

# Timely Data Delivery in a Realistic Bus Network

Utku Acer\*, Paolo Giaccone†, David Hay‡, Giovanni Neglia\*, and Saed Tarapiah†

\*INRIA, Sophia Antipolis, France

†Dipartimento di Elettronica, Politecnico di Torino, Turin, Italy

‡ Dept. of Electrical Engineering, Columbia University, NY, USA

**Abstract**—WiFi-enabled buses and stops may form the backbone of a metropolitan delay tolerant network, that exploits nearby communications, temporary storage at stops, and predictable bus mobility to deliver non-real time information.

This paper studies the problem of how to route data from its source to its destination in order to maximize the delivery probability by a given deadline. We assume to know the bus schedule, but we take into account that randomness, due to road traffic conditions or passengers boarding and alighting, affects bus mobility. In this sense, this paper is one of the first to tackle *quasi-deterministic* mobility scenarios.

We propose a simple stochastic model for bus arrivals at stops, supported by a study of real-life traces collected in a large urban network with 250 bus lines and about 7500 bus-stops. A succinct graph representation of this model allows us to devise an optimal (under our model) single-copy routing algorithm and then extend it to cases where several copies of the same data are permitted.

Through an extensive simulation study, we compare the optimal routing algorithm with three other approaches: minimizing the expected traversal time over our graph, maximizing the delivery probability over an infinite time-horizon, and a recently-proposed heuristic based on bus frequencies. We show that, in general, our optimal algorithm outperforms the three, but it essentially reduces to either minimizing the expected traversal time when transmissions are always successful, or maximizing the delivery probability over an infinite time-horizon when transmissions fail frequently. For reliable transmissions and “reasonable” values of deadlines, the multi-copy extension requires only 10 copies to reach almost the performance of costly flooding approaches.

## I. INTRODUCTION

We consider an opportunistic data network formed by (some) buses and bus stops equipped with wireless devices, e.g. based on WiFi technologies, like in DieselNet [9]. Most of the stops act as disconnected relay nodes (the throwboxes in [3]), and a few of them are also connected to the Internet. Data are delivered across town following the *store-carry-forward* network paradigm [29], based on multi-hop communication in which two nodes may exchange data messages whenever they are within transmission range of each other.

A bus-based network is a convenient solution as wireless backbone for delay tolerant applications in an urban scenario. In fact, a public transportation system provide access to a large set of users (e.g. the passengers themselves), and is already designed to guarantee a coverage of the urban area, taking into account human mobility patterns. Moreover, such a wireless backbone is not significantly constrained by power and/or memory limitations: a throwbox can be easily placed on a bus and connected to its power supply, or be put in an appropriate place in bus stops, which are usually already

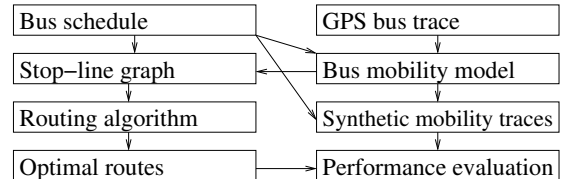


Fig. 1. High-level evaluation framework

connected to the power grid to provide lights and electronic displays, but also in any other places where power supply is available. Finally, travel times can be predicted from the transportation system time-table; even if the actual times are affected by varying road traffic conditions and passengers’ boarding and alighting times, such a backbone still provides strong probabilistic guarantees on data delivery time that are not common in opportunistic networks.

Indeed, this paper explores the basic question: “*how to route data over a bus-based network, from a given source to a given destination, so that the delivery probability by a given deadline is maximized?*”. We rely on the knowledge of bus schedule information and some stochastic characterization of bus mobility, supported by real data traces.

We consider two classes of routing schemes over such a network. The first class relies only on forwarding a *single copy* of the data is propagated along a *single path*. The second class takes advantage of *multiple copies* spread in the network to increase delivery probability and reduce delivery time, albeit with higher bandwidth usage.

Another architectural choice is between exploiting only bus-bus contacts, only bus-stop contacts, or both types of contacts. While the latter case should provide better performance, the two kinds of transmission opportunities have very different characteristics, making it hard to model both of them together in a common framework. For example, a potential contact between two buses traveling along orthogonal trajectories can be *completely avoided* if there is even a slight delay of one of them. On the other hand, in case of a bus-stop communication, the contact always happens eventually, but may be delayed. Most prior art (see Sec. II) considered only bus-bus communications. Our approach arises from considering rather bus-stop communications (nevertheless, we briefly discuss a bus-bus extension in Sec. VI).

Fig. 1 depicts the high-level framework used in the paper to study routing in the proposed network. Our starting point is a simple mobility model for buses (described in Sec. III-A), that

is supported by the statistical analysis of a set of real traces of the public transportation system of Turin in Italy, which serves an area of about 200 km<sup>2</sup> through about 7500 stops and 1500 vehicles distributed among 250 lines. These traces include the complete schedule for a working day and the corresponding GPS traces with the positions of all the vehicles during the morning rush hour period (6 AM–10 AM).

This mobility model allows us to represent the transportation system appropriately in terms of a graph with independent random weights, that we call the *stop-line graph* (Sec. IV). Under this representation, our original optimization problem to identify routes maximizing the delivery probability by a given deadline (or maximizing the on-time delivery probability) becomes equivalent to a specific stochastic shortest path problem on the stop-line graph. We are able to find an optimal algorithm, called ON-TIME, for the single-copy case (Sec. IV-A) and then to extend it for the multi-copy case through a greedy approach (Sec. IV-C). We compare the performance of these proposed algorithms with three other heuristics (Sec. IV-B) that also operate on the stop-line graph: an adaptation of the routing algorithm proposed in [28] for bus-bus communications (we refer to it as MIN-HEADWAY), and the two naïve algorithms, MIN-DELAY, that determines the path with the least expected weight, and MAX-PROB, that maximizes the delivery probability on an infinite time-horizon. Since the number of real-life traces we obtained is limited, the comparison (Sec. V) is based on simulations carried on a large set of synthetic traces generated on the basis of our bus mobility model and the schedule of Turin bus system.

