Timely Data Delivery in a Realistic Bus Network

Timely Data Delivery in a Realistic Bus Network

Utku Günay Acer, *Member, IEEE*, Paolo Giaccone, *Member, IEEE*, David Hay, *Member, IEEE*, Giovanni Neglia, *Member, IEEE*, and Saed Tarapiah

***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 routing problem in such a network. Assuming the bus schedule is known, we maximize the delivery probability by a given deadline for each packet. Our approach takes the randomness into account, which stems from road traffic conditions, passengers boarding and alighting, and other factors that affect the 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. 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 our optimal algorithm shows the best performance, but it essentially reduces to minimizing the expected traversal time. When transmissions fail frequently (more than half of the times), the algorithm behaves similarly to a heuristic that maximizes the delivery probability over an infinite time-horizon. For reliable transmissions and values of deadlines close to the expected delivery time, the multi-copy extension requires only 10 copies to almost reach the performance of the costly flooding approach.

I. INTRODUCTION

We consider an opportunistic data network formed by (some) buses and bus stops in a town equipped with wireless devices, e.g. based on WiFi technologies, like in DieselNet [1]. Most of the stops act as disconnected relay nodes (the throwboxes in [2]), and a few of them are also connected to the Internet. Data are delivered across the town following the *store-carry-forward* network paradigm [3], 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 provides 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 connected to the power grid to provide lights and electronic displays. 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 may still provide strong probabilistic guarantees on data delivery time that are not common in opportunistic networks.

Given this scenario, 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, obtained from 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 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 in 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. In this paper, we focus on the other alternative, relying only on *bus-stop* communications. Sec. V-B provides some evidence that this second scenario may lead to better performance. We discuss how to extend our approach to include bus-bus communications in Sec. VI.

U.G. Acer is with Alcatel-Lucent Bell Laboratories, Antwerp, Belgium.

P. Giaccone is with Dipartimento di Elettronica, Politecnico di Torino, Turin, Italy. His activity was supported by the PRIN People-NET project funded by the Italian Ministry for Research and University (MIUR).

D. Hay is with School of Computer Science and Engineering, Hebrew University, Jerusalem, Israel. His activity was supported by the Legacy Heritage Fund program of the Israel Science Foundation (Grant No. 1816/10).

G. Neglia is with INRIA - EPI Maestro, Sophia Antipolis, France

S. Tarapiah is with Communication Engineering Department, An-Najah National University, Nablus, Palestine.

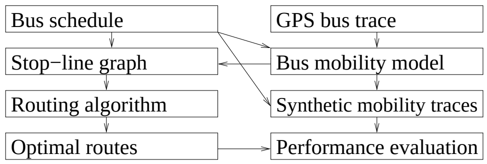

Fig. 1. High-Level Evaluation Framework

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-B), 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 extended metropolitan area through about 7,000 stops and 1,500 vehicles distributed among 250 lines, with more than 4,600 km of bus routes. These traces include the complete schedule for the morning rush hour period (6 AM–10 AM) and the corresponding GPS traces for the vehicles belonging to 26 lines.

A statistical analysis of these traces yields important conclusions, which allow 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-B) and then to extend it for the multi-copy case through a greedy approach (Sec. IV-D). In Sec. V we compare the performance of these proposed algorithms with three other heuristics (introduced in Sec. IV-C) that also operate on the stop-line graph: an adaptation of the routing algorithm proposed in [4] for bus-bus communications (we refer to it as MIN-HEADWAY), and the two naive algorithms, MIN-DELAY, that determines the path with the least expected traversal time, 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.

Additional material is presented in the appendices. The proofs of the performance bounds for multi-copy algorithms are in Appendix A. Appendix B presents an overview of bus mobility models in transportation literature, whereas Appendix C describes the algorithm we propose to generate the synthetic traces based on the actual schedule of the transportation system.

The paper provides the following main contributions: (i) Formulation of the original routing problem as a specific stochastic shortest path problem on a particular stochastic graph (Sec. IV-A). This formulation is justified by a statistical analysis of real transportation system traces (Sec. III-B). (ii) 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 a reasonable time, allowing each node to store an optimal pre-selected routing plan (Sec. IV-B). (iii) 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 (Sec. IV-D). (iv) Algorithm to generate mobility traces of the buses, based on their actual schedule (Appendix C). (v) Simulation analysis showing that the optimal algorithm mainly performs as MIN-DELAY, while it outperforms MIN-HEADWAY and MAX-PROB for reasonable values of packet loss probabilities. We provide some explanation for these results. In this sense the conclusion is that a naïve algorithm like MIN-DELAY *may be a very good heuristic for routing over realistic bus transportation networks* (Sec. V). (vi) Simulations showing that only 10 copies are needed for a multi-copy greedy approach to achieve a performance similar to that of flooding, which requires at least two order of magnitude more transmissions and copies for each single piece of data (Sec. V-A). (vii) Investigation of the effect of optimizing the location of the throwboxes covering many stops (Sec. V-C). (viii) Comparison between bus-to-bus and bus-to-stops communication paradigms (Sec. V-B).

