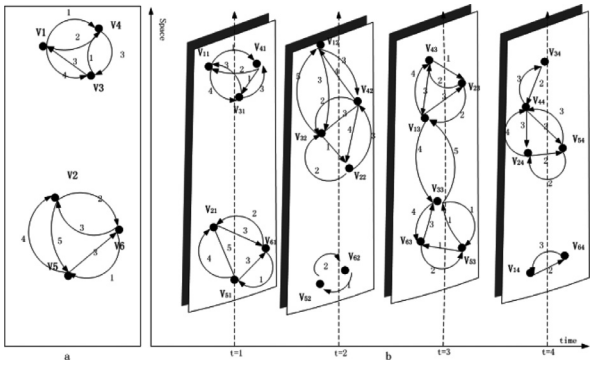

Fig. 1. Predictable DTNs topology changes over time (a) a snapshot of the network (b) time-varying topologies of the PDTNs (a sequence of snapshots).

3. Model and problem

In PDTNs, all the devices are mobile, and the communication between them is not continuous. So in some time, many devices can't direct communication with each other. Due to the devices are mobile over time, topology of PDTNs will changes dynamic over time (as shown in Fig. 1). Using the traditional static graph to represent the entire network topology cannot satisfy the practical need. Generally, the nodes in PDTNs have cyclic motion and contact patterns, and a common motion cycle Ct exists for all nodes. For example, the common motion cycle Ct for a public buses system is a day. To describe the dynamic and continuous topology changes of PDTNs, we can divide a cyclic time Ct into many time slots through the discretization method, then get the snapshot topology of each time slot, and finally combine this series of snapshot topologies together according to the order. In Fig. 1(a), the node V_i represents the mobile device, such as the bus with the WIFI running on the road, the edge E_i with a weight e_i represents the pair of nodes connected by it can communicate with the a certain energy cost indicated by the weight. When the nodes move, some nodes may lose connection, some may establish new connections, and the network topology changes dynamically. Fig. 1(b) describe time-varying topologies of the PDTNs, we use V_{ij} (i = 1,2,…,6,j = 1,2,…,T) represent the mobile device i in time slot j.

3.1. Layered space-time directed graph

Space-time graph model captures both the space and time dimensions of the network topology. The main idea is to construct a series of graph, where each graph corresponds to a discrete time interval (a time slot) in the life of the network. In this paper, we take the transmitting cost of each pair of nodes into consideration, and we model PDTNs as a layered space-time directed graph which is different from the previous plane space-time directed graph model that only includes one hop connections in one snapshot. Layered space-time directed graph includes multi-hop path between the pairs of nodes in one snapshot. It is the reason we call it "layered". The snapshot of each time slot likes a routing layer where multi-hop connection is possible. It increases the complexity of topology control, but also provides more choices of links for selection including efficient combinations of spatial and temporal links and multi-hops path of each snapshot. As shown in Fig. 1, each static graph is a snapshot of PDTNs observed at a time slot, where the weight of each edge represent the energy cost of data transferring between its connecting nodes in this period of time, and this value is decided by the node's transmitting power. In each graph, a lot of nodes are not connected,

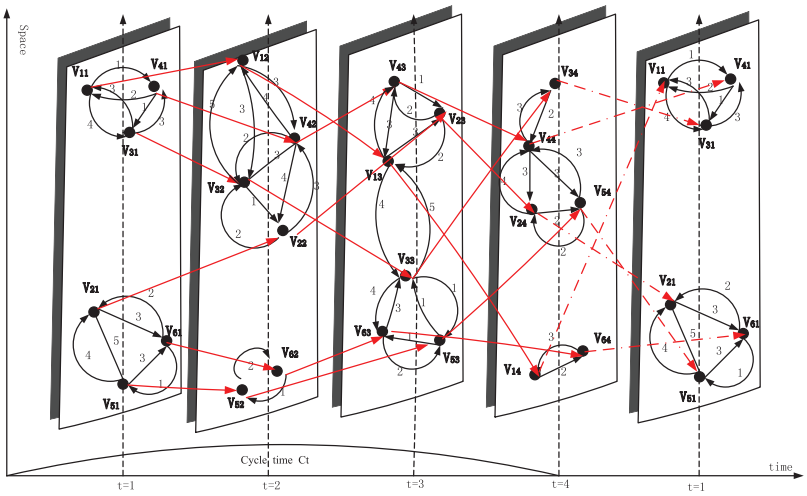

Fig. 2. Layered space-time directed graph that represents the network described by Fig. 1.

which indicates that the network is not complete connected in a certain time. This series of snapshot topologies is combined together according to the time order, which leads to as a layered space-time directed graph representing the dynamic predictable DTNs.

Assume that the cycle time Ct is divided into discrete and equal time slots, such as {1,...,T}. Let $V = \{V_1,...,V_n\}$ be the set of all individual nodes in the network (which represents the set of wireless devices). E^t represents the set of the edges of the network. Let $e^t = \{t = 1,...,T\}$ (in this paper we use the energy cost as the weight of the edge, in fact, the weight can represent the delay, connectivity probability and etc. according to practical application) be the set of energy cost of each edge for the corresponding directed graph in the t time slot. Let $G^t = (V^t, E^t, e^t)$ be a weighted directed graph representing the topology of the network at a time slot, and if directed edge $V_{it}V_{jt}$ exit, which indicate that the two nodes can communicate with each other with the energy cost equaling the weight of the $V_{it}V_{jt}$. So dynamic network will be modeled as a union of a series of weighted directed graphs $\{G^t | t = 1,...,T\}$.

Let G = (V, E, e) representing the layered space-time directed graph, including temporal and space information. Fig. 2 shows the layered space-time directed graph composed of a series of directed graphs. Because there are T time slots, and at the snapshot of each time slot there are n nodes, the node set can be represented by $V = \{V_{jt} | j = 1,...,n, t = 1,...,T\}$, which means there are n∗T nodes in the graph G. The edges from G^t are space edges representing the pair of nodes it connects can communication at a certain time slot (black line in Fig. 2). Besides space edges, time edges are introduced to connect the same node at different time slot (red line in Fig. 2). The weight of time connecting edge is set to zero since the same node in different slot don't need to deliver data but only inherit. The other edges' weights are the values of the corresponding G^t. Due to the network is cyclic, the data can be transmitted in the next cycle T, which means every node has the time link from the latest snapshot of the previous cycle to the first snapshot of the next cycle. These links marked as the dotted red lines in Fig. 2.

3.2. Reduced aggregated directed graph

Due to the layered space-time directed graph is very complex, we also model the PDTN as a reduced aggregated directed graph. In this model, between each pair of nodes, the space edge with the smallest energy cost is retained and the rest edges with bigger energy cost are removed as shown in Fig. 3. Assume $V = \{V_1,...,V_n\}$ be the set of all individual nodes in the network (which represents the set of wireless devices), the direct edges between each pair of nodes represents the two nodes can communicate, and the weight of each directed edge is represent by $e_c^t(t = 1,...,T)$, t represents the time slot and c represents the energy cost of the edge. The nodes V_{jt}, $t = 1, \ldots, T$, with the same j that represent the same node in the different time slots are aggregated to one node V_j. The graph shown in Fig. 3 is transformed into the reduced aggregated directed graph shown in Fig. 4. For example, e_2^3 is the weight of the edge from vertex V_2 to V_1 in Fig. 4, which means the edge from vertex V_2 to V_1 with the smallest energy cost exists in time slot 3, and the corresponding energy cost is 2.