The paper has the following main contributions and conclusions. (1) Formulation of the original routing problem as a specific stochastic shortest problem on a particular stochastic graph, that is justified by a statistical analysis of real transportation system traces. (2) Optimal (under our model) routing scheme for the single copy case. While this offline routing scheme has, in theory, an exponential worst-case time complexity, in practice it is able to find the optimal route in reasonable time, allowing each node to store an optimal pre-selected routing plan. (3) Extensions to multi-copy case, based on greedy approaches applied to the single-copy scheme. We prove a tight bound of  $1/k$  for the on-time delivery probability in comparison to an optimal (non-greedy)  $k$ -copy scheme. (4) Simulation analysis showing that the optimal algorithm outperforms the MIN-HEADWAY heuristic, but it performs as the MIN-DELAY algorithm when there is no packet loss, and as MAX-PROB when packet losses are significant across the network. We provide some explanation for these results. In this sense the conclusion is that naïve heuristics like MIN-DELAY or MAX-PROB algorithms may be very good heuristics for routing over realistic bus transportation networks. (5) Simulations showing that only 10 copies are needed for a multi-copy greedy approach to reach performance close to flooding routing policies; the latter requires at least two orders of magnitude more transmissions and copies for each single piece of data.

## II. RELATED WORK

Employing a bus network as a mobile backbone for dense vehicular networks was first proposed in [30], using standard routing protocols for mobile ad-hoc networks (e.g., DSR or AODV). More recently, buses employment in a disconnected scenario has been considered, e.g. in the seminal *DieselNet* project [9]. Since our paper addresses routing in such a network, in what follows we only mention work related to routing issues.

Most of the research has focused on bus-bus communications [2], [8], [13], [14], [28] with the following routing approach: Each vehicle learns at run time about its meeting process; then, the vehicles exchange their local view with other vehicles and use the information collected to decide how to route data. The goals of the proposed algorithms were either to reduce the expected delivery time or to maximize the delivery probability. Unlike these studies, we mainly focus on *bus to stop data transfers* and derive a single-copy routing algorithm to maximize the delivery probability by a given deadline. We then extend the algorithm to address settings where several copies of the same data are permitted. On the other hand, we do not consider buffer or bandwidth constraints, as they are not a major concern in our settings: When the mobile devices are buses (as opposed, for example, to cellular phones), it is reasonable to assume that there is sufficient storage available; in addition, since buses communicate with stops (as opposed to other moving buses), the amount of data transferable during a meeting is larger.

The use of fixed relay nodes was also considered in [3], [4]. In [4] an architecture is proposed where bus passengers may use the cellular network to require content that will be delivered to access points along the bus trajectory. This data can be replicated also on other buses, taking advantage of possible data transfers between vehicles. Their analysis considers only a simplistic single-street scenario and does not address routing issues. [3] reports that the performance of a vehicular network is improved by adding some infrastructure, like base stations connected to the Internet, a mesh wireless backbone, or fixed relays (which are similar to our stops). The most important results are (i) there are scenarios where a mesh or relay hybrid network is a better choice over a base station networks; (ii) deploying some infrastructure has a much more significant effect on delivery delay than increasing the number of mobile nodes. These findings, which were verified both analytically and by experiments on DieselNet testbed, support our proposed architecture that relies on an opportunistic connectivity between vehicle nodes and fixed relays.

In order to provide low cost Internet connectivity to fixed kiosks in rural areas of developing countries, KioskNet architecture has been proposed [15]. In this architecture, buses carry data between the kiosks and the gateways that are connected to the Internet. Routing of such data is achieved by simple flooding. On the other hand, gateways are delegated to a kiosk via a scheduling mechanism that considers the schedule of the

buses which serve the kiosk.

The routing algorithms proposed by [18]–[21] are intrinsically more suited for bus to bus data transfers. [19] and [21] propose algorithms that take advantage of cyclic mobility patterns, according to which nodes meet periodically, albeit with some probability. Even if a given bus may meet multiple times the same stop, this approach does not fit our scenario for three reasons. First, the bus-stop contact process is not necessarily periodic because vehicles may be assigned to different lines during one operation day. Second, even if a vehicle operates always on the same line, its frequency changes significantly along the day. Third and more importantly, even when a period may be defined, its time duration ranges from 30 minutes to 2 hours (depending mainly on the length of the bus trajectory and on inactivity times at terminus), and it is then comparable with the deadlines we are targeting, so that it is not possible to take advantage of such long term periodicity. Other forms of long-term regularities in the contact process of the different nodes [20] are too general for our settings, since we have significantly more information on the meetings that can be exploited to improve the performance. Finally, [18] proposes hierarchical routing for a deterministic network, whereas we consider non-deterministic mobility.

Almost all the papers above have considered only small bus networks (40 buses for DieselNet, 16 buses on a cyclic path for MobTorrent [4]). Only [13] considers an urban setting with a public transportation system comparable to ours (70 different bus lines), but, differently from us, they do not use any real mobility trace and simulate bus movement assuming that the bus speed is chosen uniformly at random from a given interval.

From the theoretical point of view, our optimization goal can be reformulated (under some assumptions) as a particular *stochastic shortest path problem* that deals with a graph  $G$  whose edge lengths (or equivalently, traversal times over the edges) are random variables. Several optimality criteria were considered in the past for routing in stochastic graphs. The most common one is the *least expected traversal time*, which can be generalized to any linear (or affine) utility function [27]. Other optimality criteria are deviance [5], monotonic quadratic utility functions [7] and prospect-theory-based functions [16]. Recent and comprehensive surveys of the different utility functions and corresponding solutions appear in [6], [26]. Our paper deals with the *reliability* of the chosen path, namely, finding a path which maximizes the probability of on-time arrival (given some deadline). This problem was first studied by Frank [12] and then was also investigated in [22]–[24] and more recently in [10], [11], [25], [26]. Current state-of-the-art algorithms still have exponential worst-case time complexity, based on enumerating over some set of candidate paths [26]. Yet, our problem differs from Frank’s problem essentially in three aspects. First, we have considered a real transportation system and therefore we are not interested in the worst-case complexity of some general graphs. Second, our transportation model has two kinds of entities: stations and buses; we need to take into account waiting time at the stops and not only buses travel times, as explained in details in Sec. IV. Third,

all the previous work considered a single-copy model, while our model deals also with multiple copies where the objective is that at least one of the copies arrives at the destination before the deadline. Finally, we observe that we use the bus network for data transfer as it is used for passenger transfer. Thus, one could expect that the same problem has already been addressed in the transportation literature (see [1] for more details). However, this is not the case: First, the possibility to exploit multi-copy is clearly absent in the transportation of people or merchandise. Second, the probability to miss a transfer opportunity is also not considered in transportation, while data transfer between two nodes may fail because of insufficient contact duration, channel noise or collisions. Third, even for single-copy routing, bus network passenger routes usually aim to *minimize the expected traversal time* (possibly limiting the maximum number of bus changes) and not to maximize the delivery probability by a given deadline, as we are doing. The fact that finally minimizing the expected traversal time may provide almost optimal performance in some scenarios is an a-priori unexpected finding of this research.