II. RELATED WORK

Employing a bus network as a mobile backbone for dense vehicular networks was first proposed in [5], using standard routing protocols for mobile ad-hoc networks (e.g., DSR or AODV). More recently, the use of buses in a disconnected scenario has been considered; e.g. in the seminal *DieselNet* project [1]. Since our paper considers routing in such a network, in what follows we only mention work related to routing issues. Appendix B will be devoted to discuss previous work on bus mobility models.

Most of the research has focused on bus-bus communications [4], [6]–[9] 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, (e.g., as in [6], [7]) 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. Nevertheless, characterizing the bandwidth of the contacts and incorporating these constraints into our framework for bandwidth-hungry applications is part of our ongoing research.

The use of fixed relay nodes was also considered in [2], [10]. In [10], 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. [2] 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 the DieselNet testbed, support our proposed architecture that relies on opportunistic connectivity between vehicle nodes and fixed relays.

In order to provide low cost Internet connectivity to fixed kiosks in rural areas of developing counties, KioskNet architecture has been proposed [11]. In this architecture, buses carry data between the kiosks and a set of gateways that can communicate to a proxy on the Internet. Routing of such data between the kiosks and the gateways 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 kiosks [12].

The routing algorithms proposed by [13]–[16] are intrinsically more suited for bus-to-bus data transfers. [14] and [16] 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 value ranges from 30 minutes to 2 hours depending mainly on the length of the bus trajectory and on inactivity times at terminus. It is then comparable with the deadlines we are targeting, making it impossible to take advantage of such long term periodicity. Other forms of long-term regularities in the contact process of the different nodes [15] are too general for our settings, since we have significantly more information on the meetings that can be exploited to improve the performance. Finally, [13] 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 [10]). Only [8] 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 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 [17], [18]. Other optimality criteria are deviance [19], monotonic quadratic utility functions [20] and prospect-theory–based functions [21]. Recent and comprehensive surveys of the different utility functions and corresponding solutions appear in [22], [23]. 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 [24] and then was also investigated in [25]–[27] and more recently in [22], [28]–[30]. Current state-of-the-art algorithms still have exponential worst-case time complexity, based on enumerating over some set of candidate paths [22].

Our problem differs from Frank's problem essentially in three aspects. First, we consider a real transportation system and therefore we are not interested in the worst-case complexity of the algorithm on 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 detail 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. 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 (cf. [31]–[33] and references therein). The fact that finally minimizing the expected traversal time may provide almost optimal performance in some scenarios (when message transmissions do not fail) 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 the notation 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: $\text{traj}(v) = (s_0, s_1, \ldots s_n)$. In addition, each vehicle v is associated with a *trip*, denoted $\text{trip}(v)$, which is a time-stamped trajectory:

$$\text{trip}(v) = ((s_0, \tau_0), (s_1, \tau_1), \ldots (s_n, \tau_n)),$$

such that a vehicle v should arrive at stop s_i along its trajectory at time $\tau_i = \tau(v, s_i)$. We distinguish between the *scheduled time* τ_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 at a stop s_i, $t(v, s_i)$, and its corresponding scheduled arrival time $\tau(v, s_i)$ is the *lateness* of the vehicle at stop s_i, $l(v, s_i)$: $l(v, s_i) = t(v, s_i) - \tau(v, s_i)$. Note that the lateness is negative when the vehicle arrives earlier that its scheduled arrival. The *delay* between the stops s_i and s_j, $d(v, s_i, s_j)$, 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 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 $line(v) = \{v' \in \mathcal{V} \mid \text{traj}(v) = \text{traj}(v')\}$. Note that lines introduce an important characteristic of a bus transportation system: if a passenger misses a specific vehicle v, he/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. Communication Model

We assume that a bus is able to communicate with the throwbox at the stop only when it comes close to the stop, i.e. it is in the transmission range of the throwbox. In our model, 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 less important to specify the duration of the transmission opportunity between a bus and a stop. In practice, we assume:

Assumption 1: Transmission opportunities are *instantaneous* and occur at the arrival time of the bus at the stop position.

A drawback of this approach is that two overlapping transmission opportunities are artificially ordered and some transmission possibilities are lost. For example, if v_1 and v_2 can respectively transfer to s in $[t_1, t_3]$ and in $[t_2, t_4]$, with $t_1 < t_2 < t_3 < t_4$, data can be transferred in the two directions (from v_1 to v_2 and from v_2 to v_1), but when the transmission opportunities are ordered, only one direction is still feasible.

Furthermore, 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 opportunity, or contact duration being insufficient to transfer the data. We assume:

Assumption 2: Message success probabilities of different contacts are independent.

B. Measurements on Bus Mobility and their Implication

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 Appendix B. 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[1]. 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 which operates mainly buses but also trams and subway trains. The network consists of around 250 lines and a fleet of almost 1,500 vehicles. Some manual inspection is needed to be able to assign specific trip to their schedule (in order to evaluate metric like the lateness), so that we worked on a subset of the trace, consisting of 26 lines in both direction, with a total of 408 trips and 11,097 arrivals at bus stops.