3.3. Energy model

In PDTNs data transferring needs energy, energy cost is a very important problem. The goal of topology control is minimizing energy consumption, which make energy consumption model necessary. For the time edge (red edge), we set the energy cost is 0, because the data in the same node does not need to transport. For real data transmissions along the space edges, we adopt free space propagation model $Pt(d) \propto d^2$, where Pt is the transmitted signal power and d denotes the Euclidean distance between transmitter and receiver. To simplify the implementation, the energy cost of the space edge is given the appropriate value 1–5 according to the distance between each pair of nodes. We use $C(V_{it}V_{jt})$ to represent the energy cost of communication between node i and node j in time slot t. If node i and j are connected by time edge, the energy cost is 0. If they are connected by space edge, it is the edge weight. Since the space edge only connect the nodes in the same time slot, the two connected nodes share the same t. The total energy cost of the ST is represented by C(G) shown in the following formula:

$$C(G) = \sum_{t=1}^{T} \sum_{i=1,j=1}^{n} C(V_{it}V_{jt}) \times (V_{it}V_{jt} \in G \text{ and } V_{it}V_{jt} \text{ in the minimum spanning tree})$$

3.4. Problem statement

The aim of topology is to maintain network connectivity with the minimum energy cost. In this paper we mainly focus on many to one communication patterns that most frequently used in PDTNs. Topology control problem can be defined as a tree construction problem, which is finding the spanning tree of layered space-time directed graph and reduced aggregated directed graph model, such that (1) include all nodes of the network; (2) the total energy cost of the ST is minimized; and (3) satisfy the perfor-

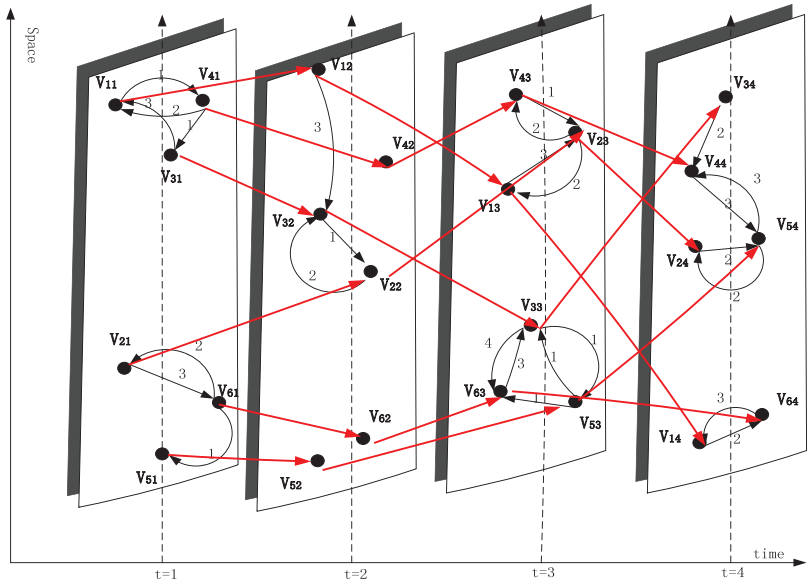

Fig. 3. Reduced directed graph that represents the network described by Fig. 2.

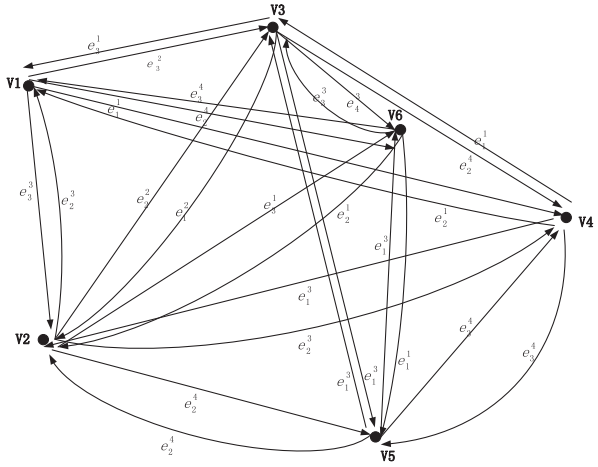

Fig. 4. Reduced aggregated directed graph that represents the network described by Fig. 2.

mance requirement, for example, the time delay that is considered in this paper. Given a directed graph G(V, E, e) and a time delay(td) threshold Λ_{td}, the problem is to find a ST of G(V, E, e) that satisfies,

minimize: C(G)

subject to: td $< \Lambda_{td}$

The precise time delay that one node transfers the data to the certain sink node depends on the exact data generating time. For example, when the single slice of snapshot is good connected (i.e. the spanning tree sits in a single snapshot), if the data generates before the time point of this snapshot, the exact time delay equals the snapshot's time minus data generating time; otherwise the actual data transferring will happen in the next cycle, and the delay equals the time of this snapshot in the next cycle minus data generating time. In general, the delay is decided by the transferring time caused by the final ST and the data generating time. Since data generating are random in many cases, to simplify the calculation of the delay, we use the total time edges of the deepest path of the final ST to represent the delay. And when this value is larger, the longer delay will be. We think it is reasonable to use this metric to represent the delay given the uncertain data generating time.

4. ST based topology control algorithm

4.1. Variant Kruskal algorithm base on layered space-time directed graph (VKASPG)

The basic idea of the first greedy algorithm is finding the least cost edges that satisfy the time delay threshold to ensure all nodes of the networks can deliver data to the sink node in time.

The VKASPG contains the following steps:

Step 1: All the edges are sorted in the ascending order according to their energy cost defined in the Layered space-time directed graph. And every vertex of the entire network is initiated as a connected component.

Step 2: Choosing the directed edge with the smallest energy cost, and merge the vertexes of this edge as one connected component.

Step 3: Choosing the directed edge with the second smallest energy cost. If the vertexes of this edge satisfy the certain conditions, they will be merged into the previous connected component. These conditions are:

(1) The vertexes are in different connected component.
(2) The vertexes haven't connected with the edges. We consider the vertexes representing the same node in the different time slot as one vertex, for example, V11, V12, V13 and V14 in different time slot represent the same node V1. If one of these vertexes has been connected with space edge, the others are not allowed to be connected.
(3) The vertex does not represent the sink node and has only one outgoing edge.

Step 4: If the above conditions are not satisfied, the directed edge with the ith (i = 3,…) smallest cost is selected and Step 3 is performed repeatedly until all vertexes have been reached by the selected edges. If one connected component contains all these vertexes, this connected component is the Spanning Tree (ST). Otherwise, the vertexes in the CC will be connected with the time edges to construct the ST.

Step 5: The time delay and the ratio of total cost are calculated. If the time delay is smaller than the threshold Λ_{td}, this algorithm ends with the established ST. Otherwise it will start again from Step 2 choosing the edges with the ith(i = 2,3,…) smallest cost.

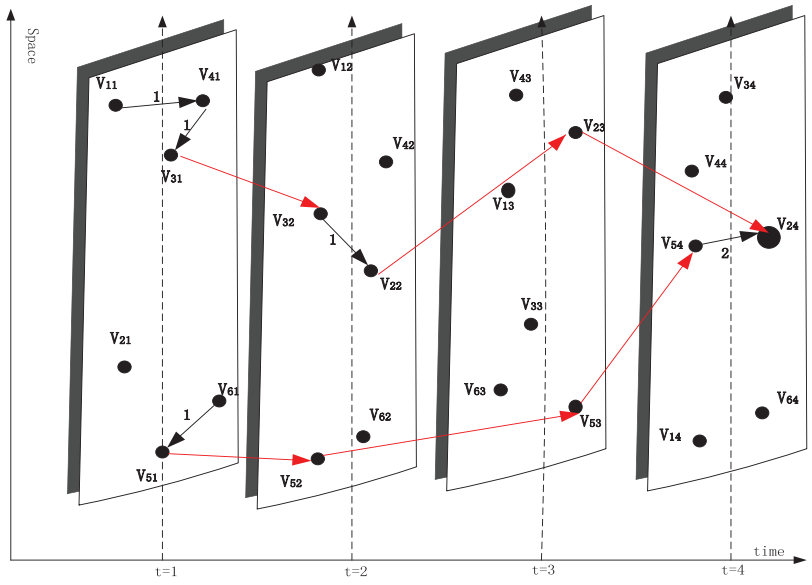

Fig. 5. ST found by variant Kruskal algorithm.

