

Flow-Based Transmission Scheduling in Constrained Delay Tolerant Networks

Zhenguo Yang^{†,‡}, Liusheng Huang^{†,‡}, Mingjun Xiao^{†,‡} and Wang Liu^{†,‡}

[†]Department of Computer Science & Technology, University of Science & Technology of China, China

[‡]Suzhou Institute for Advanced Study, University of Science & Technology of China, China

E-mail: {zgyang@mail, lshuang@, xiaomj@, wangliu@mail}.ustc.edu.cn

Abstract—Routing is one of the most challenging problems in Delay-Tolerant Networks (DTNs) due to network partitioning and intermittent connectivity. Most existing protocols are based on unlimited bandwidth and buffer size. Previous protocols based on limited bandwidth or buffer size (e.g., MaxProp, RAPID, etc) only consider the scenario that each node has at most one contact opportunity during any time interval, and discuss the packet scheduling mechanisms for each separate contact to minimize the average delivery delay. However, in many applications of DTNs, more than one contact opportunities with the same source node, which are called as related contact opportunities, may arise in the same time interval. But previous scheduling algorithms cannot apply to such applications. Therefore, we propose a more general scheduling algorithm to optimize the delivery delay in constrained DTNs including the scenario that multiple related contact opportunities may arise in the same interval, called Flow-Based Transmission Scheduling (FBTS). We evaluate it on publicly available data sets against MED, MaxProp and RAPID. The results show that the delivery delay of FBTS is at least 35% shorter than that of existing works.

Index Terms—DTNs, routing, flow, scheduling

I. INTRODUCTION

Delay tolerant networks (DTNs) [1] are frequently-partitioned networks, due to high node mobility, low node density, and short radio range, etc. Opportunistic communication between nodes is the major characteristic of DTNs. Thus contemporaneous end-to-end paths may not exist in DTNs. Delay Tolerant Network Research Group (DTNRG) [2] has designed a special architecture to describe such networks, under which communication between pairwise nodes is called *contact* and packets are generally transferred in the manner of *store-carry-forward*. Because the emergence of contacts is *time-varying* and *uncertain*, researches on routing in DTNs become active and challenging.

So far, many DTN routing protocols are designed to optimize a special routing metric, such as delivery probability or delivery delay [3]–[9]. Most of these existing protocols are based on *unlimited bandwidth or buffer*

size, except for MaxProp [6] and RAPID [7]. Both of them are based on limited bandwidth, which are more realistic in real DTNs. In MaxProp and RAPID, packet scheduling mechanisms are employed so that when a contact emerges, the packets with highest delivery probability or shortest delivery delay would be prior transferred, the packets with low priority would be discarded finally due to the memory constraint or the living time exhausted. However, both of them only consider the scenario that each node usually has at most one contact opportunity during any time interval such as UMassDieselNet [10] bus system, but leave another scenario out of consideration. Under the ignored scenario, multiple related contact opportunities may come up in the same time interval, which is more common especially in DTNs based on social networks [11], [12]. Figure 1 shows an example of NUS [11], where students with communication equipments (e.g., motes, PDAs, etc) attend different classes in a day. The features of the scenario are as followings:

- It is a typical delay tolerant network due to the mobility of students (e.g., node *A* might encounter with different nodes in different class sessions).
- It is common that multiple related contact opportunities may arise in the same time interval due to the gregariousness of students (e.g., node *B* and *C* are both in the communication range of *A* in current Algorithm Class).

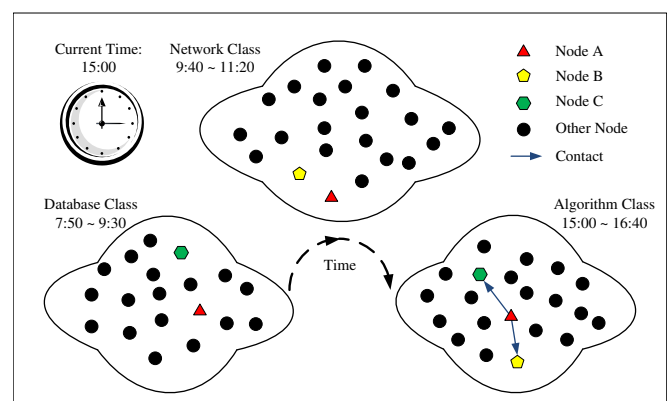


Figure 1. Students with communication equipments attend different classes in a day. They can communicate with each other when in the same class session. And current time is 15 : 00, when Algorithm Class begins.

This work is supported by the National Grand Fundamental Research 973 Program of China (Grant No.2011CB302905), the National Science and Technology Major Project under Grant No. 2011ZX03005-002, the National Natural Science Foundation of China (Grant No. 60803009), and the Natural Science Foundation of Jiangsu Province in China (Grant No.BK2009150).