Fig. 2 shows the empirical autocorrelation function for lateness, delay, and travel time. In particular, we have considered for each vehicle the sequence of latenesses at consecutive stops[2] $(l(s_0), l(s_1), \ldots, \ l(s_n), \ldots)$, the sequence of delays between consecutive stops $(d(s_0, s_1), d(s_1, s_2), \ldots, d(s_n, s_{n+1}), \ldots)$ and the sequence of travel times between consecutive stops $(t_1 - t_0, t_2 - t_1, \ldots, t_{n+1} -$

[1] This distinction is expressly advertised by Turin public transportation system, that label lines as frequency-based and schedule-based.

[2] With a slight abuse a notation, we omit the dependence on vehicle v, when it is clear from the context.

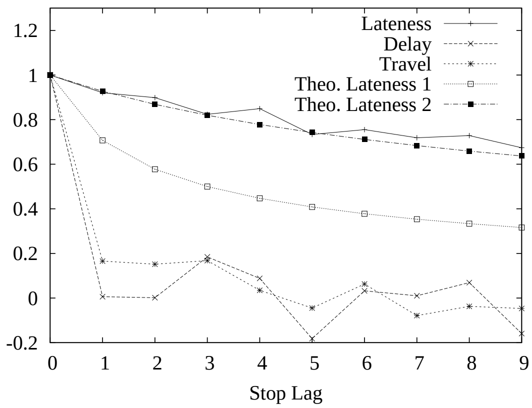

Fig. 2. Autocorrelation functions for lateness, delay and travel time.

$t_n, \ldots$). We have assumed that the sequences (relative to the same quantity) obtained for different vehicles are samples of the same random process, and we have used them to evaluate the empirical autocorrelation function. Fig. 2 demonstrates that the lateness values at consecutive stops are highly correlated. It is then clear that a simplistic bus mobility model, where the actual arrival time of vehicle v at stop s is equal to the scheduled one plus some noise that is independent from one stop to another ($t(v,s) = \tau(v,s) + n(v,s)$), is unrealistic. At the same time, we note that delays and travel times are significantly less correlated; this suggests the following model, in terms of travel time:

$$t(v, s_k) = \tau_0 + l(s_0) + \sum_{i=0}^{k} tt(v, s_i, s_{i+1}), \tag{1}$$

where we can assume that travel times are independent random variables (and then also delays are independent).

If we assume that delays are independent and identically distributed and that the lateness at the first stop $l(s_0)$ is distributed as $d(s_i, s_{i+1})$, it is possible to evaluate analytically the expression of the autocorrelation function. This is represented in Fig. 2 by the curve "theoretical lateness 1". We note that there is still a strong part of the correlation to be justified. A specific analysis of the lateness at the first stop shows that $l(s_0)$ is not distributed as $d(s_i, s_{i+1})$, and moreover its variance is almost 6 times larger. This shows that the variability of vehicle departure times is a significant component of the variability of arrival times at following stops. If we correct the expression of the autocorrelation function taking into account this empirical finding, we can obtain the new curve "theoretical lateness 2" that matches the empirical one very well.

As a conclusion of this statistical analysis, we assume in the rest of the paper that

Assumption 3: Bus travel times at consecutive stops are independent (but not necessarily identically distributed; in particular, their distribution will depend on the corresponding scheduled value).

We continue our statistical analysis by determining realistic distributions for the lateness at the first stop $l(s_0)$ and the delay distribution, in order to completely characterize the random

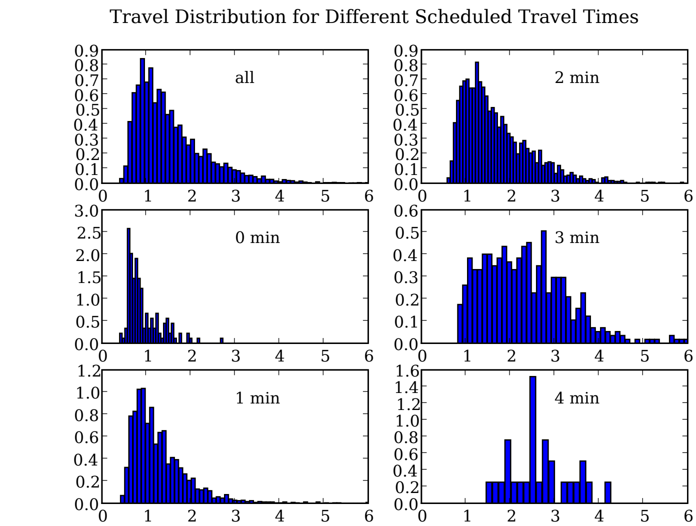

Fig. 3. Travel time distribution (aggregated and for different scheduled travel times).

variables of Eq. (1). This also allows us to use this recursive formula to generate realistic random traces (see Appendix C for the details). For example, Fig. 3 shows the empirical distribution of the travel times (assumed to be homogeneous across different lines) when all the samples are aggregated and when they are separated according to the corresponding scheduled travel times. It is evident that different distributions have to be used, depending on the different scheduled travel times. Since it is quite common in transportation literature to use the lognormal distribution to model travel times (see Appendix B), we have accepted this assumption and characterized the parameters of the lognormal distributions for different scheduled travel times by moment matching techniques.