In conclusion, to the best of our knowledge, this is the first paper that proposes an optimal routing algorithm that takes advantage of bus schedule information as well as a stochastic characterization of bus mobility, supported by real data traces.

### III. MODEL DEFINITIONS AND ASSUMPTIONS

In this section, we formally define the terms and notations we use to describe a transportation system, following the terminology used in transportation literature.

A transportation system  $\mathcal{T}$  has a set of stops, denoted by  $\mathcal{S}$ , and a set of vehicles (buses), denoted by  $\mathcal{V}$ , which travel between the stops according to a predetermined path and a predetermined schedule. For each vehicle  $v \in \mathcal{V}$ , the schedule allows us to determine its *trajectory*, denoted  $\text{traj}(v)$ , which is the ordered sequence of stops the vehicle traverses:  $\text{traj}(v) = (s_0, s_1, \dots, s_n)$ . In addition, each vehicle  $v$  is associated with a *trip*, denoted  $\text{trip}(v)$ , which is a time-stamped trajectory:  $\text{trip}(v) = ((s_0, \tau_0), (s_1, \tau_1), \dots, (s_n, \tau_n))$ , such that a vehicle  $v$  should arrive at stop  $s_i$  along its trajectory at time<sup>1</sup>  $\tau_i = \tau(v, s_i)$ . We distinguish between the scheduled time  $\tau_i$  and the actual time  $t_i = t(v, s_i)$ , which is a random variable depending on road traffic fluctuations, passengers boarding and alighting, etc. The difference between the actual arrival time  $t(v, s_i)$  at a stop  $s_i$  and its corresponding scheduled arrival time  $\tau(v, s_i)$  is the *lateness*  $l(v, s_i)$  of the vehicle at stop  $s_i$ :  $l(v, s_i) = t(v, s_i) - \tau(v, s_i)$ . Note that the lateness is negative when the vehicle arrives earlier than its scheduled arrival. The *delay*  $d(v, s_i, s_j)$  between the stops  $s_i$  and  $s_j$  is the change in the lateness:  $d(v, s_i, s_j) = l(v, s_j) - l(v, s_i)$ . The time difference between the arrivals of a vehicle at two different stops  $s_i$  and  $s_j$ , is called the actual *travel time* between the two

<sup>1</sup>We do not introduce explicitly a departure time from the stop, because in our paper we do not take into account bandwidth constraints so that it is not important to specify the duration of the transmission opportunity between a bus and a stop. Moreover from our traces it is possible to determine the arrival time, but not the departure time.

stops,  $tt(v, s_i, s_j) = t(v, s_j) - t(v, s_i)$ . The scheduled travel time is simply the difference between the scheduled arrivals at the two stops.

A key concept in bus networks is the notion of *lines*, which are basically different vehicles with the same trajectory (at different times). Let  $\mathcal{L}$  denotes the set of lines. For each vehicle  $v \in \mathcal{V}$  we denote its corresponding line by  $line(v) = \{v' \in \mathcal{V} | \text{traj}(v) = \text{traj}(v')\}$ . Note that lines introduce an important characteristic of a bus transportation system: if a passenger misses a specific vehicle  $v$ , she can still catch another vehicle  $v'$  in  $line(v)$  and reach the same set of stops. The time between two consecutive arrivals of vehicles belonging to the same line at the same stop is called *headway*.

In the sequel, we will refer to the transportation system  $\mathcal{T}$  as the quintuple  $\langle \mathcal{S}, \mathcal{V}, \mathcal{L}, \tau(), t() \rangle$ , where the function  $\tau()$  is a way to represent the schedule and  $t()$  denotes a characterization of the stochastic process of vehicle arrivals at the stops. In the next section, we are going to start characterizing this stochastic process.

#### A. Bus Mobility and Communication Models

The problem of maximizing the delivery probability by a given deadline requires a realistic statistical characterization of bus mobility patterns, which is also useful to generate a large set of synthetic traces and evaluate the performance of our routing algorithms.

Transportation literature does not provide a universally valid model for bus movements in an urban environment, since they are strongly affected by vehicular and passenger traffic conditions, road organization (availability of separate lanes for buses), traffic signal control management (priority may be given to the approaching buses over the other traffic), company policies (penalties to the bus drivers for delays), and so on; details of our transportation literature survey are in [1]. Two extreme cases can be considered: 1) buses that are late at one stop can always recover their delay at the following stop (speeding up and reducing their travel times), 2) buses move almost in the same way, and they do not try to recover their delay. The first case better describes lines with high headway, while the second is probably more adapt for lines with short headways, where buses try to respect a given frequency, rather than an exact schedule<sup>2</sup>. In terms of the quantities we have defined above, in the first case, latenesses at consecutive stops are almost independent, while in the second case they are highly correlated.

We have performed a statistical analysis of a one day trace with actual bus arrivals at their stops provided to us by Turin's public transportation company. Due to lack of space, the full details of this analysis appears only in the accompanied technical report [1].

Here, we only present the following two consequences of this analysis, and refer to them as Assumptions 1 and 2. These hypotheses are going to be kept for granted in the rest of

the paper and will be fundamental to develop our routing algorithm.

*Assumption 1:* Bus travel times at consecutive stops are independent (but not necessarily identically distributed; in particular, their distribution will depend on the corresponding scheduled value).

*Assumption 2:* The distribution of the waiting time at a stop only depends on the stop and the characteristic of the departing bus line, not on the line of the arriving bus.

We note that Assumption 2, which plays an important role in enabling a graph representation with additive edge weights, is partially a consequence of Assumption 1: Consider buses moving according to the schedule, and transferring from line  $\ell_1$  to line  $\ell_2$  at stop  $s$ . It is clear that the waiting time at the stop can be evaluated a-priori on the basis of the scheduled arrival time of the  $\ell_1$  vehicle and the departure time of the following  $\ell_2$  vehicle. But under Assumption 1, arrival times of  $\ell_1$  buses at stop  $s$  are random variables and so are the corresponding waiting times. Intuitively, if the variability of  $\ell_1$  arrival times is large in comparison to the headway<sup>3</sup> of line  $\ell_2$ , the waiting time will have almost the same distribution of the waiting time seen by a Poisson observer, thus it is independent of  $\ell_1$ 's schedule. This is also supported by our trace analysis [1].

Finally, in our scenario we assume that data transfer during a transmission opportunity can fail. This can be due to different causes: channel noise and collisions, but also nodes failing to discover the communication opportunity, or contact duration being insufficient to transfer the data. Our main assumption is the following:

*Assumption 3:* The success probabilities for the message transmissions are independent.