TABLE I.
EXPECTED DELAY (ED) AND CORRESPONDING UTILITY OF RAPID
(δU) IN DIFFERENT NODES (HOUR)

	Node A		Node B		Node C	
	ED	δU	ED	δU	ED	δU
p_1	6	0	8	-2	5	1
p_2	9	0	7	2	6	3
p_3	7	0	6	1	7	0

We further present the insufficiency of previous scheduling mechanisms through above example. Students with communication equipments can form a constrained DTN, where nodes encounter with each other probabilistically (assuming the bandwidth is one packet during each class session). Node A, B and C are all in Algorithm Class at current time. Considering packet scheduling of node A, which has three packets p_1 , p_2 and p_3 to deliver, our goal is to minimize the average delivery delay of these packets. We assume that the expected delay for each packet in different nodes is known in Table I. The calculation of the expected delay will be introduced in Section 2. RAPID [7] protocol schedules packets for each separate contact in descending order of δU_i , which equals the corresponding expected delay in the sender minus that in the receiver (see Table I). Figure 2(a) shows the total expected delay of RAPID is 19 hours, when scheduling packets for node B first then node C. Figure 2(b) shows another result with a smaller expected delay, which can be obtained by our transmission scheduling algorithm. From this simple example, we draw that existing protocols cannot always optimize a routing metric in constrained DTNs.

In this paper, we focus on a more general *transmission scheduling* problem in constrained DTNs. In such a transmission scheduling, packets in the sender are distributed to multiple related contact opportunities, which may arise in the same time interval, to minimize the average delivery delay. Firstly, We formalize the problem into a *linear programming* problem. Secondly, we transform the linear programming problem to a *minimum cost maximum flow (mcmf)* problem. Thirdly, we propose our transmission scheduling algorithm Flow-Based Transmission Scheduling (FBTS). Moreover, we prove the correctness of FBTS and study the problem of starvation and congestion, as well as a distributed algorithm. We also provide the improvement on them. FBTS is finally evaluated on public trace.

Since the emergence of contacts in DTNs is uncertain, transmission scheduling in DTNs is different from packet scheduling in traditional networks, where packets are scheduled to certain link. The main contributions of this paper are:

- 1) propose a more general transmission scheduling problem in DTNs, where multiple related contact opportunities may arise in the same time interval.
- 2) design a novel flow-based transmission scheduling algorithm with improvement on starvation prevention and congestion avoidance, as well as local

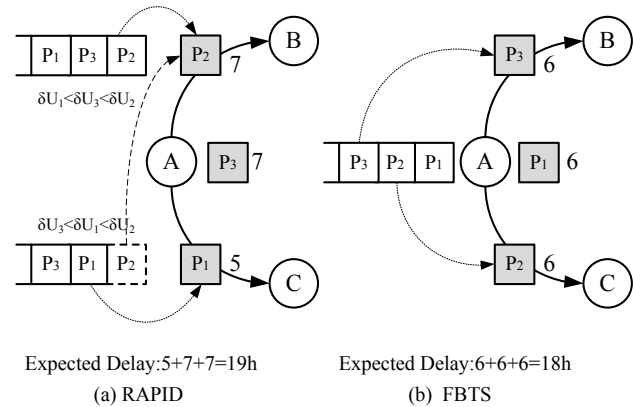


Figure 2. An example to show the insufficiency of previous scheduling

TABLE II.
SYMBOLS AND NOTATIONS OF PACKET k

Notations	Descriptions
$a(k)$	time since creation
$d(k)$	destination node
$h(k)$	current node
$q(k)$	queueing time in current node
$r(k)$	expected remaining time
$s(k)$	source node
$x(k)$	the node which has it at next time slot

estimation on expected delay.

- 3) evaluate FBTS on traces publicly available in the CRAWDAD archive [13].

The rest of this paper is organized as follows. Related Works are presented in Section 2. We provide the preliminaries including network modeling and problem formalization in Section 3. In Section 4, we propose our Flow-Based Transmission Scheduling algorithm, including network construction and transmission scheduling. The analysis and improvement are presented in Section 5, they includes correctness proof, starvation prevention, congestion avoidance and local estimation. FBTS is evaluated on NUS trace by comparing with some existing protocols in Section 5 and conclusion is presented in Section 6.

II. RELATED WORKS

In the past few years, many protocols are designed so as to minimize average delivery delay (or maximize delivery ratio). For example, Liu and Wu *et al.* proposed RCM [14] to solve the routing problem in cyclic mobispace with markov decision process and probability-based OPF [15] with optimal stopping rule. Gao *et al.* applied knapsack algorithm to solve the multicast routing problem in social network, SDM [16]. Yuan *et al.* employed a time-homogeneous semi-markov process model and proposed predict and relay (PER) [17] algorithm. Li *et al.* proposed a Social Selfishness Aware Routing (SSAR) algorithm [18] following the philosophy of design.