Our final assumption concerns the waiting time at a stop when commuting from one line to another:

Assumption 4: 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 arrival line.

We note that Assumption 4, which plays an important role in enabling a graph representation with additive edge weights, is partially a consequence of Assumption 3. Indeed, consider buses moving according to the schedule, and a passenger transferring from line ℓ_1 to line ℓ_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 ℓ_1 vehicle and the departure time of the following ℓ_2 vehicle. But under Assumption 3, arrival times of ℓ_1 buses at stop s are random variables and so are the corresponding waiting times. Intuitively, if the variability of ℓ_1 arrival times is large[3] in comparison to the headway of line ℓ_2, the waiting time will have almost the same distribution of the waiting time seen by a Poisson observer, thus it is independent of ℓ_1 schedule.

[3]Note that, according to our model, the variance of the lateness increases along the trajectory and this condition tends to hold.

IV. Routing Algorithms in a Bus Network

As mentioned before, our routing algorithms aim to determine off-line routes for the transportation system that maximize 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 graph representation: computational complexity, intrinsic properties of the bus transportation system (namely, the existence of lines), characteristic of the stochastic process $t()$ (namely, waiting times at stops depend on the departing line), and an advantage coming from working with additive edge weights. For the sake of simplicity, in the following discussion we will first consider that all the transmissions are successful.

A. The Graph Representation

A simple way to represent the transportation system $\mathcal{T}$ is by a *temporal network* [34], that is a multi-graph whose set of nodes consists of $\mathcal{S} \cup \mathcal{V}$ (i.e., a node for each vehicle and for each stop) and each edge represents a transmission opportunity between a vehicle v and a stop s (or vice versa) occurring at the time instant $t(v, s)$ and can therefore be represented by the triple $\langle v, s, t(v, s) \rangle$ (or $\langle s, v, t(v, s) \rangle$). A possible route in such graph would then be a path connecting the source s_s and the destination s_d, i.e. a sequence of edges, like $(\langle s_s, v_0, t(v_0, s_s) \rangle$, $\langle v_0, s_1, t(v_0, s_1) \rangle$, $\langle s_1, v_1, t(v_1, s_1) \rangle$, $\ldots$, $\langle v_n, s_d, t(v_n, s_d) \rangle)$. This route is able to deliver the data from s_s to s_d, only if $t_{start} \leq t(v_0, s_s) \leq t(v_0, s_1) \leq t(v_1, s_1) \leq \ldots \leq t(v_n, s_d) \leq t_{stop}$.

While the temporal network is useful in general for deterministic scenarios, it is not suitable for the transportation system we are considering. The first reason is that, in a large-scale transportation network, this graph would have a very large number of nodes ($|\mathcal{S} \cup \mathcal{V}|$) and of edges. For example, if the time interval $[t_{start}, t_{stop}]$ spans a few hours, a stop in a dense traffic can exhibit hundreds of edges. The second reason is that it ignores the fact that in a bus network a vehicle in such route can be in some sense “replaced” by another vehicle of the same line. Finally, given our performance metric, we would need to evaluate $\Pr\{t_{start} \leq t(v_0, s_s)) \leq t(v_0, s_1)) \leq t(v_1, s_1) \leq \ldots \leq t(v_n, s_d) \leq t_{stop}\}$. However, the results of Sec. III-B show that lateness values at consecutive stops are strongly correlated, making it impossible to evaluate this probability in a simple way.

For these reasons it appears more beneficial to directly look for routes from the source to the destination in terms of lines. We can consider an alternative data structure, the *line-based graph* $G_{lines} = \langle \mathcal{S}, E_{lines} \rangle$, shown in Fig. 4.(a), in which nodes are bus stops and there is an edge between two stops s_i and s_j if and only if there is a line $\ell \in \mathcal{L}$ that goes from s_i to s_j (only stops which are served by at least two lines need to be considered for relay purposes). It is important to notice an intrinsic difference between the temporal network and the line-based graph: in the temporal network we check the feasibility of the path, by evaluating the probability that it maintains the chronological order between contacts. On the other hand, in the line-based graph, we are interested to check whether their total length (that is, the total traversal time of the path) is less than $t_{stop} - t_{start}$. Note that the traversal-time along a specific path is a random variable which is the sum of two kinds of random variables: edge random variables, which capture how travel time between two specific stops on a specific line is distributed, and node random variables, which capture the distribution of the waiting time at the stops.