#### IV. ROUTING ALGORITHMS IN A BUS NETWORK

As mentioned before, our routing algorithms aim to define an off-line routing for the transportation system that maximizes data delivery probability by a given deadline:

*Definition 1:* Given a transportation system  $\mathcal{T} = \langle \mathcal{S}, \mathcal{V}, \mathcal{L}, \tau(), t() \rangle$ , a source stop  $s_s$ , a destination stop  $s_d$ , a start time  $t_{start}$ , and a deadline  $t_{stop}$ , the *on-time delivery problem* is to find a route between  $s_s$  and  $s_d$  that starts after time  $t_{start}$  and maximizes the on-time delivery probability, i.e.  $\Pr\{\text{data is delivered before time } t_{stop}\}$ .

We first discuss how we represent the transportation system as a graph, considering the natural operation of a bus system with transfers from buses to stops and then to buses (i.e., involving only bus-stop communications). The following four issues lead to our final representation: computational complexity, intrinsic properties of the bus transportation system (namely, the existence of lines), characteristic of the stochastic process  $t()$  (namely, waiting times in the stops depends on the departing line), and an advantage coming from working with additive edge weights.

Specifically, in our representation, which we call *stop-line graph*  $G_{sl} = \langle V_{sl}, E_{sl} \rangle$ , the nodes are  $(s, \ell)$  pairs where  $s$  is

<sup>2</sup>This distinction is expressly advertised by Turin public transportation system, that label lines as frequency-based and schedule-based.

<sup>3</sup>According to our model the variance of the lateness increases along the trajectory.

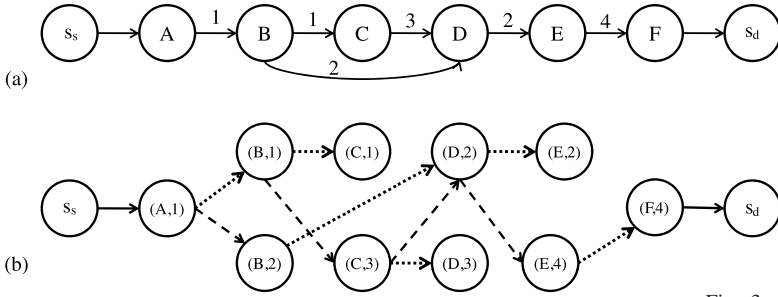


Fig. 2. (a) Example of bus network with  $S = \{A, B, C, D, E, F\}$  and  $\mathcal{L} = \{1, 2, 3, 4\}$ : the node corresponds to a stop and the label on the edge represents the line connecting the two stops. (b) The corresponding line-stop graph  $G_{sl}$ . Dotted edges are travel edges, while dashed edges are travel-switch edges.

a stop and  $\ell$  is a line;  $(s, \ell) \in V_{sl}$  if and only if line  $\ell \in \mathcal{L}$  arrives (or depart) at stop  $s \in S$ . In addition, we add two nodes  $s_s$  and  $s_d$  which are connected to all nodes that correspond to the source and destination stops. The edges of  $G_{sl}$  are defined as follows: An edge between  $(s, \ell)$  and  $(s', \ell')$  corresponds to routes between stops  $s$  and  $s'$  with line  $\ell$  that continue from stop  $s'$  on line  $\ell'$ . If  $\ell = \ell'$  we call the edge a *travel edge*, while if  $\ell \neq \ell'$  we call it a *travel-switch edge*. An example of  $G_{sl}$  appears in Fig. 2.

We now define the random variables associated to the edges in  $E_{sl}$ . The random variable of a travel edge describes the corresponding travel time between two stops: formally, a travel edge  $e = ((s, \ell), (s', \ell))$  is associated with the random variable  $w_e = tt(\ell, s, s')$  describing the travel time of a line  $\ell$  bus from stop  $s$  to stop  $s'$ . The random variable of a travel-switch edge includes the travel time between the corresponding stops and the waiting time for the next line, taking into account possible transmission failures. Formally, a travel-switch edge  $e = ((s, \ell), (s', \ell'))$  is associated with the following random variable  $w_e$ .

$$w_e = \begin{cases} +\infty & \text{with prob. } p_f \\ tt(\ell, s, s') + wt(\ell', s', k) & \text{with prob. } (1 - p_f)^2 p_f^{k-1} \end{cases}$$

for any  $k \geq 1$ ; here,  $p_f$  is the transmission failure probability and  $wt(\ell', s', k)$  is the waiting time at stop  $s'$  before the arrival of the next  $k$ th bus of line  $\ell'$ . Note that, to be able to switch the data successfully from one bus to another, two transmissions must succeed: the one from a bus of  $\ell$  to  $s'$  and the one from  $s'$  to a bus of  $\ell'$ . We assume that all the random variables defining  $w_e$  are known (they will be characterized in Sec. IV-A); moreover, by Assumptions 1, 2 and 3, they are all independent.

It is important to notice that the stop-line pair representation provides a unified approach to deal with waiting times at the stops, thus solving shortcoming in previous approaches (e.g., temporal network [17], or graphs with stops as nodes and lines as edges); further, although out of the scope of this paper,  $G_{sl}$  is also usable in settings where Assumption 2 does not hold. Our model allows us to define simply the overall traversal time of the data along a weighted path  $\mathcal{P}$

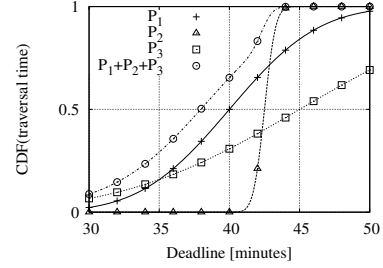


Fig. 3. Delivery probability CDFs of three disjoint paths  $P_1$ ,  $P_2$  and  $P_3$ , connecting a source and a destination with different traversal times and without transmission failures ( $p_f = 0$ ). Path  $P_1$  has the lowest expected traversal time; the variance of  $P_2$  is the smallest, while  $P_3$ 's variance is the largest.  $P_1$ ,  $P_2$  and  $P_3$  are respectively the optimal paths computed by ON-TIME for deadlines between 34 and 43 minutes, larger than 43 minutes, and shorter than 34 minutes. The curve labeled  $P_1 + P_2 + P_3$  corresponds to the success probability obtained by a multi-copy approach exploiting all the three paths.

as:  $tr(\mathcal{P}) = \sum_{e \in \mathcal{P}} w_e$ . Now, given the graph  $G_{sl}$ , the on-time delivery problem corresponds to finding a path  $\mathcal{P}$  such that  $\Pr\{tr(\mathcal{P}) \leq t_{stop} - t_{start}\}$  is maximized. Note that, under this construction, our problem is similar to the problem defined by Frank [12], with the differences highlighted at the end of Sec. II.