For the network described in Fig. 2, the final ST found by VKASPG is shown in Fig. 5. V_2 is the sink node. The final ST includes $V_{11}V_{41}$, $V_{41}V_{31}$, $V_{31}V_{32}$, $V_{32}V_{22}$, $V_{22}V_{23}$, $V_{23}V_{24}$, $V_{61}V_{51}$, $V_{51}V_{52}$, V_{52} V_{53}, $V_{53}V_{54}$ and $V_{54}V_{24}$. The edges $V_{11}V_{41}$, $V_{41}V_{31}$, $V_{32}V_{22}$, $V_{54}V_{24}$and $V_{61}V_{51}$ are space edges, and the others are time edges. The total energy cost is 6, and the time delay is 3 time slot.

The pseudo code of VKASPG algorithm is as follows, in which CC represents the connected component, N_e represents the number of the space edge, $V(E_i)$ represent the two vertexes of edge E_i, n represents the number of the vertex in one time slot, td represents the calculated time delay, and V_{sink} represents the sink node:

Input:
The original Layered space-time directed graph, including the energy cost of each edge, the sink node V_{sink}, time delay (td), and threshold Λ_{td}
Output:
The time delay (td) and Spanning Tree (ST);
1: Sort the edges of G in ascending order according to the energy cost;
2: k = 1;
3: j = k, initiated every vertex as a CC;
4: **for** $i = j$;j<N_e;j ++ **do**
5: Select the edges with the ith smallest cost, merge the two vertex of this edge as one CC;
6: **if** (the two vertexes are in different CC) and (the vertexes haven't connected with any space edges) and (The vertexes has only one outgoing edge and is not V*sink*) **then**
7: merge(CC,V(E_j));
8: **if** all vertexes are connected with n−1 space edges and in one CC **then**
9: ST is constructed;
10: **else**
11: **if** all vertexes are connected with n−1 space edges, but not in one CC **then**
12: Connected the different CC as one CC with the time edges to construct ST;
13: **end if**
14: **end if**
15: **end if**
16: **end for**
17: Calculating the time delay;
18: **if** td<Λ_{td} **then**
19: exit;
20: **else**
21: **for** $k = 2$;k<e; k ++ **do**
22: goto 3;
23: **end for**
24: **end if**

The time complexity of line 1 is $O(n^2\times T)$, the time complexity of line 4–16 is $O(n\times\log N_e)$, and the time complexity of line 17–23 is $O(n\times N_e\times\log N_e)$. So the time complexity of the algorithm is $O(n^2\times T + n\times N_e\times\log N_e)$.

Theorem 1. *The time complexity of the above algorithm is $O(n^2\times T + n\times N_e\times\log N_e)$, in which n represents the number of the nodes in the network, T represents the number of the time slots, and N_e represents the number of the space edges of the network.*

4.2. Variant Prim algorithm base on layered space-time directed graph (VPASPG)

The basic idea of the second greedy algorithm is to find the minimum energy cost tree rooted at sink node that satisfy the time delay threshold by adding nodes. The VPASPG contains the following steps:

Step 1: The connected component (CC) is initiated as empty. The vertex representing the sink node in the last time slot is chosen and added to the CC.
Step 2: In the current time slot the incoming edge to the CC's vertex with the smallest energy cost is selected. The starting vertex of the selected edge is added to the CC if it is not in the CC.
Step 3: Choosing the incoming edge to the CC's vertexes with the smallest energy cost in the current time slot respectively. If the vertexes of the edge have existed in CC, the edge is re-selected. If there have the same node in the selected edges, the vertexes connected with the smallest energy cost path are selected and added to CC. Otherwise, the starting vertexes are added to CC. If CC still does not contain all the vertexes, the vertexes that are in the previous time slot and connected with CC's current vertexes through time edge are set as current vertexes, and Step 3 is repeated until all vertexes are in the CC.
Step 4: The time delay is determined based on the depth of constructed tree. If the time delay exceeds Λ_{td}, the above steps are repeated to find other tree until this constraint is satisfied.

For the network described in Fig. 2, the final ST found by VPASPG is shown in Fig. 6. V_2 is the sink node. As shown in the Fig. 6, the ST generated by VPASPG has the total energy cost 8, and the total time delay is 2 time slot. It includes $V_{12}V_{42}$, $V_{42}V_{43}$, $V_{43}V_{23}$, $V_{23}V_{24}$, $V_{62}V_{52}$,$V_{52}V_{53}$, $V_{33}V_{53}$, $V_{53}V_{54}$ and $V_{54}V_{24}$. The edges $V_{12}V_{42}$, $V_{43}V_{23}$, $V_{62}V_{52}$, $V_{33}V_{53}$ and $V_{54}V_{24}$ are space edges, and the others are time edges.

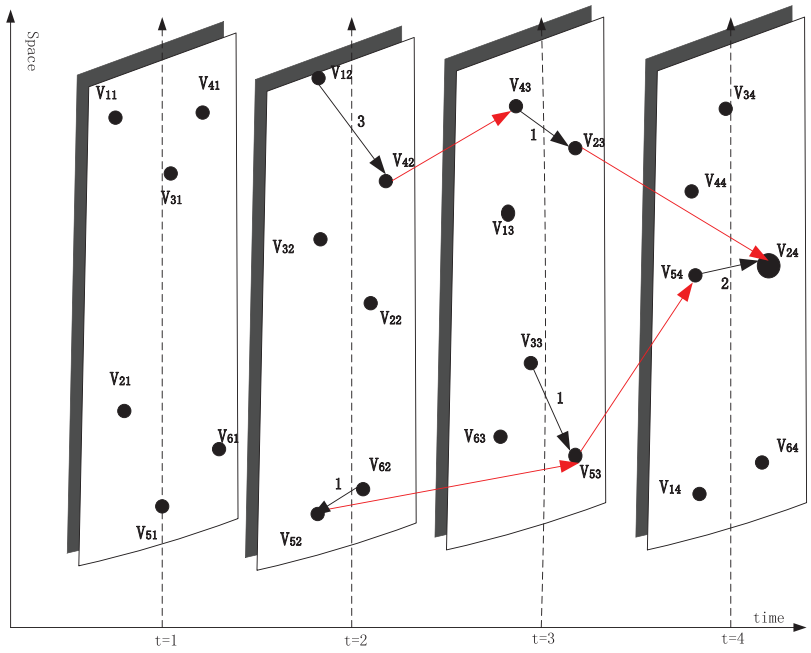

Fig. 6. ST found by variant Prim algorithm.