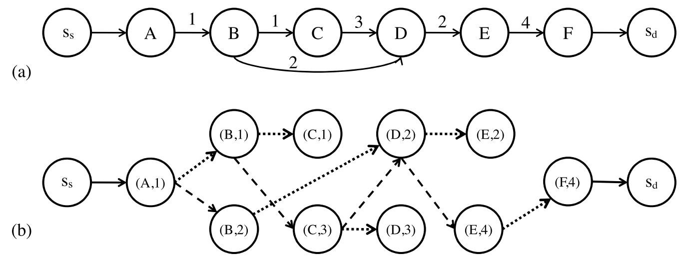

Fig. 4. (a) Example of line-based graph G_{lines} describing a bus network with stops $\mathcal{S} = \{A, B, C, D, E, F\}$ and lines $\mathcal{L} = \{1, 2, 3, 4\}$: the node corresponds to a stop and the label on the edge represents the line connecting the two stops. (b) The corresponding line-stop graph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

The waiting time at a stop poses a major difficulty on the design of a routing algorithm, because it is not simply related to the stop but it depends on the specific route under consideration, and more specifically on the stop's outgoing and incoming edges in that route. For example, if both edges correspond to the same line, the waiting time at the stop is 0. On the other hand, when switching lines at the stop, the waiting time depends only on the *headway* of the departing line by Assumption 4. Hence, this graph is also not well suited for our purposes.

In our proposed representation, which we call *stop-line graph* $G_{sl} = \langle V_{sl}, E_{sl} \rangle$, the nodes are (s, ℓ) pairs where s is a stop and ℓ is a line; $(s, \ell) \in V_{sl}$ if and only if line $\ell \in \mathcal{L}$ arrives at (or depart from) 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, ℓ) and (s', ℓ') corresponds to traveling from stop s to stop s' with line ℓ and then continuing from stop s' on line ℓ'. 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. 4.(b).

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 ℓ 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 kth bus of line ℓ'. To explain the formula for w_e, note that, to be able to forward the data successfully from one bus to another, two transmissions must succeed: the one from a bus of ℓ to s' (which may fail) and the one from s' to a bus of ℓ' (which will be successful after a geometric number of failures). The formula assumes that the transmission failure probability is the same for every possible transmission, but the model can be easily extended to consider the case where it depends on the stop and on the line to which the vehicle belong. We assume that all the random variables defining w_e are known (they will be characterized in Sec. IV-B); moreover, by Assumptions 3, 4 and 2, they are all independent.

It is important to notice that the stop-line graph G_{sl} provides a unified approach to deal with waiting times at the stops, thus solving shortcoming in previous approaches (e.g., temporal network [34], 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 simply calculate the overall traversal time of the data along a weighted path $\mathcal{P}$ as: $tr(\mathcal{P}) = \sum_{e\in\mathcal{P}} w_e$. When transmission failures can occur, the Cumulative Distribution Function (CDF) of the delivery time is scaled by a factor equal to $(1-p_f)$ for each transmission from a bus to a stop. Then the CDF of the delivery time along a given route has the horizontal asymptote $y = (1-p_f)^m$, where m is the total number of bus-to-stop transmissions in the route. Now, given the graph G_{sl}, the on-time delivery problem corresponds with finding a path $\mathcal{P}$ from s_s to s_d 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 [24], with the differences highlighted at the end of Sec. II.

B. Single-Copy Routing Algorithm and Implementation

We now propose 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 the (relative) deadline $t_{stop} - t_{start}$. For example, Fig. 5 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.

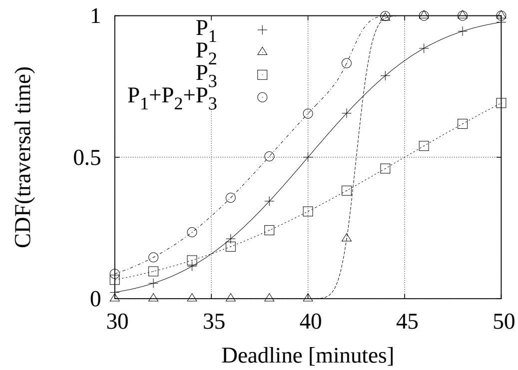

Fig. 5. Delivery probability CDFs of three disjoint paths P_1, P_2 and P_3, connecting a source and a destination with different traversal times and without transmission failures ($p_f = 0$). Path P_1 has the lowest expected traversal time; the variance of P_2 is the smallest, while P_3's variance is 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 concurrently.

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. Being that the traversal time is obtained by adding non-negative link weights, 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 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 on 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 a path prefix that the pruning mechanism cannot discard, 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 think 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 ℓ_1 then line ℓ_2, and then again line ℓ_1. We expect that these paths have normally worse performance than those where a data message just remains on line ℓ_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, $\mu_e = \text{E}[w_e \mid w_e{<}\infty]$ and $\sigma_e^2 = \text{Var}[w_e \mid w_e{<}\infty]$), 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 $\prod_{e\in P} p_e$. In the case of travel edges, average and variance of $tt(l, s, s')$ can be estimated directly from the traces. In the case of travel-switch edges, we have also to evaluate the average and variance of $wt(\ell, s, k)$ using the first three moments of the interarrival times of the line ℓ buses to stop s (which can be also estimated from the traces) and some basic Palm calculus [35].