When considering the number of replicas, existing DTN routing protocols can be classified as either *replication-based* [3], [4], [6], [7] or *forwarding-based* [5], [9], [19]. Replication-based protocols trade huge overhead for performance, such as famous Epidemic [3] protocol, which can achieve an optimal performance at the expense of huge overhead. But huge overhead will decrease the throughput of network in constrained DTNs. Besides, most of these protocols are based on unlimited bandwidth, except for MaxProp [6] and RAPID [7]. Both MaxProp and RAPID employ packet scheduling mechanisms to schedule packets for separate contact opportunity. However, neither of them consider the scenario that multiple contact opportunities with the same source node may arise in the same time interval. Therefore, existing protocols cannot obtain an optimal metric under all scenarios of DTNs. Similar transmission scheduling problem has also been investigated in [20], which is a centralized scheduling algorithm for all packets in the network and only based on a determined model in which all contact events are pre-determined. Therefore, we need a more general scheduling algorithm.

III. PRELIMINARIES

A. Network Model

In this paper, We focus on the effectiveness of transmission scheduling in DTNs. Therefore, we make some simple assumptions which will be addressed as part of our future work. For simplicity, time is divided into small fixed time slot, at which a contact either emerges or not. The bandwidth is limited due to short duration of contact and low bandwidth of radio. Besides, the buffer size is also limited for each node. We assume that a node can discover all other nodes within its communication range at the beginning of each time slot and two consecutive forwardings (e.g., forward message x from node A to B and then to C) cannot happen in the same time slot [15]. Moreover, we assume that packets can be successfully transferred in one time slot, without regarding conflict and retransmission, etc.

Formally, we model a DTN as a widely adopted weighted network in which each pairwise nodes have different contact rates [19], referring to some real DTNs, students in NUS [11] and rollerbladers in Paris [12], for instance. The contact rate between node u and v at any time slot is denoted by $\lambda_{uv} \in [0, 1]$. The expected delay of direct delivery between node u and v is ED_{uv} . The probability of delivery which happens in this time slot is λ_{uv} and corresponding delay is 1 time slot. On the other hand, delivery happens in future with probability $1 - \lambda_{uv}$ and delay $1 + ED_{uv}$ time slots. So $ED_{uv} = \lambda_{uv} + (1 - \lambda_{uv}) \times (1 + ED_{uv})$, $ED_{uv} = 1/\lambda_{uv}$ [9], [19]. Further, expected minimum delay is the expected time an optimal opportunistic routing scheme takes from source u to destination v , denoted by $EMD(u, v)$. The calculation method of EMD is in [5], where a shortest path algorithm is adopted. The transmission bandwidth between node u and v is defined as the number of packets

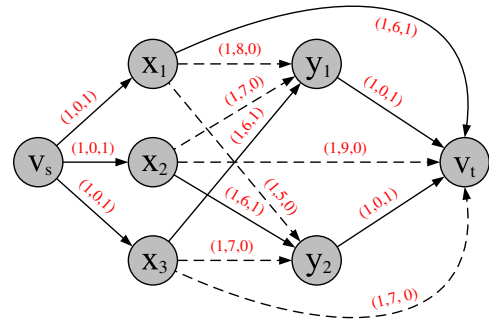


Figure 3. Cost Flow Network and corresponding Minimum Cost Maximum Flow of node A in Algorithm Class shown in Figure 1. Vertex x_i denotes the i^{th} packet in A and node y_j denotes the j^{th} neighbor of A . The red expression on each edge (c, w, f) denotes the capacity, cost, and flow of minimum cost maximum flow respectively.

that can be transferred via corresponding contact during one time slot, denoted by $B(u, v)$. Given a node u , we use $C(u)$ to denote its current available buffer size. For a given packet k , there are some definitions in Table II.

B. Problem Formalization

Based on previous works, we can identify several desirable design goals for a transmission scheduling scheme in constrained DTNs. Specifically, an efficient transmission scheduling algorithm proposed in this context should distribute packets to multiple related contact opportunities to minimize the average delivery delay. In this section we formalize the problem into a *linear programming problem*.

Before the formalization, we define some notations to describe the network parameters. Considering node u at current time slot, $P_u = \{p_u^i | 1 \leq i \leq n_u\}$ denotes the packets to forward. The nodes which are within the communication range of node u are called its neighbors, denoted by $C_u = \{c_u^j | 1 \leq j \leq m_u\}$. Because packets and neighbors are different at each time slot, we only define them at current time slot. In addition, we use $forward_{i,j} \in \{0, 1\}$ to indicate whether node u forwards packet p_u^i to node c_u^j in current time slot. And $forward_{i,j} = 1$ means forwarding, otherwise not. So

$$forward_{i,j} = 0 \text{ or } 1, \quad 1 \leq i \leq n_u, 1 \leq j \leq m_u \quad (1)$$

Each packet has only one copy in the network during the transmission from the source to the destination. That is

$$0 \leq \sum_{j=1}^{m_u} forward_{i,j} \leq 1, \quad 1 \leq i \leq n_u \quad (2)$$

Besides, due to the constraint of bandwidth, the number of packets that a neighbor might receive during current time slot is not more than the bandwidth between them. We have

$$0 \leq \sum_{i=1}^{n_u} forward_{i,j} \leq B(u, c_u^j), \quad 1 \leq j \leq m_u \quad (3)$$