#### A. Single-Copy Routing Algorithm and Implementation

We now turn to define our routing algorithm, called ON-TIME, which aims at solving the on-time delivery problem. ON-TIME finds, in general, different paths for different values of (relative) deadline  $t_{stop} - t_{start}$ . For example, Fig. 3 compares the Cumulative Distribution Functions (CDF) for the delivery times of 3 different paths, for a given source-destination pair and no transmission failures ( $p_f = 0$ ). In this case, ON-TIME chooses one of the three paths depending on the given deadline. Nevertheless, the larger the deadline, the larger the resulting on-time delivery probability is.

ON-TIME works by first determining a potentially good path between the source to the destination (for example, that with the minimum expected traversal time), and evaluating its on-time delivery probability. This can be done by performing a (numerical) *convolution* of the different random variables distributions along the path, yielding the end-to-end traversal time distribution. By this distribution, it is then easy to calculate (using the corresponding CDF) the delivery probability by the deadline.

Then, the algorithm proceeds by exploring the graph through a breadth-first search, looking for paths with a higher on-time delivery probability. A *pruning* mechanism avoids the need to determine and evaluate all the paths. By the associativity of the convolution operator and the fact that our random variables are all non-negatives, for any path  $\mathcal{P}$  and any prefix  $\mathcal{P}'$  of  $\mathcal{P}$ ,  $\Pr\{tr(\mathcal{P}) \leq t\} \leq \Pr\{tr(\mathcal{P}') \leq t\}$ . Thus, we can perform hop-by-hop convolution and compute, for each resulting distribution, the probability that the weight (that is, traversal time) of this path's prefix is less than  $t_{stop} - t_{start}$ ; if the probability is smaller than that of the current best path, there is no need to consider the rest of the path. From a

practical point of view, working with a real transportation network, this simple pruning mechanism significantly reduces the number of paths to be considered, even if theoretically we may have a factorial number of paths to explore.

In our implementation, we have introduced some other simplifications, which reduce the computation time, but, at the same time, may lead to suboptimal paths. First, we have introduced a limit  $h$  of the exploration depth during the search. Given  $h$  as a constant, the algorithm is then guaranteed to run in polynomial time. We observe that upon termination, we may be able to say if the algorithm has selected the optimal path or there may be a better one. In fact, when we stop, if there is still some path prefix of length not larger than  $h$  such that the pruning mechanism cannot discard it, then there could be a longer path with higher on-time delivery probability. But if this is not the case, then the current best candidate is actually the optimal path. In our experiments on Turin transportation network,  $h = 8$  was enough to find all the best paths. Although this value may change for other networks, we expect that it will remain a relatively small constant. Note that a suitable  $h$  for each network can be found by conducting experiments similar to ours.

A second simplification is that we restrict the set of eligible paths such that each line can be used only in consecutive edges. This prevents the algorithm to explore paths using line  $\ell_1$  then line  $\ell_2$ , and then again line  $\ell_1$ . We expect that these paths have normally worse performance than those where data message just remains on line  $\ell_1$ .

Finally, we have avoided the computation burden of performing numerical convolution by assuming that the end-to-end traversal time, which is a sum of independent random variables, can be approximated by a normal distribution. In this case, it is sufficient to take into account the mean and the variance of each edge weight, conditioned on the fact that it is finite (respectively,  $\mu_e = E[w_e | w_e < \infty]$  and  $\sigma_e^2 = Var[w_e | w_e < \infty]$ ), and the probability that the edge weight is finite (denoted by  $p_e$ ). Then, the CDF of the traversal time of path  $\mathcal{P}$  is equal to the CDF of a normal distribution with mean  $\sum_{e \in \mathcal{P}} \mu_e$  and variance  $\sum_{e \in \mathcal{P}} \sigma_e^2$ , multiplied by a scaling factor  $\prod_{e \in \mathcal{P}} p_e$ . In the case of travel edges, average and variance of  $tt(\ell, s, s')$  can be measured directly on the traces. In the case of travel-switch edges, we have to also to evaluate the average and variance of  $wt(\ell, s, k)$  using the first three moments of the interarrival times of the line  $\ell$  buses to stop  $s$  (which can be also measured on the traces) and some basic Palm calculus. For example, assuming perfect periodic bus arrivals with period  $\delta$  and failure probability  $p_f$ ,  $E[wt(\ell, s, k)] = \delta(1/2 + p_f/(1 - p_f))$  and  $E[wt(\ell, s, k)^2] = \delta^2(1/3 + 2p_f/(1 - p_f)^2)$ . Note that these values can be computed for the specific arrival process observed in bus traces.

In what follows, we evaluate the performance of ON-TIME for different source-destination pairs under similar kind of deadlines. If we had fixed a given deadline for all the pairs, then this deadline could be unfeasible for some of them (in the sense that there is no way to deliver the message

by this deadline, e.g. if the deadline is smaller than the time a vehicle would take to move from the source to the destination), and trivially satisfiable for other pairs (many different paths would deliver with probability almost one). For this reason, given a source  $s_s$ , a destination  $s_d$  and a real value  $x \in [0, 100]$ , let  $\phi(x, s_s, s_d)$  be the deadline  $t_{stop}$  for which the on-time delivery probability of the path from  $s_s$  to  $s_d$  with minimum expected traversal time is  $x\%$  (assuming  $p_f = 0$ ). We denote by ON-TIME( $x$ ) the on-time routing algorithm where the deadline is set equal to  $\phi(x, s_s, s_d)$  for every source-destination pair  $(s_s, s_d)$ . Intuitively, the smaller  $x$  is, the “shorter” the deadlines are considered, where “short” is in relation to the expected traversal time from  $s_s$  to  $s_d$  and not in an absolute sense.

### B. Other Routing Approaches

Although the algorithm we described is optimal under our model assumptions, we also consider sub-optimal but simpler heuristics.

The most intuitive approach (denoted as MIN-DELAY) is to route in  $G_{sl}$  along the path whose expected traversal time is minimal. Note that MIN-DELAY is equivalent to ON-TIME(50) under the Gaussian assumption on the distribution of the traversal time. Fig. 3 shows that path  $P_1$ , found by MIN-DELAY, does not always correspond to the highest on-time delivery probability. On the other hand, MIN-DELAY is computationally attractive, because the path with the least expected traversal time can be easily computed with Dijkstra’s algorithm (by linearity of expectation). In Sec. V, we compare our optimal algorithm to this sub-optimal heuristic and show that it often suffices to use this simple approach.