For example, assuming perfect periodic bus arrivals with period δ and failure probability p_f, it can be shown that

$$E[wt(\ell, s, k)] = \delta(1/2 + p_f/(1 - p_f))$$
$$E[wt(\ell, s, k)^2] = \delta^2(1/3 + 2p_f/(1 - p_f)^2)$$

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 considered deadlines are, where "short" is in relation to the expected traversal time from s_s to s_d and not in an absolute sense.

C. 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, when the the transmission failure probability is null, MIN-DELAY is equivalent to ON-TIME(50) under the Gaussian assumption on the distribution of the traversal time. This is not true for different deadlines. For example Fig. 5 shows that path P_1, found by MIN-DELAY, does not always provide 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 $-\log(p_e)$. For high transmission failure probabilities, we can expect MAX-PROB and ON-TIME to select the same path. At the end of Sec. V we will show that this is the case.

Another approach, denoted MIN-HEADWAY, tries to minimize the sum of all lines headways along a path [4], 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 worst performance in our settings among all the different algorithms.

D. Extension to Multi-Copy Routing

As shown in the toy-case of Fig. 5, 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.

We consider only Multi-Copy schemes, such that at most k distinct copies of each data packet are present in the network at a given time instant. Without such a constraint a flooding scheme that copies the data whenever there is a contact, namely in an *epidemic manner*, would achieve the best possible delivery probability.

We propose a greedy Multi-Copy algorithm for on-time delivery problem, denoted simply as MC-ONTIME. It selects the k paths with the highest on-time delivery probability, without considering the interaction among 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 compare the current path prefix with the k-th best path discovered so far (rather than the best path). We can similarly extend the heuristics MIN-DELAY, MAX-PROB and MIN-HEADWAY presented in Sec. IV-C to respectively select the k paths with minimal expected traversal time, maximal success probability and minimal total headway.

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 overlap). 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 Appendix A, 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-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-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 have considered 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 can show that the worst-case performance of MC-ONTIME-DISJOINT is the same as MC-ONTIME. Moreover, some preliminary simulations have shown that MC-ONTIME is superior in practice, and therefore this is the multi-copy routing algorithm we consider in the sequel[4].

V. PERFORMANCE EVALUATION

We consider a set of 180 source-destination (s_s–s_d) stop pairs among the 2550 stops inside the metropolitan area of Turin. 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 at a main transportation hub within the city center (close to the main train station of Turin), and all the destinations s_d have been chosen uniformly at random.

To reduce deployment costs, we assume to employ one single throwbox covering close by stops. Hence, stops are aggregated after setting the transmission range of each throwbox equal to 100m; Sec. V-C discusses the effect of the transmission range on the total number of throwboxes to deploy.

We generate a set of 100 traces with the parameters obtained by the statistical analysis. The traces include the trips of all vehicles of 250 bus lines for the four hours available from the schedule. Appendix C discusses in details the trace generation process. We have developed a simulator that computes the delivery probability of each path by averaging across these 100 traces; note that the real-life trace alone would not be enough to compute this probability with any accuracy. Moreover this trace includes only a small fraction of the lines in Turin, and the number of possible paths between a source and a destination using only this subset of lines is drastically reduced. Data is assumed to be available at the source stop at 7 AM.

[4]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.

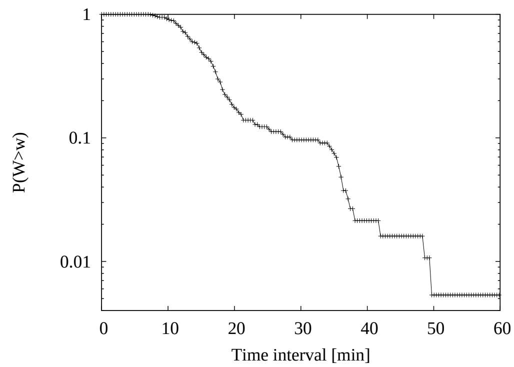

Fig. 6. Complementary CDF of the critical time window W guaranteeing on-time delivery probability $\in [0.1, 0.9]$ for the minimum expected traversal-time path.

As we mentioned in Sec. IV-B, for very short deadlines, there is probably no route that could deliver the packet with a reasonable probability, while for very long deadlines, many different routes are able to deliver it with probability almost one (if all the transmissions succeed). Then, it exists an interval of deadline values for which it makes sense to “spend effort” to determine good routes. In order to quantify this interval, we introduce the *“critical” time window* of a route, defined as $W = \phi(90) - \phi(10)$: this is the amplitude of the interval of deadlines for which the route determined by ONTIME(50) achieves delivery probabilities in $[0.1, 0.9]$. Fig. 6 shows the complementary CDF of W, for 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.