The expected delay of a packet is composed of two parts: the time since creation and the expected remaining time. Then

$$D(p_u^i) = a(p_u^i) + r(p_u^i), \quad 1 \leq i \leq n_u \quad (4)$$

We use the expected minimum delay of a packet between the node which will have the copy at next time slot and its destination to estimate the expected remaining time of it. So

$$r(p_u^i) = EMD(x(p_u^i), d(p_u^i)), \quad 1 \leq i \leq n_u \quad (5)$$

The main purpose of our transmission scheduling is to minimize the average expected delay, which is equivalent to the minimum of total expected delay of all packets. That is

$$\min \sum_{i=1}^{n_u} D(p_u^i) \quad (6)$$

Up to now, we have formalized transmission scheduling problem in constrained DTNs into a linear programming problem. In next section, we will solve it further by transforming it to a minimum cost maximum flow problem.

IV. TRANSMISSION SCHEDULING ALGORITHM

To solve the linear programming problem above, we transform it to corresponding *minimum cost maximum flow* problem first. Then, we propose our transmission scheduling algorithm based on *mcmf*.

A. Cost Flow Network Construction

Before the construction, we present some definitions of cost flow network, denoted by $G = (V, E)$. Each edge (a, b) has capacity $c(a, b)$, flow $f(a, b)$ and cost $w(a, b)$. We use \mathcal{A} to denote the minimum cost maximum flow of G , where $f(\mathcal{A})$ and $w(\mathcal{A})$ are equal to the maximum flow and the minimum cost respectively.

Accordingly, every node in DTN has a corresponding cost flow network at each time slot, where flow is equivalent to the distribution strategy and cost is equivalent to corresponding expected delay. Considering node u at current time slot, the construction of $G = (V, E)$ is as followings:

\underline{V} is consisted of a source v_s , a sink v_t and two independent vertex sets $X = \{x_i | 1 \leq i \leq n_u\}$ and $Y = \{y_j | 1 \leq j \leq m_u\}$, where x_i denotes packet p_u^i and y_j denotes node c_u^j respectively.

\underline{E} is consisted of four categories of edges as follows:

- (v_s, x_i) , an edge from source v_s to each vertex $x_i \in X$, with one capacity and zero cost. So

$$c(v_s, x_i) = 1, \quad w(v_s, x_i) = 0 \quad (7)$$

- (x_i, v_t) , an edge from each vertex $x_i \in X$ to sink v_t with one capacity, whose cost equals to the expected remaining time of packet p_u^i in node u . So

$$c(x_i, v_t) = 1, \quad w(x_i, v_t) = EMD(u, d(p_u^i)) \quad (8)$$

- (x_i, y_j) , an edge from vertex x_i to vertex y_j with one capacity, whose cost equals to the expected remaining time of packet p_u^i in node c_u^j . So

$$c(x_i, y_j) = 1, \quad w(x_i, y_j) = EMD(c_u^j, d(p_u^i)) \quad (9)$$

- (y_j, v_t) , an edge from each vertex $y_j \in Y$ to sink v_t with zero cost, whose capacity is equal to the minimum value between transmission bandwidth from node u to node c_u^j and the available buffer size for node c_u^j . So

$$c(y_j, v_t) = \min\{B(u, c_u^j), C(c_u^j)\}, \quad w(y_j, v_t) = 0 \quad (10)$$

So far, a flow network $G = (V, E)$ for node u at current time slot has been constructed. Figure 3 shows the construction of node A introduced in Section 1.

B. Flow-Based Transmission Scheduling

Minimum cost maximum flow problem is a very classic problem. The surveys by Ahuja, Magnanti, and Orlin [1989,1991] and by Goldberg, Tardos, and Tarjan [1989] provide details concerning this field. Because some existing algorithms are based on nonnegative integer, we can transform the rational number of cost to nonnegative integer by adding a positive number first then multiplying by a suitably large number, which does not affect our results. The computational complexity of minimum cost maximum flow procedure is $O((n+m+2)^3)$, where n is the number of packets and m is the number of related contacts. The corresponding minimum cost maximum flow \mathcal{A} is shown in Figure 3 by solid lines.

After the flow procedure, we distribute packets according to the flow result \mathcal{A} . We will distribute packet p_u^i to node c_u^j if $f(x_i, y_j) = 1$. Packet p_u^i will not be transferred on condition that $f(x_i, v_t) = 1$. The integrated algorithm is formally described in Algorithm 1.

V. ANALYSIS AND IMPROVEMENT

To analyze the effectiveness of FBTS, we firstly prove the correctness of it. Then, we discuss the problem of starvation and congestion and provide the improvement on starvation prevention and congestion avoidance.

A. Correctness Proof

According to the relationship between linear programming and minimum cost maximum flow, the cost flow network we constructed satisfies the linear constraints in Section 3. Therefore, our work is to prove the optimality of FBTS based on corresponding minimum cost maximum flow.

Theorem 1: The average expected delay of FBTS is minimal according to corresponding *minimum cost maximum flow*.