A second algorithm, MAX-PROB, selects the path that maximizes the delivery probability on an infinite time-horizon. Also this path can be determined running Dijkstra’s algorithm on the line-stop graph with edge weights equal to  $-\log(p_e)$ . MAX-PROB and ON-TIME tend to select the same path, for low transmission success probabilities, as shown at the end of Sec. V.

Another approach, denoted MIN-HEADWAY, tries to minimize the sum of all lines headways along a path [28], thus preferring frequent lines over infrequent ones; it was proposed originally for bus-to-bus communications. In Sec. V, we show that it has the worse performance in our settings among all the different algorithms.

### C. Extension to Multi-Copy Routing

As shown in the toy-case of Fig. 3, using a multi-copy scheme (the curve labeled “ $P_1 + P_2 + P_3$ ”) to exploit several paths simultaneously increases the on-time delivery probability to deliver the data within the deadline. In this specific example, path  $P_2$  becomes “useful” only for large deadlines, whereas  $P_3$  is “useful” for any deadline.

For multi-copy scheme, we consider only non-flooding algorithms, such that at most  $k$  copies of the packets are made throughout the execution (otherwise, an optimal flooding scheme can copy the data whenever there is a contact, namely

in an *epidemic manner*, thus achieving the best possible delivery probability).

We propose a greedy Multi-Copy algorithm for on-time routing, denoted simply as MC-ONTIME. It computes the on-time delivery probability of all paths in isolation and choose the  $k$  best paths (without considering the interaction between them). This can be easily implemented by saving the best  $k$  paths while enumerating all possible paths as in ON-TIME. Moreover, our pruning mechanism is changed accordingly to consider the  $k$ -th best value discovered so far (rather the maximum value as in the single-copy settings)<sup>4</sup>.

However, since our algorithm works in a greedy manner, it does not consider the interaction between the paths, and more specifically the gain in probability over previously-selected paths (which can be very small in case the paths overlaps). This leads to a theoretical performance degradation with respect to an optimal, infeasible algorithm that considers the joint-probability over all sets of paths. The following theorem, whose proof is in [1], provides tight bounds on this performance degradation:

*Theorem 1:* The MC-ONTIME algorithm always achieves at least  $1/k$  of the on-time delivery probability of an optimal  $k$  multi-copy algorithm. In addition, there is a valid transportation graph for which MC-ONTIME achieves at most  $\frac{1}{(1-\varepsilon)^k}$  of the on-time delivery probability of an optimal  $k$  multi-copy algorithm, for arbitrarily small  $\varepsilon > 0$ .

The performance degradation is mainly due to path overlapping; consider two paths with high success probability that differ only in one edge: MC-ONTIME will choose both paths, while, in fact, the marginal gain in choosing the second path is small. Thus, we consider also an algorithm that ensures that the paths are disjoint. Namely, the MC-ONTIME-DISJOINT algorithm iteratively chooses the path with the highest on-time delivery probability, among all paths from source to destination whose corresponding lines are not used by any previously-selected path. However, we show that the worst-case performance of MC-ONTIME-DISJOINT is the same as MC-ONTIME. Our simulations clearly show that the MC-ONTIME is superior in practice, and therefore this is the multi-copy routing algorithm we consider in the sequel.<sup>5</sup>

## V. PERFORMANCE EVALUATION

We consider a set of 180 source-destination ( $s_s-s_d$ ) stop pairs; in the first 90 pairs both the source and the destination have been chosen uniformly at random in the entire metropolitan area; in the second 90 pairs, the source  $s_s$  is located in a main transportation hub within the city center (close to the main train station), and all the destinations  $s_d$  are chosen uniformly at random. We generate a set of 100 traces with

<sup>4</sup>When comparing to the heuristics of Sec. IV-B, we can similarly get the  $k$  paths with minimal expected traversal time, total headway or maximal success probability.

<sup>5</sup>MC-ONTIME-DISJOINT and MC-ONTIME are two extremes as for the amount of overlapping between the paths. In our future research, we plan to look also on hybrid heuristics with strict bounds on the number of overlapping edges. While these variants yield the same  $\frac{1}{k}$  worst-case approximation, they might be proved superior in real-life traces.

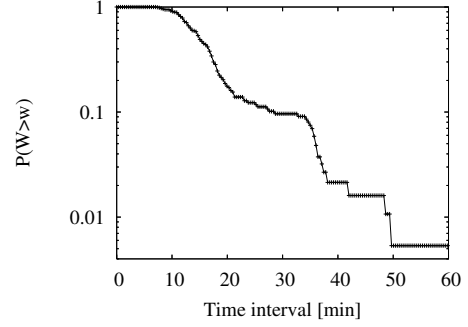


Fig. 4. Complementary CDF of the critical time window  $W$  guaranteeing on-time delivery probability  $\in [0.1, 0.9]$  for the minimum expected traversal-time path.

the parameters obtained by the statistical analysis, covering all 250 lines for the four hours available from the schedule. In addition, we have developed a simulator that computes the delivery probability of each path by averaging across these 100 traces; note that the one day real-life trace alone would not be enough to compute this probability with any accuracy. Data is assumed to be available at the source stop at 7 AM.

For these 180  $s_s-s_d$  pairs, we start to evaluate the size of the “critical” time window defined as  $W = \phi(90) - \phi(10)$ : this is the amplitude of the interval of “reasonable” deadlines for which MIN-DELAY and ON-TIME(50) achieve delivery probability in  $[0.1, 0.9]$ ; intuitively, when considering any deadline outside this critical time window, one is either likely to fail or to succeed, and the randomness in the transportation system does not play a major factor. Fig. 4 shows the inverse CDF of  $W$ , considering the whole set of 180 pairs. For more than 90% of  $s_s-s_d$  pairs, the windows is larger than ten minutes and for more than 17% of them, it is even larger than 20 minutes. The maximum critical window size we observed is 67 minutes. As a consequence, the time window for which the deadline plays an important role on the delivery probability cannot be neglected for most of these 180  $s_s-s_d$  pairs.

Then, for all 180 pairs and for all 100 traces, we evaluate the optimal paths found by the ON-TIME algorithm and compare their theoretical on-time delivery probability with the empirical one determined by simulations. We found a reasonable agreement, even if not perfect in absolute values since in our model we assumed that line frequency and headway distribution do not change over time or between station along the same line; in real-life, there are small fluctuations in these values. In addition, while generating the synthetic traces, we introduce some inhomogeneity in the travel time distribution to ensure that buses maintain their order; our model, on the other hand, considers homogeneous travel time distribution that depends only on the scheduled travel time.