The pseudo code of VPASPG algorithm is as follows:

Input:
The original Layered space-time directed graph, including the energy cost of each edge, the sink node V_{sink}, time delay (td), and threshold Λ_{td}
Output:
The time delay (td) and Spanning Tree (ST);
1: j = 1;
2: CC = {V_{sink}};
3: i = j;
4: In the current time slot choosing the incoming edge to the CC's vertex with the *i*th smallest energy cost;
5: **if** the starting vertexes of the edge is not in CC **then**
6: add them to CC;
7: **end if**
8: **while** all vertexes are not in CC **do**
9: Choosing the incoming edge to the CC's vertex edge with the smallest energy cost in the current time slot respectively;
10: **if** the starting vertex of the edge has existed in CC **then**
11: reselect the edge;
12: **else**
13: **if** there have the same node in the selected edges **then**
14: choosing the vertexes connected with the smallest energy cost edge and add them to CC;
15: **else**
16: add them to CC;
17: **end if**
18: **end if**
19: selecting the vertexes that are in the previous time slot and connected with CC's current vertexes through time edge and set them as current vertexes;
20: **end while**
21: Calculating the time delay;
22: **if** td > Λ_{td} **then**
23: **for** *j* = 2;j < n;*j* ++ **do**
24: goto 2;
25: **end for**
26: **end if**

The time complexity of line 4 is $O((N_e+n)\times\log n)$, and for line 8–20 it is $O(T\times n\times(N_e+n)\times\log n)$, line 22–26 is $O(T\times n^2\times(N_e+n)\times\log n)$. So the time complexity of the algorithm is $O(T\times n^2\times(N_e+n)\times\log n)$.

Theorem 2. *The time complexity of the above algorithm is $O(T\times n^2\times(N_e+n)\times\log n)$, in which n represents the number of the node in the network, T represents the number of the time slot, and N_e represents the number of the space edges of the network.*

Table 1
Original matrix representing the reduced aggregated directed graph.

	V_1	V_2	V_3	V_4	V_5	V_6
V_1	–	e_3^3	e_3^2	e_1^4	∞	e_2^4
V_2	e_2^3	–	e_2^2	e_2^3	e_2^4	e_3^1
V_3	e_3^1	e_1^2	–	e_2^4	e_1^3	e_4^3
V_4	e_2^1	e_1^3	e_1^1	–	e_3^4	∞
V_5	∞	e_2^4	e_1^3	e_3^4	–	e_1^3
V_6	e_3^4	e_2^1	e_3^3	∞	e_1^1	–

4.3. ST construction algorithm base on the matrix of the reduced aggregated directed graph (MSTRAG)

MSTRAG method is based on the reduced aggregated directed graph that is a reduced version of layered space-time directed graph. Table 1 shows the matrix M that is used to describe the nodes' connections of the reduced aggregated directed graph in Fig. 4. The entry in the *i*th row and *j*th column represents the connection status of the corresponding pair of nodes. For example, for the graph shown in Fig. 4, the entry M_{25} is e_2^4, which means node V_5 can communicate with node V_2 in time slot 4 with the energy cost 2. The entry ∞ in Table 1 represents there is no one hop path between the corresponding pair of nodes. In this case, Dijkstra's algorithm is used to find a multi-hop transmission path and the entry would be the energy cost of this found path as shown in Table 2. For example, $V_{12}V_{32} \rightarrow V_{32}V_{33} \rightarrow V_{33}V_{53}(4)$ is a new entry value representing the connection between V_1 and V_5, which represent V_1 can communicate with V_5 through the path $V_{12}V_{32} \rightarrow V_{32}V_{33} \rightarrow V_{33}V_{53}$, and the total energy cost of the path is 4. $V_{32}V_{33}$ is time edge, $V_{12}V_{32}$ and $V_{33}V_{53}$ are space edges.

The constructed ST includes space edges and time edges, in which space edges are added to the tree according to the entries value, and the time edges are added for connect the same node in different time slots. The weight of the space edge of ST is the energy cost inherited from the matrix, and the weight of the time edge is 0.

Table 2
The matrix including multi-hop paths.

	V_1	V_2	V_3	V_4	V_5	V_6
V_1	–	e_3^3	e_3^2	e_1^1	$V_{12}V_{32}\rightarrow V_{32}V_{33}\rightarrow V_{33}V_{53}(4)$	e_2^4
V_2	e_2^3	–	e_2^2	e_2^3	e_2^4	e_3^1
V_3	e_3^1	e_1^2	–	e_2^4	e_1^3	e_4^3
V_4	e_2^1	e_1^3	e_1^1	–	e_3^4	$V_{41}V_{11}\rightarrow V_{11}V_{12}\rightarrow V_{12}V_{13}\rightarrow V_{13}V_{14}\rightarrow V_{14}V_{64}(4)$
V_5	$V_{53}V_{63}\rightarrow V_{63}V_{64}\rightarrow V_{64}V_{14}(4)$	e_2^4	e_1^3	e_3^4	–	e_1^3
V_6	e_3^4	e_2^1	e_3^3	$V_{61}V_{51}\rightarrow V_{51}V_{52}\rightarrow V_{52}V_{53}\rightarrow V_{53}V_{54}\rightarrow V_{54}V_{44}(4)$	e_1^1	–

We also prove the correctness of this algorithm.

Proof. The formed reduced aggregated directed graph includes the single-hop connection information of each pair of nodes, thus the matrix M of the reduced aggregated directed graph only has the single-hop edges. If all nodes can connect the sink node through the single-hop edges of the matrix M, a ST is constructed. For example, in Table 1 of my paper, all nodes can connect node V_2, a ST can be construct as shown in Fig. 7. If ST cannot be formed through the matrix M with only single-hop information, we can construct multi-hop matrix N through the single-hop matrix M, generally ST will be formed through N.

The MSTRAG contains the following steps:

Step 1: The matrix of the reduced aggregated directed graph is established as shown in Table 1.
Step 2: Finding the aggregation nodes (all the nodes in the network can deliver data to it) from the matrix. In Table 1, V_2 and V_3 are aggregation nodes, because all the nodes can communicate with V_2 or V_3. If one of these nodes is the sink node, ST is constructed. The edges of ST include space edges and time edges. The weight of the space edge is the energy cost inherited from the matrix, the weight of the time edge is 0.
Step 3: If the sink node is not found in the original matrix M that only includes single-hop path, we use Dijkstra's algorithm to find multi-hop transmission paths and add them to the original matrix M to obtain a matrix as shown in Table 2. Then Step 2 is repeated to get a ST.
Step 4: Calculating the time delay and the ratio of total cost, if the time delay calculating from the constructed ST is smaller than the threshold Λ_{td}, this algorithm is end, otherwise it will continue performing Step 2.

For the network described in Fig. 2, the final ST found by MSTRAG is shown in Fig. 7. V_2 is the sink node. The ST generated by MSTRAG has the total energy cost 9, and the total time delay is 3 time slot. It includes $V_{61}V_{21}$, $V_{21}V_{22}$, $V_{32}V_{22}$, $V_{22}V_{23}$, $V_{13}V_{23}$, $V_{43}V_{23}$, $V_{23}V_{24}$ and $V_{54}V_{24}$. The edges $V_{61}V_{21}$, $V_{32}V_{22}$, $V_{13}V_{23}$, $V_{43}V_{23}$, and $V_{54}V_{24}$ are space edges, and the others are time edges.