Proof: considering node u at current time slot, obviously, the maximum flow of corresponding \mathcal{A} is $f(\mathcal{A}) = n_u$. Thus, there is one flow out of each vertex $x_i \in X$. $f(x_i, y_j) = 1$ means node u will forward packet

Algorithm 1 Flow-Based Transmission Scheduling Algorithm

Require: EMD, B, u, t .
Ensure: *forward*.

```

1: for  $i = 1$  to  $n_u$  do
2:    $c(v_s, x_i) = 1, w(v_s, x_i) = 0$ .
3:    $c(x_i, v_t) = 1, w(x_i, v_t) = EMD(u, d(p_u^i))$ .
4: end for
5: for  $j = 1$  to  $m_u$  do
6:    $c(y_j, v_t) = B(u, c_u^j), w(y_j, v_t) = 0$ .
7: end for
8: for  $i = 1$  to  $n_u$  do
9:   for  $j = 1$  to  $m_u$  do
10:     $c(x_i, y_j) = 1, w(x_i, y_j) = EMD(c_u^j, d(p_u^i))$ .
11:   end for
12: end for
13: Adopt minimum cost maximum flow algorithm
14: for  $i = 1$  to  $n_u$  do
15:   for  $j = 1$  to  $m_u$  do
16:     $forward_{i,j} = f(x_i, y_j)$ .
17:   end for
18: end for

```

p_u^i to node c_u^j . And $f(x_i, v_t) = 1$ means packet p_u^i will not be transferred in this time slot. Since the cost of \mathcal{A} is:

$$w(\mathcal{A}) = \sum_{f(x_i, y_j)=1} w(x_i, y_j) + \sum_{f(x_i, v_t)=1} w(x_i, v_t) \quad (11)$$

According to (8) and (9), then

$$w(\mathcal{A}) = \sum_{i=1}^{n_u} EMD(x(p_u^i), d(p_u^i)) \quad (12)$$

Because $w(\mathcal{A})$ is minimal and $\sum_i a(p_u^i)$ is only related to the given time slot, $\sum_i D(p_u^i) = \sum_i a(p_u^i) + w(\mathcal{A})$ is minimal. Therefore, we draw the conclusion that the average expected delay is minimal in FBTS according to corresponding minimum cost maximum flow. ■

B. Starvation Prevention

Due to the low priority, some packets might not be scheduled to relay nodes till expiring of TTL, on condition that packets with higher priority always exist in queue. Therefore we use a special utility to prevent starvation.

Definition 1: (Transfer Priority) Transfer priority is a per-packet priority utility derived for preventing starvation of packets, which increases as the queuing time increases with factor α , denoted by $TP(k)$ for packet k .

$$TP(k) = \alpha \cdot q(k), \quad \alpha \in (0, 1]$$

Considering node u , the method of preventing starvation is to modify the cost of edges out of vertex $x_i \in X$ by subtracting corresponding $TP(p_u^i)$. Because *Transfer Priority* is only related to the queuing time, we can easily draw that the modification does not affect the correctness of FBTS. So new cost of \mathcal{F} becomes:

$$w(\mathcal{F}) = \sum_{i=1}^{n_u} EMD(x(p_u^i), d(p_u^i)) - \alpha \times \sum q(p_i)$$

$\sum q(p_i)$ is certain at time t , so the modification does not affect the correctness of our scheme. Besides, we can get another two corollaries:

- The possibility of distributing a packet to relay nodes increases as its queuing time increases.
- The packets with same destination are transferred in queuing order.

As queuing time of packet p_u^i increases, corresponding $w(x_i, y_j)$ decreases. Accordingly, the possibility of augmenting a flow through edge (x_i, y_j) increases. The second corollary can be proved by the reverse proving. If two packets with the same destination are not transferred in queuing order, we could exchange their distribution strategies to obtain a better delivery delay. Through our simulation, we find that $\alpha = 0.1$ can achieve better results.

C. Congestion Avoidance

The queuing time of a packet will be long when many packets with the same destination are in front of it. Therefore, we introduce the improvement on avoiding congestion if many same destination packets are in queue.

Definition 2: (Expected Transfer Speed) For the packets with the same destination v in node u , Expected Transfer Speed is the expected number of them that can be transferred during one time slot, denoted by $ETS(u, v)$. Let $RS_{uv} = \{z | EMD(z, v) < EMD(u, v)\}$ denote the nodes whose expected delay is smaller than that in u , so

$$ETS(u, v) = \sum \lambda_{u,z} \cdot B(u, z)$$

Definition 3: (Expected Queueing Time) Expected Queueing Time is the expected time a packet will spend on queuing in current node, denoted by $EQT(k)$ for packet k . Let N_k denote the number of the same destination packets queuing in front of k , then

$$EQT(k) = N_k / ETS(h(k), d(k))$$

Considering node u , the method of avoiding congestion is to modify the cost of each edge (x_i, v_t) by adding corresponding $EQT(p_u^i)$. So the possibility of distributing the packets queuing in the back to other emerged nodes increases (if there is redundant bandwidth).