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, considering that there are some differences between the model and the synthetic traces. In fact, in our model we considered a constant line frequency in the time period (while there are some small changes in the schedule and then in the synthetic traces), and the same headway distribution at each stop along the trajectory (while for example the headway variability is larger for the last stops than for the first ones). Moreover, in the synthetic traces we made sure that two buses of the same line cannot overtake each other (see Appendix C for the details). This introduces some further inhomogeneity that is not taken into account in the model.

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). We first assume that transmissions are reliable, i.e. $p_f = 0$. Recall that in this case MIN-DELAY is equivalent to ON-

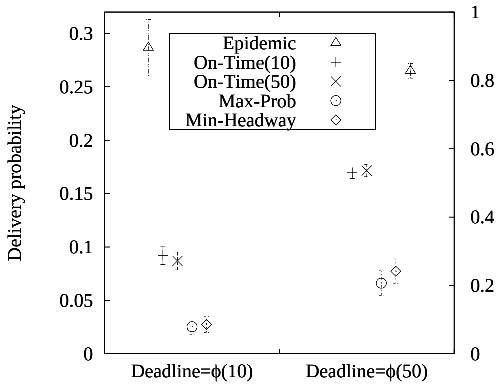

Fig. 7. Delivery probability (average and 90% confidence interval) for two deadlines and different routing algorithms, for reliable transmission ($p_f = 0$). MIN-DELAY is the same as ON-TIME(50).

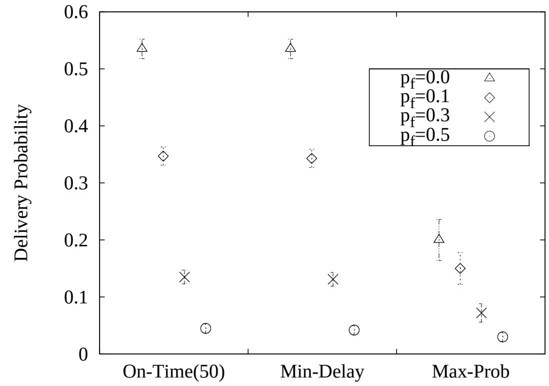

Fig. 8. Delivery probability (average and 90% confidence interval) for ON-TIME(50), MIN-DELAY and MAX-PROB for deadline $\phi(50)$ and for different values of transmission failure probability p_f.

TIME(50). 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. We will report the results only for $x = 10$ ("short deadline") and $x = 50$ ("average deadline"), since these cases are representative.

Fig. 7 compares the delivery probability of the different algorithms for the two deadlines. The gain of EPIDEMIC with respect to all the other single-copy algorithms decreases as the deadline increases: EPIDEMIC achieves a delivery probability 3 times larger than ON-TIME for deadline $\phi(10)$, but only 1.5 times larger for deadline $\phi(50)$. Indeed, when the deadline is large enough just one copy of the data is enough in order to reach 100% delivery probability. In such a case, EPIDEMIC does not introduce any gain in terms of performance, and the cost in terms of copies and transmissions is much larger than under single-copy algorithms. For example we observed on average more than 600 copies for $\phi(10)$ and more than 900 copies for $\phi(50)$ under EPIDEMIC up to the deadline, while for all single-copy algorithms the number of transmissions for each data is on average 5.0, 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 results is that in many cases (121 out of 180) ON-TIME(10) performs *exactly* as ON-TIME(50) (or, equivalently, as MIN-DELAY). In fact 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 deadline values: The optimal route is not very sensitive to the deadline. In most of the cases the best path computed by ON-TIME(50) is the best for every deadline $\phi(x)$ with $x \in [0, 100]$. Recall the example in Fig. 5, showing that the best path does in general depend on the deadline. Our experiments lead us to conclude that these cases are very rare in a real transportation system when transmissions always succeed. 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.

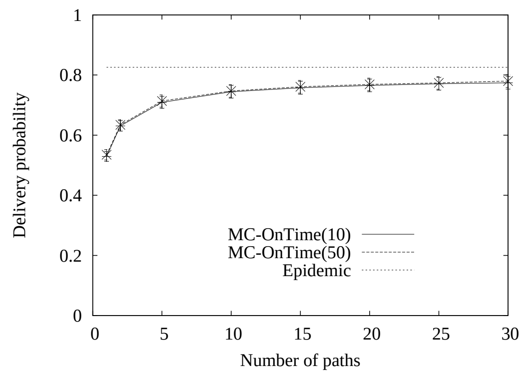

Fig. 9. Delivery probability (average and 90% confidence interval) vs. number of paths for deadline $\phi(50)$ and for multi-copy routing and no transmission failures ($p_f = 0$).

We now investigate the effect of transmission failures. Fig. 8 shows the delivery probability for different values of transmission failure probability p_f. Even with failures, ON-TIME(50) and MIN-DELAY behave similarly. Only when p_f increases, MAX-PROB shows an average delivery probability comparable to the two other algorithms; indeed, MAX-PROB becomes more efficient when the transmission failures are high, since the optimal policy tends to minimize the number of transmissions. Hence, all the algorithms appear to behave efficiently for large p_f.