We start to compare the performance of the algorithms defined in Sec. IV—namely, MIN-DELAY, ON-TIME, MAX-PROB and MIN-HEADWAY—with the EPIDEMIC algorithm that floods the network by taking advantage of all the possible contacts (and therefore making very large number of copies).

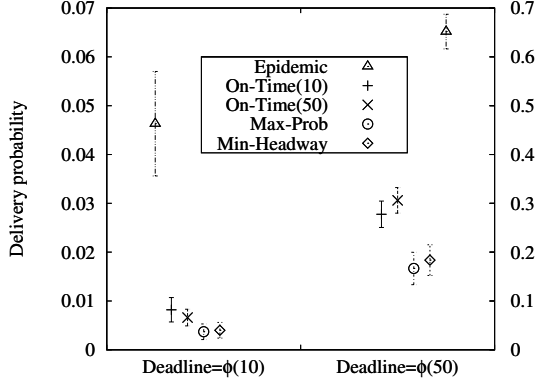


Fig. 5. Delivery probability (average and 90% confidence interval) for two deadlines and different routing algorithms, for reliable transmission ( $p_f = 0$ ). MIN-DELAY is the same as ON-TIME(50).

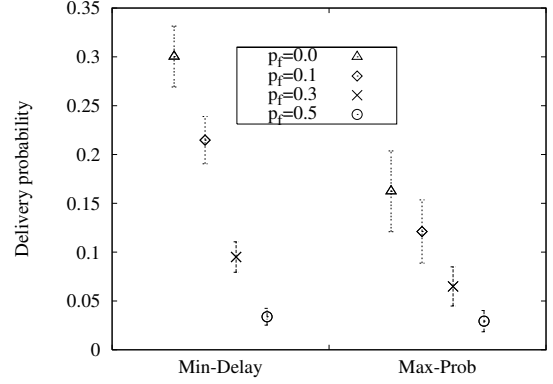


Fig. 6. Delivery probability (average and 90% confidence interval) for MIN-DELAY (i.e., ON-TIME(50)) and MAX-PROB for deadline  $\phi(50)$  and for different values of transmission failure probability  $p_f$ .

We first assume that transmissions are reliable, i.e.  $p_f = 0$ . We evaluate the actual on-time delivery probability of the best path obtained by each algorithm; for each pair  $s_s - s_d$ , we set the deadline to  $\phi(x)$  for different values of  $x$ , and we compute the 90% confidence interval of the delivery probability considering all the possible 180 pairs. Due to the lack of space, we will report the results only for  $x = 10$  (“short deadline”) and  $x = 50$  (“average deadline”), since these cases are representative.

Fig. 5 compares the delivery probability of the different algorithms for the two deadlines. The gain on the delivery probability of EPIDEMIC with respect to all the other single-copy algorithms decreases as the deadline increases: the factor of gain is more than 5 for deadline  $\phi(10)$  and around 2-3 for deadline  $\phi(50)$ . Indeed, when the deadline is large enough, outside the critical time window, just one copy of the data is enough, independently from the actual path found by the specific routing algorithm; in such a case, EPIDEMIC does not introduce any gain in terms of performance, and the cost in terms of copies and transmissions is prohibitive (we observed on average more than 600 copies for  $\phi(10)$  and more than 900 copies for  $\phi(50)$ ) than the single-copy algorithms, for which the number of transmissions for each data is on average 5.5, and always less than 12.

ON-TIME(10) and ON-TIME(50) obtain the maximum delivery probability respectively, for deadline  $\phi(10)$  and  $\phi(50)$ , as expected. But comparing the corresponding confidence intervals, they behave almost the same. A somewhat surprising result is that in many cases (121 out of 180) ON-TIME(10) performs *exactly* as ON-TIME(50) (or, equivalently, as MIN-DELAY). In such cases, we verified by direct inspection that ON-TIME(10) and ON-TIME(50) select exactly the same optimal path.

These results have been confirmed also for other deadlines: ON-TIME(50) usually selects the best path computed by ON-TIME( $x$ ) for the deadline  $\phi(x)$ . Recall the example in Fig. 3, showing that the best path may depend on the deadline. While it is possible in a general setting, our experiments lead us to

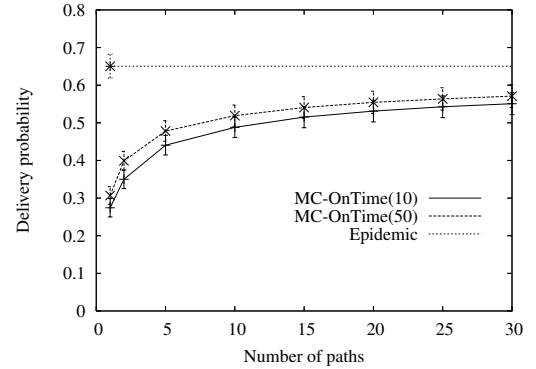


Fig. 7. Delivery probability (average and 90% confidence interval) vs. number of paths for deadline  $\phi(50)$  and for multi-copy routing and no transmission failures ( $p_f = 0$ ).

conclude that these cases are very rare in a real transportation system. Thus, one can choose the path solely on the basis of the minimum expected travel time (that is, the simple MIN-DELAY algorithm), making it redundant to run the complex optimal algorithm ON-TIME.

We now investigate the effect of transmission failures. Fig. 6 shows the delivery probability for different values of transmission failure probability  $p_f$ . When  $p_f$  increases, MIN-DELAY behaves very similarly to MAX-PROB; we expect that MAX-PROB becomes very efficient when the transmission failures are high, since the best policy must minimize the number of transmissions. Hence, both MIN-DELAY and MAX-PROB appear to behave very efficiently for large  $p_f$ .

We turn now to deal with multi-copy settings. Fig. 7 shows the performance of the MC-ONTIME policy, applied to the first best pre-computed paths found by each routing algorithm in all the considered 180 source-destination pairs, assuming reliable transmissions ( $p_f=0$ ). For deadline  $\phi(50)$ , ON-TIME with one copy reaches a delivery probability which is slightly less than half (more precisely, 42% and 47%) than the epidemic case, and few copies of MC-ONTIME improve the performance significantly by a factor 1.7-1.9. Yet,



after 10 copies we observe only a negligible improvement. This is partially due to the fact that MC-ONTIME exploits a given sequence of paths provided by the algorithms, whose internal “diversity” among the paths is limited. Furthermore, EPIDEMIC exploits low-probability paths that are efficient just for the specific trace instance considered in each simulation run; since the number of these low-probability paths can be very large, due to the redundant connectivity of the bus transportation system metropolitan area, there might be a high probability that at least one of them will be used to deliver to the destination. Note that the cost in terms of transmissions and copies for EPIDEMIC (on average, more than 900) is two order of magnitude larger than the multicopy approach using a pre-selected subset of 10 paths.