D. Local Estimation

In this sub-section, we investigate an estimation of EMD based on the partial information which can be obtained directly by a given node. We get help from "ego network", which is used as local data structure to maintain the partial information in social network analysis. Ego networks [21] consist of a focal node ("ego") and the nodes to whom ego is directly connected to (these are called "alters") plus the ties, if any, among the alters. Therefore, each node is responding for maintaining the average delay of inter-meeting for every neighbor (D_i for the i^{th} neighbor). Besides, we also define an associated value of delay for each node (DV), which is used to predicate the delay of future meeting. To track the delay of inter-meeting, a node also needs to maintain its latest

meeting time for each neighbor, denoted by T_i for the i^{th} neighbor. Hence, the value of average delay and meeting time are updated when the node encounters with a neighbor, and its corresponding ego network also needs to be updated. When two nodes encounter with each other, they firstly update their local data, and then exchange their latest local data, finally update their local data again.

Given node v , for each message $m \in P_v$ in the queue, if the destination of m is one neighbor of v , namely $d(m) \in S_v$, for every neighbor $u \in S_v$, we will use the delay of inter-meeting between u and the destination $d(m)$ to approximate the expected remaining delay of message m on node u . Otherwise, we assume each neighbor can deliver m to its destination with different delivery delay, which is related to respective delay value DV , with a factor D_{init} . Therefore, given a node v

$$E\tilde{M}D(u, d(m)) = \begin{cases} EMD(u, d(m)) & \text{if } d(m) \in N_v \\ DV_u \cdot D_{init} & \text{otherwise} \end{cases}$$

Besides, different D_{init} can lead to different results. According to our simulation on different D_{init} s based on *NUS Trace*, D_{init} equals to 20 can achieve a better delivery, seen figure 4, on condition that bandwidth equals to 40 pkts/slot and buffer size is infinite.

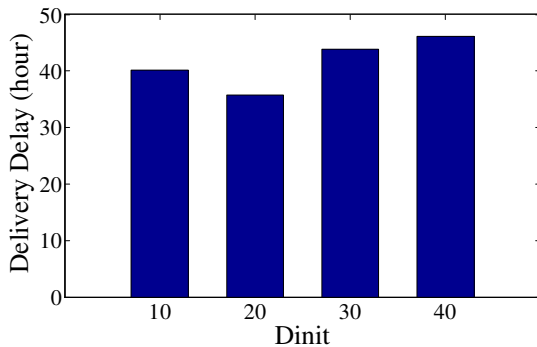


Figure 4. Delivery Delay versus D_{init} in NUS

VI. SIMULATION

In this section, we evaluate our FBTS with above improvement against three other protocols by using a wide variety traces: student contact patterns in NUS [11]. The simulator we experiment on is developed ourselves in JAVA on Eclipse, which is similar to a prevailing DTN simulator, ONE [22].

A. Protocols in comparison

The protocols in comparison are MED [5], MaxProp [6] and RAPID [7]. For fairness in the comparison, we use their enhanced versions which make use of the same level of prior knowledge of historical connectivity patterns as FBTS does. In our copy-controlled enhanced versions, all of them are based on *single-copy*. We use FIFO mechanism for MED. Similar experimental design can be seen in [14], [15].

TABLE III.
SETTINGS FOR NUS TRACE

parameter name	default or range
number of sessions	30
number of students	100
length of time slot	1 hour
time-to-live of packet	40 hours
simulation time	400 hours
packet production rate	5~40 packets per time-slot
transmission speed (bandwidth)	10~80 packets per time-slot

MED [5]. It is a source routing protocol based on the same level of prior knowledge as FBTS. The difference is MED fails to exploit superior edges which become available after the route has been computed.

MaxProp [6]. This protocol is based on limited bandwidth. MaxProp takes f_j^i to denote the probability of the next meeting node of node i is j . The delivery probability from a source to a destination is the total cost on their shortest path. In the enhanced version MaxProp*, we let $f_j^i/f_k^i = \lambda_{ij}/\lambda_{ik}$.

RAPID [7]. This protocol is also based on limited bandwidth. RAPID forwards packets with the highest value of δU_k , which denotes the increase in U_k by forwarding packet k . The corresponding utility U_k is the negative value of expected delay. In the enhanced version RAPID*, U_k equals the time since creation plus the remaining expected delay.

B. Simulation on NUS

Accurate information of human contact patterns is available in several scenarios such as university campuses. Student contact patterns in National University of Singapore (NUS) [11] were inferred from the information on class schedules and class rosters for the Sprint semester of 2006 in which there were 22341 students and 4885 class sessions included. If one knows class schedules and student enrollment for each class on a campus, accurate information about contact patterns between students over large time scales can be obtained without a long-term contact data collection. The advantage of the trace is that it exhibits a DTN based on a real social network and provides contact patterns of a large population over a long period. Many existing protocols [14], [15] are evaluated on this trace.

We select several class sessions M and a number of students N in each experiment. Contacts related to non-selected students or non-selected class sessions are ignored. Assume each student attends a class with a attendance probability 0.8 and sends packets to others randomly at each time slot. We define the number of packets generated by each student every time slot as *production rate*.