Details of the pseudo code of MSTRAG algorithm is as follows:

Input:
The Constructed matrix M of the reduced aggregated directed graph, including the energy cost of each edge, the sink node V_{sink}, time delay(td), and threshold Λ_{td}
Output:
The time delay (td) and Spanning Tree (ST);
1. Constructing matrix M of the reduced aggregated directed graph
2. Finding the sink node from M to construct ST
3. **If** there is no sink node in M
4. **Then** {Using Dijkstra's algorithm to construct matrix N with M;M = N; goto 2;}
5. End If
6. Calculating the time delay and the ratio of total cost
7. **If** (td > Λ_{td})
8. **Then** goto 2;
9. End If

The time complexity of line 1 is $O(n^2\times T)$,the time complexity of line 2 is $O(n^2)$,the time complexity of line 3–5 is $O(n^2)$, the time complexity of line 7–9 is $O((1+n+n^2)\times\log n)$. So the time complexity of the algorithm is $O(n^2\times T+(1+n+n^2)\times\log n)$.

Theorem 3. *The time complexity of the above algorithm is $O(n^2\times T+n^2\times\log n)$, in which n represents the number of the node in the network, T represents the number of the time slot.*

The following table compares these three algorithms in terms of energy cost and time delay. Table 3.

5. Performance evaluation and analysis

In this section, we will evaluate and analyze our proposed heuristic algorithms, namely, Variant Kruskal algorithm base on Layered space-time directed graph (VKASPG), Variant Prim algorithm base on Layered space-time directed graph (VPASPG), and ST construction algorithm base on the Matrix of the reduced aggregated directed graph (MSTRAG) by comparing their performance in terms of the energy cost and time delay. To the best of our knowledge, GrdLCP [35] is the latest algorithm proposed to address the topology control of PDTNs. We compared our algorithm with it. We used the network simulator NS-3, to evaluate our proposed algorithms, because the NS-3 is suitable for analyzing the energy cost of the network and the time delay. We implement all these algorithms in NS-3 environment. The underlying PDTNs are randomly generated from a random graph model or directly extracted from Cambridge Haggle tracing data [33].

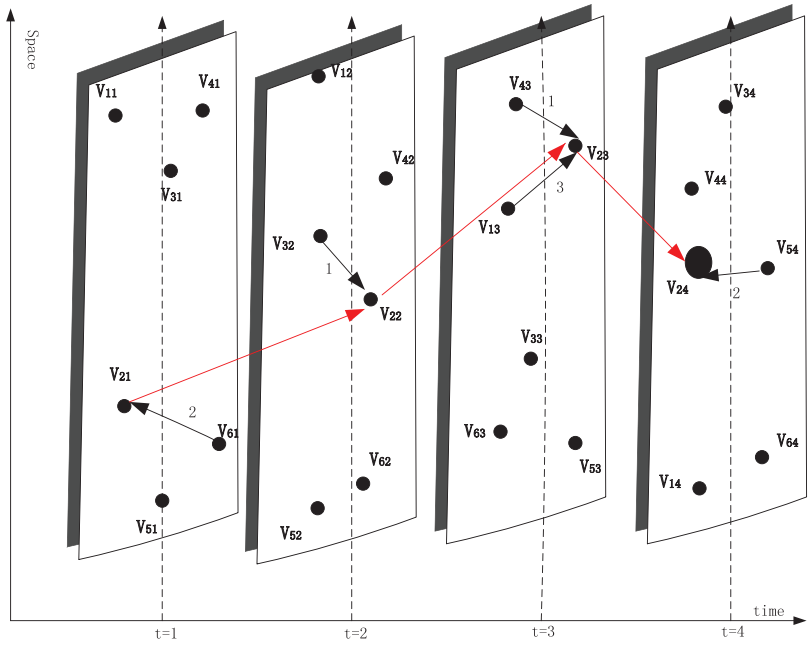

Fig. 7. ST found by MSTRAG.

Table 3
Comparison of the three algorithms in terms of energy cost and time delay.

Algorithm	Total energy cost	Time delay
VKASPG	6	3 time slots
VPASPG	8	2 time slots
MSTRAG	9	3 time slots

In simulation, we mainly consider the following performance metrics and parameters:

(1) The ratio of total cost: the performance metric is mainly used to evaluate whether the algorithm is good or bad by the energy efficiency, the smaller the total energy cost, the algorithm is more optimal.
The ratio of total cost = total energy cost value/the whole network energy value.
(2) Link density (ρ): It refers to the sparse/dense degree of the network topology.
Link density = the number of links of current topology/ the number of links of the corresponding complete graph of the current topology.
(3) Time delay: It refers to the time from data generation to data delivering. In this paper the time delay can be obtained from calculating the total time edges of the deepest path of the final ST.
(4) Time delay threshold Λ_{td}: It is a fixed value set up according to the practical application.

5.1. Simulations on random networks

We use the layered space-time directed graph and reduced aggregated directed graph constructed by a series of static weighted directed graphs to represent the predictable DTN. It has 10 nodes and spreading over 9 time slots. The static graph of each time slot is randomly generated by the classical random graph generator. The distribution of nodes is different for each run. The cost of each space edge is randomly chosen from 1 to 5, and the cost of the time edge is 0. For all the simulations, we repeat the experiment for 100,000 times and report the average values of the metric.

In our simulation, due to the variance of the total X is unknown. So we use sample variance S^2 instead the σ^2. The following formula is known:

$$T = \frac{\bar{X} - \mu}{\sqrt{S^2/n}} \sim t(n-1)$$

If given a certain α, set the following formular:

$$P\left\{\left|\frac{\bar{X} - \mu}{\sqrt{S^2/n}}\right| \le t_{\frac{\alpha}{2}}(n-1)\right\} = 1 - \alpha$$

Then check the distribution table t, obtain the value of $t_{\frac{\alpha}{2}}(n-1)$.

$$P\left\{\bar{X} - \frac{S}{\sqrt{n}} t_{\frac{\alpha}{2}}(n-1) \le \mu \le \bar{X} + t_{\frac{\alpha}{2}}(n-1)\right\} = 1 - \alpha$$

Thus the confidence interval of μ is:

$$\left[\bar{X} - \frac{S}{\sqrt{n}} t_{\frac{\alpha}{2}}(n-1), \bar{X} + \frac{S}{\sqrt{n}} t_{\frac{\alpha}{2}}(n-1)\right]$$

We use above theory to ensure the 95% confidence interval for our simulation. For example, we use 40 samples of the ratio of the total cost in VPASPG algorithm while the network's link density is 0.1. Through calculating, $\bar{X} = 0.09$, $S^2 = 0.024^2$, set $\alpha = 0.05$. First, check the distribution table:

$$t_{\frac{0.05}{2}}(n-1) = t_{0.025}(39) = 2.207$$

Then according to above formulas, the confidence interval of μ is obtained which is [0.08, 163, 0.09, 837]. We repeat the experiment for 100,000 times, and compared the results with μ. If it is in the confidence interval [0.08, 163, 0.09, 837], it will be retained, otherwise it will be discarded. Finally, the average values of the retained results are our results. All of our experiment firstly calculating the confidence interval with $\alpha = 0.05$ through above method. Thus our experiment can ensure 95% confidence interval.

Figs. 8 and 9 depict the impact of link density ρ to the ratio of total cost when the time delay threshold Λ_{td} is 10 and 9. As can be seen from the figures, our three algorithms are always better than GrdLCP, especially when the link density is smaller. That's

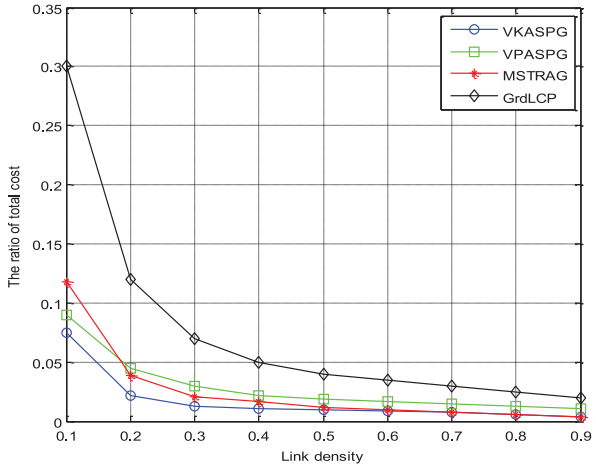

Fig. 8. Ratio of total cost of different link density ρ while $\Lambda_{td} = 10$.