## VI. CONCLUSIONS

This paper lays the foundations for a framework to analyze bus-based networks, where communication is between the mobile buses and the stops along their trajectories. Through a statistical analysis of traces, taken from a real transportation system of a large urban area, we were able to obtain a succinct stochastic graph representation of the system, and to devise routing algorithms on this graph. In addition, we were able to develop a synthetic trace generator, which in turn allowed us to perform an extensive simulation study, verifying the performance of our proposed algorithms.

An important outcome of this study is that, although different from the optimal but computationally-intensive algorithm, the simple MIN-DELAY algorithm achieves excellent results in term of success probability for any reasonable deadline. In addition, we show that increasing the number of data copies beyond 10 does not provide any meaningful boost in performance.

As final comment, we note that our model can be extended to bus-bus communications by introducing some *virtual stops*, located in correspondence to possible physical contact points between two different lines. By appropriate choice of weights on the corresponding edges (e.g., no waiting time and high failure probability), one can capture the nature of this kind of communication as well. The main challenge, left for future research, is to locate the physical contact points and to bound their number so that the running times of the algorithms remain feasible.

## REFERENCES

- [1] U. Acer, P. Giaccone, D. Hay, G. Neglia, and S. Tarapiah. Timely data delivery in a realistic bus network. Technical report, 2010. available online at <http://www.tlc-networks.polito.it/~giaccone/infocom-techreport>.
- [2] A. Balasubramanian, B. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. *SIGCOMM Comput. Commun. Rev.*, 37(4):373–384, 2007.
- [3] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine. Relays, Base Stations, and Meshes: Enhancing Mobile Networks with Infrastructure. In *Proc. ACM Mobicom*, pages 81–91, San Francisco, CA, USA, September 2008.
- [4] C. B. Bin and C. M. Choon. MobTorrent: A Framework for Mobile Internet Access from Vehicles. In *Proceedings of INFOCOM*, 2009.
- [5] S. D. Boyles. Reliable routing with recourse in stochastic, time-dependent transportation networks. Master’s thesis, University of Texas at Austin, 2006.
- [6] S. D. Boyles. *Operational, Supply-Side Uncertainty in Transportation Networks: Causes, Effects, and Mitigation Strategies*. PhD thesis, University of Texas at Austin, 2009.
- [7] S. D. Boyles and S. T. Waller. The impact of nonlinear objective functions in online shortest paths. In *86th annual meeting of transportation research board*, 2007.
- [8] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. IEEE INFOCOM*, April 2006.
- [9] UMass DieselNet Research. <http://prisms.cs.umass.edu/dome/umassdieselnet>.
- [10] Y. Fan, R. Kalaba, and J. Moore. Arriving on time. *Journal of Optimization Theory and Applications*, 127(17):497–513, December 2005.
- [11] Y. Fan and Y. Nie. Optimal routing for maximizing the travel time reliability. *Networks and Spatial Economics*, 6(3):333–344, 2006.
- [12] H. Frank. Shortest Paths in Probabilistic Graphs. *Operations Research*, 17(4):583–599, 1969.
- [13] S. Gaito, D. Maggiorini, E. Pagani, and G. P. Rossi. Distance Vector Routing for Public Transportation Vehicular Networks: Performance Evaluation on a Real Topology. In *Proceedings of 2nd IFIP Wireless Days Conference*, Paris, France, 2009.
- [14] F. Giudici, D. Maggiorini, and G. P. Rossi. Performance Evaluation of Routing Protocols for Public Transportation Vehicular Networks. In *Proceedings of 1st IFIP Wireless Days Conference*, Dubai, United Arab Emirates, 2008.
- [15] S. Guo and S. Keshav. Fair and efficient scheduling in data ferrying networks. In *CoNEXT ’07: Proceedings of the 2007 ACM CoNEXT conference*, pages 1–12, New York, NY, USA, 2007. ACM.
- [16] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.
- [17] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and Inference Problems for Temporal Networks. In *Journal of Computer and System Sciences, Volume 64, Special issue on STOC 2000*, 2002.
- [18] C. Liu and J. Wu. Scalable routing in delay tolerant networks. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 51–60, Montreal, Quebec, Canada, 2007. ACM. TBR.
- [19] C. Liu and J. Wu. Routing in a cyclic mobispace. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 351–360, Hong Kong, Hong Kong, China, 2008. ACM. TBR.
- [20] C. Liu and J. Wu. An optimal probabilistic forwarding protocol in delay tolerant networks. In *MobiHoc ’09: Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pages 105–114, New York, NY, USA, 2009. ACM.
- [21] C. Liu, J. Wu, and I. Cardei. Message Forwarding in Cyclic MobiSpace: the Multi-copy Case. In *Proceedings of the Sixth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS’09)*, 2009.
- [22] E. Miller-Hooks. *Optimal routing in time-varying, stochastic networks: algorithms and implementations*. PhD thesis, University of Texas at Austin, 1997.
- [23] E. Miller-Hooks and H. Mahmassani. Path comparisons for a priori and time-adaptive decisions in stochastic, time-varying networks. *European Journal of Operational Research*, 146(1):67 – 82, 2003.
- [24] P. B. Mirchandani. Shortest distance and reliability of probabilistic networks. *Computers & Operations Research*, 3(4):347 – 355, 1976.
- [25] Y. a. Nie. Arriving-on-time problem: Discrete algorithm that ensures convergence. *Transportation Research Record: Journal of the Transportation Research Board*, 1964(1):193–200, 2006.
- [26] Y. M. Nie and X. Wu. Shortest path problem considering on-time arrival probability. *Transportation Research Part B: Methodological*, 43(6):597 – 613, 2009.
- [27] J. S. Provan. A polynomial-time algorithm to find shortest paths with recourse. *Networks*, 41(2):115–125, 2003.
- [28] M. Sede, X. Li, D. Li, M.-Y. Wu, M. Li, and W. Shu. Routing in large-scale buses ad hoc networks. In *Proc. of IEEE WCNC*, pages 2711–2716, 2008.
- [29] V. Cerf and S. Burleigh and A. Hooke and L. Torgerson and R. Durst and K. Scott and K. Fall and H. Weiss. RFC 4838: Delay-Tolerant Networking Architecture.
- [30] K. J. Wong, B. S. Lee, B. C. Seet, G. Liu, and L. Zhu. Busnet: Model and usage of regular traffic patterns in mobile ad hoc networks for inter-vehicular communications. In *Proceedings of ICT conference*, 2003.