Our data processing includes the following steps: (1) The selection of M class sessions. If they are selected randomly, we cannot guarantee the connectivity of network. Thus we design a new selection method as follows. The first session is selected randomly. We select the k^{th} session c_k as the one with the highest score

$\sum comm(c_i, c_k)$ among the sessions that are not yet selected and not conflict with selected sessions in fields of time, where the function $comm$ is defined as the number of common students enrolled in both class sessions. (2) The selection of N students is in decrease order of the number of selected class sessions each student enrolled in. To guarantee the connectivity of network, the candidate must have some contacts with selected students (except the first student). (3) Calculate encounter probability of each pair of selected students at each class session.

C. Results of Simulation

In the simulation, we evaluate Delivery Ratio, Delay and Hop versus bandwidth (from 10 to 80 packets per time-slot), packet production rate (from 5 to 40 packets per hour) and buffer size (from 1000 to 8000 packets per node). The detailed simulation settings for NUS trace are shown as Table III. Figure 5 illustrates delivery ratio, delay and hop versus bandwidth, with default TTL (48 time-slots) and packet production rate (10 pkts/hour). The delivery ratio is approximately 24% greater than RAPID and 55% greater than MaxProp. For the delivery delay, the other three protocols are at least 56% longer than FBTS. The hop of FBTS is the smallest in four, whose mean value is 2.4. Figure 6 presents these three metrics versus packet production rate on condition that the bandwidth is 40 pkts/time-slot and the buffer size is infinite. Compared with three other protocols, FBTS shorten at least 30% on delivery delay. When the TTL equals to 48 hours, FBTS can achieve 70.7%, 57.5% and 31.2% more than MED, MaxProp and RAPID, respectively. The average hop of FBTS is 2.4, which is the smallest in all. These metrics versus buffer size are shown on Figure 7. The delivery delay of MED is 53.1% longer than FBTS on average, and MaxProp and RAPID obtain 65.0% and 52.0% longer, respectively. If TTL is 48 hours, the delivery ratio of FBTS is 39.8%, 47.7% and 21.2% greater than MED, MaxProp and RAPID, respectively. At the mean time, the smallest hop is also obtained by FBTS, whose value is 2.8 on average.

VII. CONCLUSION

In this paper, we first investigated a more general transmission scheduling problem, and then formalized the problem into a *linear programming* problem, which was transformed to a *minimum cost maximum flow* problem further. Finally, we proposed our *Flow-Based Transmission Scheduling* algorithm to minimize the average delivery delay, which can apply to more applications of DTNs, including the scenario that multiple related contact opportunities may arise in the same time interval. In addition, we proved the correctness of FBTS and provided the improvement on starvation prevention and congestion avoidance, as well as estimation on the expected delivery delay locally. Moreover, FBTS was evaluated on the real DTNs traces against MED, MaxProp and RAPID. And the simulation results verified the efficiency of FBTS in delivery ratio, delay and hop.

ACKNOWLEDGMENT

This work is supported by the National Grand Fundamental Research 973 Program of China (Grant No.2011CB302905), the National Science and Technology Major Project under Grant No. 2011ZX03005-002, the National Natural Science Foundation of China (Grant No. 60803009), and the Natural Science Foundation of Jiangsu Province in China (Grant No.BK2009150).

REFERENCES

- [1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM SIGCOMM*, 2003.
- [2] "Delay Tolerant Network Research Group," <http://www.dtnrg.org/>.
- [3] A. Vahdate and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke University, Tech. Rep. CS-2000-06, June 2000.
- [4] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings WDTN*, 2005.
- [5] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical routing in delay-tolerant networks," in *Proceedings WDTN*, 2005.
- [6] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *IEEE INFOCOM*, 2006.
- [7] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *ACM SIGCOMM*, 2007.
- [8] S. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *IEEE INFOCOM*, 2009.
- [9] M. Xiao, L. Huang, Q. Dong, A. Liu, and Z. Yang, "Optimal opportunistic routing in probabilistically contacted delay tolerant networks," *Journal of Computer Science and Technology*, vol. 24, no. 5, pp. 975–986, 2009.
- [10] 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 *ACM MobiCom*, 2007.
- [11] V. Srinivasan, M. Motani, and W. T. Ooi, "Analysis and Implications of Student Contact Patterns Derived from Campus Schedules," in *ACM MobiCom*, 2006.
- [12] P. U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. de Amorim, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on dtn routing," in *IEEE INFOCOM*, 2009.
- [13] "CRAWDAD data set," <http://crawdad.cs.dartmouth.edu/>.
- [14] C. Liu and J. Wu, "Routing in a cyclic mobispace," in *ACM MobiHoc*, 2007.
- [15] —, "An optimal probabilistic forwarding protocol in delay tolerant networks," in *ACM MobiHoc*, 2009.
- [16] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *ACM MobiHoc*, 2009.
- [17] I. C. Q. Yuan and J. Wu, "Predict and relay: an efficient routing in disruption-tolerant networks," in *ACM MobiHoc*, 2009.
- [18] S. Z. Q. Li and G. Cao, "Routing in socially selfish delay tolerant networks," in *IEEE INFOCOM*, 2010.
- [19] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 773–782, 2008.
- [20] D. Hay and P. Giaccone, "Optimal routing and scheduling for deterministic delay tolerant networks," in *Wireless On-Demand Network Systems and Services*, 2009.