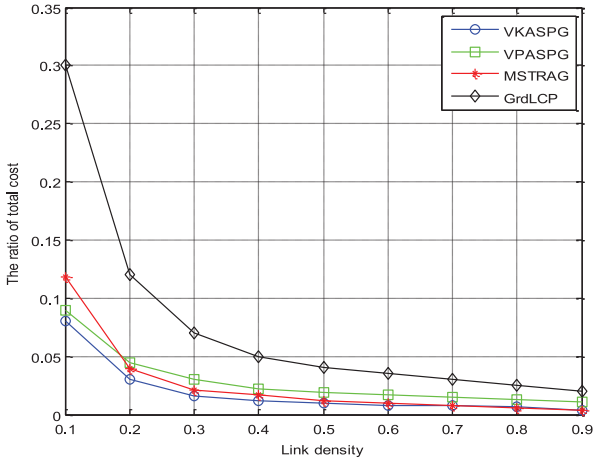

Fig. 9. Ratio of total cost of different link density ρ while $\Lambda_{td} = 9$.

because their constructed graph only considers single hop path of each pair of nodes in each time slot. Our Layered space-time directed graph not only considers single hop path, but also includes multi-hop path in each time slot. The tendency of the ratio of total cost of these three algorithms proposed in this paper is very similar when threshold Λ_{td} is 10 and 9, the ratio of total cost decreases with ρ increasing. Even when ρ is 0.1, the ratio of total cost is in the range of 0.073–0.114, which indicates that even in the sparse network especially in the challenge environment our algorithms could save more energy. VKASPG algorithm is always better than the other two algorithms. MSTRAG is better than VPASPG when ρ is larger than 0.1. The ratio of total cost of algorithm VKASPG and MSTRAG is almost same with ρ larger than 0.6, and the value is as low as 0.003. That is because the methods proposed in this paper are constructing a tree for topology control.

Figs. 10 and 11 depict the impact of link density ρ to the time delay when the time delay threshold Λ_{td} is 10 and 9. As can be seen from the figures, the tendency of the time delay of three algorithms proposed in this paper is very similar when threshold Λ_{td} is 10 and 9, the time delay decreases with ρ increasing. The only exception is MSTRAG get an increasing delay while ρ increases from 0.1 to 0.2. Even when ρ is 0.1, the time delay is in the range of 7–9, which indicates that even in the sparse network especially in the challenge environment our algorithm could transmit all the data generated before this cycle starts to the sink in this cycle time. VPASPG algorithm is always better than the other two algorithms except when ρ is 0.1. MSTRAG is the most stable

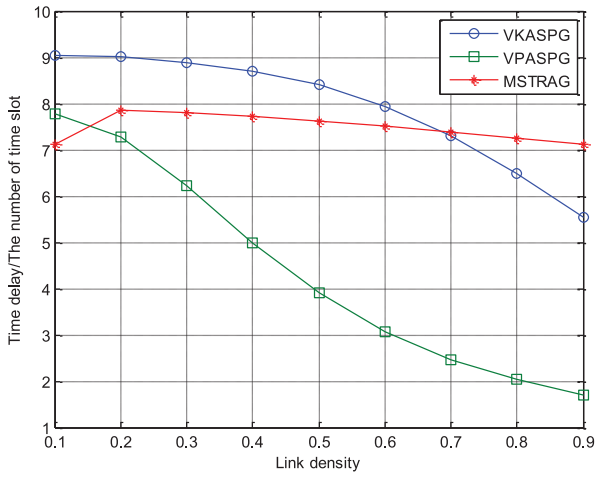

Fig. 10. Time delay of different link density ρ while $\Lambda_{td} = 10$.

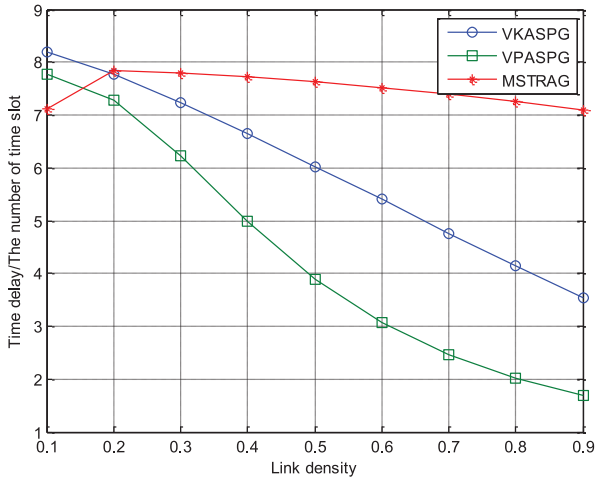

Fig. 11. Time delay of different link density ρ while $\Lambda_{td} = 9$.

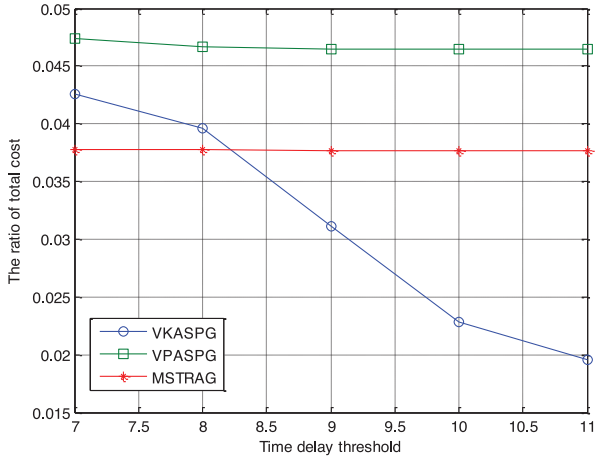

Fig. 12. Ratio of total cost of different Λ_{td} while link density $\rho = 0.2$.

algorithm of the three algorithms, the time delay always maintain between 7 and 8 with different ρ and Λ_{td}. The time delay can be lower than 2 when ρ is 0.9, which is very small.

The following Fig. 12 is the ratio of total cost of the above three algorithms VKASPG, VPASPG and MSTRAG with the time delay threshold Λ_{td} from 7 to 11 while ρ is 0.2. As can be seen from the figure, algorithms VPASPG and MSTRAG always maintain stable

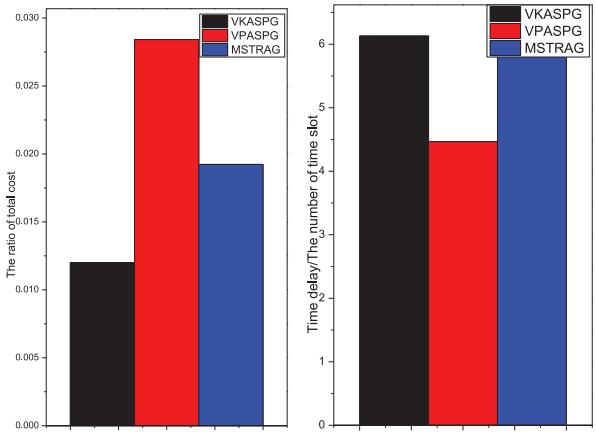

Fig. 13. Simulations results of VKASPG, VPASPG and MSTRAG on the network from the Cambridge Haggle [33] tracing data in the time period from 40,000 s to 310,000 s.

with the Λ_{td} increasing, the ratio of total cost of MSTRAG is lower than VPASPG, and the time delay of the VPASPG is lowest with link density increasing. However, the ratio of total cost of VKASPG decreasing with Λ_{td} increasing. Thus, if the required time delay is large, we can use the VKASPG, otherwise VPASPG will be used. MSTRAG is simple and easy to implement, and the energy cost and time delay are also moderate in these three algorithms. In addition, MSTRAG is very stable.