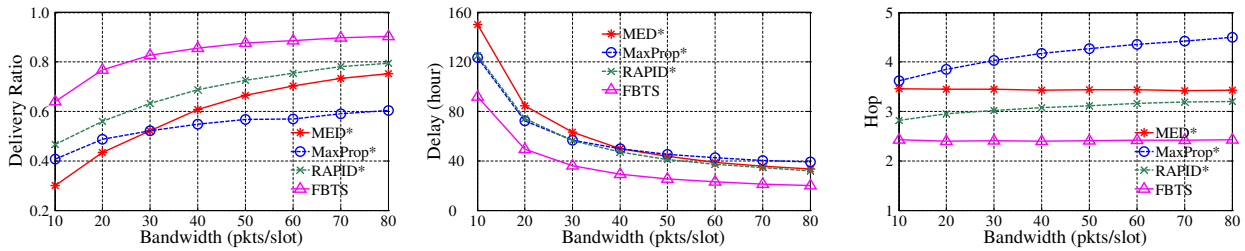


Figure 5. Delivery Ratio, Delay and Hop versus bandwidth (packets per time-slot) in NUS

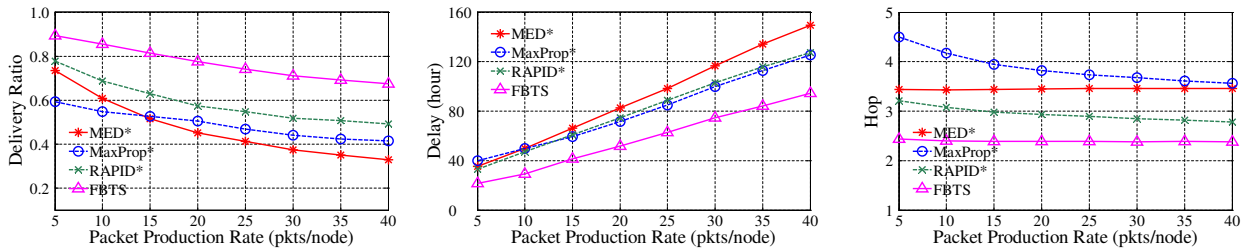


Figure 6. Delivery Ratio, Delay and Hop versus packet production rate (packets per time-slot) in NUS

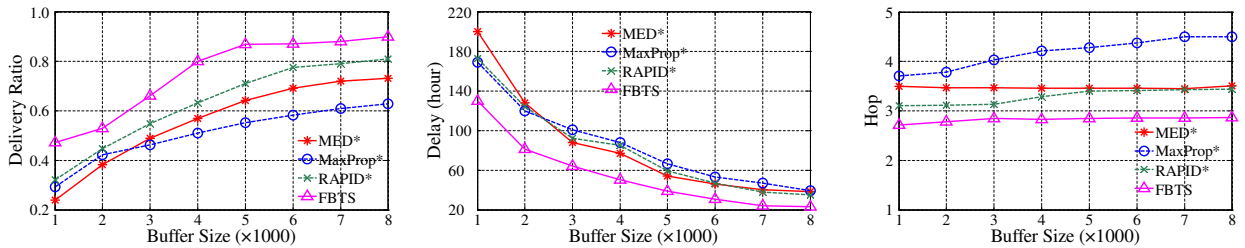


Figure 7. Delivery Ratio, Delay and Hop versus buffer size (packets per node) in NUS

[21] E. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant manets," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 5, pp. 606–621, may 2009.

[22] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. New York, NY, USA: ICST, 2009.

Zhenguo Yang, born in 1984, is currently a Ph.D. candidate in the Department of Computer Science and Technology, University of Science and Technology of China. His main research interests include wireless sensor networks and delay tolerant networks. E-mail: zgyang@mail.ustc.edu.cn.

Liusheng Huang was born in Anhui Province, China, in 1957. He received the M.S. degree in computer science from University of Science and Technology of China in 1988. He is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at the University of Science and Technology of China. He has published 6 books and more than 90 papers. His research interests are in the areas of wireless sensor networks, information security, distributed computing and high performance algorithms.

Mingjun Xiao, born in 1976. Received his Ph.D. degree from the Department of Computer Science and Technology, University of Science and Technology of China in 2004. Lecturer,

member of China Computer Federation. His major research interests include wireless sensor network and information security.

Wang Liu, born in 1985, is currently a Ph.D. candidate in the Department of Computer Science and Technology, University of Science and Technology of China. His main research interests include throughput capacity, mobility, delay within Wireless Mesh Network and Wireless Ad-Hoc Network.