5.2. Simulations on tracing data

In this paper, we use the data sets from Cambridge Haggle data [33]. In this data set, connections among 78 mobile iMote Bluetooth nodes carried by researchers and additional 20 stationary nodes are recorded over four days during IEEE Infocom 2006. In our simulation, we only consider the 78 mobile nodes. Firstly, we consider the time period from 40,000 s to 310,000 s in the tracing data. The cycle time of constructed network almost covers the whole conference duration. The simulation results represent the average performance. Secondly, we consider the special time period from 40,000 s to 54,400 s. In this period, the conference sessions were in progress, more people were in the conference location, which makes the contact occurred more frequently and leads to a higher link density of constructed network. At last, we consider the period from 86,000 s to 122,000 s, when the conference sessions were suspended. Therefore, less people were in the conference location and the link density of constructed network is lower. We respectively divide the above three time periods into 9 time slots. In each time slot, if there is a contact trace overlapping with this slot, we add a spatial edge between the two corresponding nodes in G^t. Costs are randomly generated from 1–5 for spatial edge. We repeat each round of simulation for 100,000 times, the average link densities of these three corresponding networks are respectively about 0.4, 0.85 and 0.1. The simulations results are given in Figs. 13–15, the same conclusions can be drawn: 1) These three algorithms can reduce the cost remarkably (more than 98.8%); 2) VPASPG causes the largest cost among three methods, while VKASPG has the best performance in practice; 3)The time delay of VPASPG is the smallest, even the biggest time delay is about 6, which is lower than the cycle time Ct. The data generated before that starting point of a cycle can be delivered to the destination in a cycle Ct.

From Fig. 13 we can also see the results from the real world trace are better than the results from the random generated data. As shown in Figs. 9 and 11, the ratio of the total cost of VKASPG is 0.02 and the time delay of VPASPG is 5 when the link density is

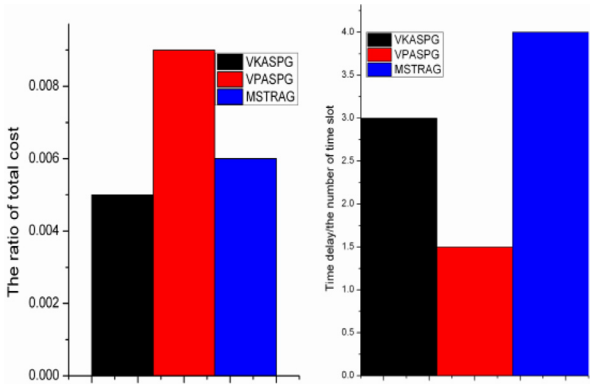

Fig. 14. Simulations results of VKASPG, VPASPG and MSTRAG in the time period from 40,000 s to 54,400 s.

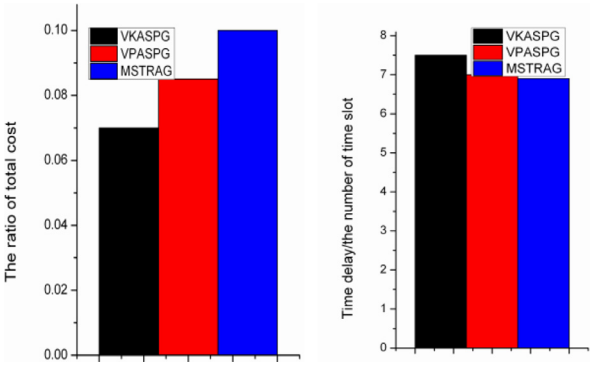

Fig. 15. Simulations results of VKASPG, VPASPG and MSTRAG in the time period from 86,000 s to 122,000 s.

0.4. Here the ratio of the total cost of VKASPG is 0.014 and the time delay of VPASPG is 4.5. The reason is that the network constructed based on the real world trace does not have the uniform link distribution like the network constructed based on randomly generated data, which helps the data forwarding.

When link density is higher, the above conclusion is still hold. As shown in Fig. 14, when the link density is 0.85, compared with the network based on randomly generated data, the ratio of the total cost is lower and the time delay is shorter. Since the contacts happen more frequently and the link density is higher, the performance is also better than the network constructed based on the real world trace with contacts happening less frequently.

When the contacts become rare and link density decreases, the performance become worse, both the ratio of the total cost and the time delay increase, as shown in Fig. 15 when the link density equals 0.1. However, the performance is still better than the network constructed based on randomly generated network. We also find the time delay is smaller than the threshold, which indicates that even in the sparse practical network our algorithms still can deliver the data generated before that starting point of a cycle to the destination in a cycle Ct after eliminating unnecessary links.

From above simulations and results, we can see tree based topology control methods proposed in this paper is very efficient in terms of energy cost and time delay. These three algorithms all have their advantage and disadvantage. VPASPG has the highest time complexity, but achieve the lowest time delay. VPASPG is more suitable for the delay sensitive applications. The energy cost of VKASPG is the smallest; it can save more energy when the application can tolerate longer delay. MSTRAG is simple and easy to implement, and the energy cost and time delay are moderate. And MSTRAG performs very stable under the different circumstances.

6. Conclusion

In this paper, we model PDTN as a layered space-time weighted directed graph and reduced aggregated directed graph and define the topology control problem as finding a ST from the graph model to minimize energy cost. We propose three heuristic topology control algorithm, VKASPG, VPASPG and MSTRAG to solve this problem.

The next step is to consider the following problems: (1) Improving the efficiency of the proposed heuristic topology control algorithms; (2) Solving the topology control problem for unpredictable DTNs;(3) Devising a method that can get topology information accurately, which is difficult to obtain in a real environment; (4) Dealing with topology control of PDTNs when some node may fail in the movement; (5) Balancing energy cost of the network.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant no. 51435009 and Dr. Start-up Foundation of Hubei University of Science and Technology no. BK1522.

References

[1] P. Juang, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with Zebranet, ACM SIGOPS Oper. Syst. Rev. 36 (5) (2002) 96–107.
[2] Y. Wang, H. Dang, H. Wu, A survey on analytic studies of delay-tolerant mobile sensor networks: research articles, Wirel. Commun. Mob. Comput. 7 (10) (2007) 1197–1208.
[3] Yifeng Shao, Jie Wu, Understanding the Tolerance of Dynamic Networks: A Routing-Oriented Approach, in: Proceedings of International Conference on Distributed Computing Systems Workshops(ICDCS), 2008, pp. 180–185.
[4] Pan Hui, J. Crowcroft, Predictability of human mobility and its impact on forwarding, in: Proceedings of International Conference on Communications and Networking in China, 2008, pp. 543–547.
[5] Pan Hui, J. Crowcroft, E. Yoneki, BUBBLE rap: social-based forwarding in delay–tolerant networks, IEEE Trans. Mob. Comput. 10 (11) (2011) 1576–1589.
[6] X. Zhang, J. Kurose, B.N. Levine, D. Towsley, H. Zhang, Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing, in: Proceedings of ACM MobiCom, 2007.
[7] M. Piorkowski, N. Sarafijanovoc-Djukic, M. Grossglauser, A Parsimonious Model of Mobile Partitioned Networks with Clustering, in: Proceedings of First Int'l Conference on Communication Systems and Networks and Workshops (COMSNETS '09), 2009.
[8] Hongxian Sun, Chuan Wu, Epidemic forwarding in mobile social networks, in: Proceedings of IEEE International Conference on Communications (ICC), 2012, pp. 1421–1425.
[9] F. Hao, G. Min, M. Lin, C. Luo, L. Yang, MobiFuzzyTrust: an efficient fuzzy trust inference mechanism in mobile social networks, IEEE Trans. Parallel Distrib. Syst. 25 (11) (2013) 2944–2955.
[10] Shashidhar Merugu, Mostafa H. (Mostafa Hamed) Ammar, Ellen W. Zegura, Routing in Space and Time in Networks with Predictable Mobility. http://hdl.handle.net/1853/6492, 2004 CC Technical Report; GIT-CC-04-07.
[11] T. De Cola, Mario Marchese, A. Raviola, Power and Bandwidth Effective Data Communications in Disaster Relief Operations through a Satellite-Based Disruption Tolerant Network Paradigm, in: Proceedings of IEEE International Conference on Communications (ICC), 2008, pp. 1876–1880.
[12] S. Burleigh, A. Hooke, Delay-tolerant networking: an approach to interplanetary internet, IEEE Comm. Mag. 41 (6) (2003) 128–136.
[13] Minsu Huang, Siyuan Chen, Ying Zhu, Bin Xu, Yu Wang, Topology Control for Time-Evolving and Predictable Delay-Tolerant Networks, in: Proceedings of IEEE Mobile Ad-Hoc and Sensor Systems, 2011.
[14] Minsu Huang, Siyuan Chen, Ying Zhu, Bin Xu, Yu Wang, Cost-Efficient Topology Design Problem in Time-Evolving Delay-Tolerant Networks, in: Proceedings of IEEE Globecom, 2010.
[15] Minsu Huang, Siyuan Chen, Fan Li, Yu Wang, Topology Design in Time-Evolving Delay-Tolerant Networks with Unreliable Links, in: Proceedings of IEEE Globecom, 2012.
[16] Ruozi Sun, Jian Yuan, Ilsun You, Xiuming Shan, Yong Ren, Energy-aware weighted graph based dynamic topology control algorithm, Simul. Model. Pract. Theory 19 (2011) 1773–1781.
[17] Chee-Wei Ang, Chen-Khong Tham, A bandwidth-guaranteed topology control algorithm for TDMA-based ad hoc networks with sectorized antennas, Comput. Netw. 52 (2008) 1675–1692.
[18] Guoliang Xing, Chenyang Lu, Xiaohua Jia, Robert Pless, Localized and configurable topology control in lossy wireless sensor networks, Ad Hoc Netw. 11 (2013) 1345–1358.
[19] Qing Dai, Jie Wu, Computation of Minimal Uniform Transmission Power in Ad Hoc Wireless Networks, in: Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03), 2003.
[20] Ning Li, Jennifer C. Hou, Lui Sha, Design and Analysis of an MST-Based Topology Control Algorithm, in: Proceedings of INFCOM, 3, 2003, pp. 1702–1712.
[21] Xiang-Yang Li, Yu Wang, Wen-Zhan Song, Applications of k-Local MST for topology control and broadcasting in wireless Ad Hoc networks 15 (12) (2004) 1057–1069.
[22] Zhang Kewang, Zhang Deyun, Jiang Weihua, A SPT Based Topology Control Algorithm for Wireless Sensor Network, in: Proceedings of Third International Conference on Semantics, Knowledge and Grid, 2007, pp. 282–285.
[23] Chunjie Chen, Demin Li, Chao Jin, Jiacun Wang, An Efficient Variable-range Transmission Power Control Algorithm for Mobile Ad Hoc Networks, in: Proceedings of Wireless Communications, Networking and Mobile Computing(WiCom '09), 2009.
[24] Ying Zhu, Bin Xu, Xinghua Shi, Yu Wang, A survey of social-based routing in delay tolerant networks: positive and negative social effects, IEEE Commun. Surv. Tutor. 15 (1) (2013) 387–401.
[25] Longxiang Cao, Ming Li, Alessio Bonti, Wanlei Zhou, Shui Yu, Multi-dimensional routing protocol in human associated delay-tolerant networks, IEEE Trans. Mob. Comput. 12 (11) (2013) 2132–2144.
[26] Y. Cao, Z. Sun, M. Riaz, Reach-and-spread: a historical geographic routing for delay/disruption tolerant networks, IET Netw. 1 (3) (2012) 163–170.
[27] Pierre-Ugo Tournoux, Je´re´mie Leguay, Farid Benbadis, John Whitbeck, Vania Conan, Marcelo Dias de Amorim, Density-aware routing in highly dynamic DTNS: the Rollernet case, IEEE Trans. Mob. Comput. 10 (12) (2011) 1755–1768.
[28] Cong Liu, Jie Wu, Practical routing in a cyclic MobiSpace, IEEE/ACM Trans. Netw. 19 (2) (2011) 369–382.
[29] Feng Li, Jie Wu, LocalCom: A Community-based Epidemic Forwarding Scheme in Disruption-tolerant Networks, in: Proceedings of Sensor, Mesh and Ad Hoc Communications and Networks(SECON '09), 2009.
[30] Yue Cao, Haitham Cruickshank, Zhili Sun, Active Congestion Control Based Routing for Opportunistic Delay Tolerant Networks, in: Proceedings of Vehicular Technology Conference (VTC Spring), 2011.
[31] Giorgos Papastergiou, Ioannis Psaras, Vassilis Tsaoussidis, Deep-space transport protocol: a novel transport scheme for Space DTNs, Comput. Commun. 32 (2009) 1757–1767.
[32] Tapiwa M. Chiwewe, Gerhard P. Hancke, A distributed topology control technique for low interference and energy efficiency in wireless sensor networks, IEEE Trans. Ind. Inform. 8 (1) (2012) 11–19.
[33] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, A. Chaintreau, CRAWDAD data set cambridge/haggle (v. 2006-09-15),. http://crawdad.cs.dartmouth.edu/cambridge/haggle, Sept. 2006 Downloaded from.
[34] Yue Cao, Zhili Sun, Routing in delay/disruption tolerant networks: a survey, taxonomy and challenges, IEEE Commun. Surv. Tutor. 15 (2) (2013) 654–677.
[35] Minsu Huang, Siyuan Chen, Ying Zhu, Yu Wang, Topology control for time-evolving and predictable delay-tolerant networks, IEEE Trans. Comput. 62 (11) (2013) 2308–2321.
[36] Fadoua Yakinea, Abdellah Idrissia, Energy-aware topology control and Qos routing in ad-hoc networks, Procedia Comput. Sci. 56 (2015) 309–316.
[37] S. Zakhary, M. Radenkovic, A. Benslimane, Efficient location privacy-aware forwarding in opportunistic mobile networks, IEEE Trans. Veh. Technol. 63 (2) (2014) 893–906.
[38] Hakki Bagci, Ibrahim Korpeoglu, Adnan Yazıcı, A distributed fault-tolerant topology control algorithm for heterogeneous wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 26 (4) (2015) 914–923.

Hongsheng Chen: Born in 1981, Ph.D. candidate, lecturer, his research interests focus on wireless sensor networks.

Ke Shi: Born in 1973, Ph.D., Professor, his research interest's focus on wireless ad-hoc/sensor networks, embedded computing, intelligent control and distributed computing.