A. Multi-copy routing

We turn now to deal with multi-copy settings. Fig. 9 shows the performance of the MC-ONTIME(x) policy, that takes advantage of the k paths with the highest delivery probability by the deadline $\phi(x)$. The figure shows the results obtained for all the 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 about 66% of that achieved by EPIDEMIC, and a few more copies significantly reduces the performance gap. Yet, after 10 copies we observe only a negligible improvement. This is partially

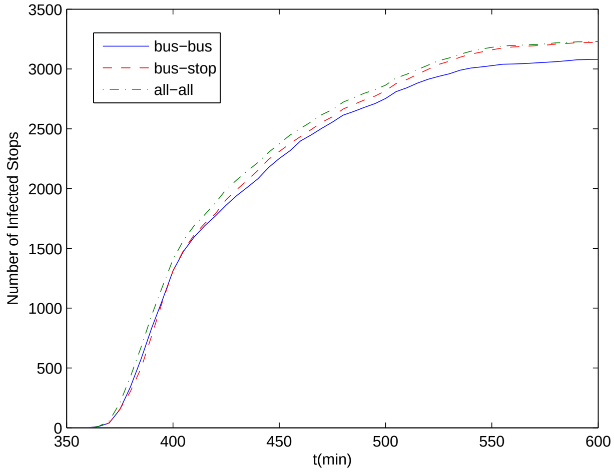

Fig. 10. Comparison of Epidemic Routing with bus-bus & bus-stop communications

due to the fact that MC-ONTIME exploits a given sequence of paths provided by the algorithms, whose internal "diversity" 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 is 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 at least one order of magnitude larger than the multicopy approach using a pre-selected subset of 10 paths.

B. Comparison of Bus-to-Bus vs. Bus-to-Stop Communications

Prior work on bus-based delay tolerant networks mainly has focused on exploiting opportunistic contacts between the buses. In our scenario, we utilize bus-stop contacts rather that bus-bus ones. In this section, we provide a first comparison of the two approaches in terms of the speed of message propagation when flooding is used.

Fig. 10 shows how fast epidemic routing diffuses a message in the network if it utilizes only stop-bus communication opportunities, only bus-bus communication opportunities or both (referred to as all-all). In all cases, we consider that the message is generated by a user located at a main bus stop in Turin and copied to all the buses that go through the stop. In order to compare the message spreading speed in the different scenarios, we have considered the stops themselves as potential destinations. Fig. 10 shows the number of stops that are reached by a copy of the message over time. It appears that using bus-stop communications is more effective than using only buses and achieves almost the same performance of the all-all scenario. In particular, we observe that not all the stops can be reached when we rely only on buses without using the stops as fixed relays. On the contrary, the bus-bus communication scenario seems to be slightly faster immediately after the generation of the message.

It is important to note that this evaluation neglects factors such as contact duration, physical/link layer constraints, etc. Burgess *et al.* report that the contact duration between two mobile nodes in a vehicular network may eventually be too short to transfer a message [6]. These effects are expected to be more significant in the bus-bus scenario when both nodes are mobile. On the contrary, in the bus-stop scenario, contact opportunities can be quite long, also because buses need to stop to let passengers board and alight. Hence, we expect that the performance gain of the bus-stop scenario in comparison to the bus-bus one increases when these other effects are considered.

TABLE I
EFFECT OF AGGREGATING STOPS IN THE ENTIRE TOWN

TX-RANGE	THROWBOXES	REDUCTION RATIO
1m	6868	0.00
50m	4385	0.36
100m	3464	0.49
150m	3055	0.56
200m	2752	0.60
250m	2394	0.75

C. Bus stops aggregation

An urban area transportation network typically contains a large number of stops. For example, there are 6868 stops in Turin, among which 2550 are in the metropolitan area. It may be unfeasible to install a throwbox at each of these stops. At the same time this may not be necessary. In fact, many of these stops are close to each other so that a single throwbox can be used to cover multiple of them. In this section we quantify how many stops can be "aggregated", in the sense that they are close enough to use a unique wireless box for all of them.

One parameter used in aggregation is the communication range of wireless boxes, d_{tx}, and it is assumed to be homogenous across all the stops. Two nodes are considered neighbors if their distance is less than d_{tx}. We use a simple, greedy heuristic algorithm to group closeby stops. Let $\mathcal{N}$ be the set of stops. We first pick the node, say it s^M, that has the largest number of neighbors. We place one wireless box at s^M. Then, we remove s^M and the neighbors of s^M from $\mathcal{N}$. We iterate this step until $\mathcal{N}$ is empty.

The number of stops groups, or number of wireless boxes that should be installed in order to cover all the 6868 stops present in Turin is shown in Table I for different values of the transmission range; the reduction is evaluated as the ratio between the number of stops that so not need a throwbox and the total number of stops. As the communication range increases, a larger number of stops are grouped together. Hence, the number of wireless boxes required to provide coverage decreases and the reduction ratio of the aggregation increases. We see that even a 100-meter communication range can result in a large reduction (almost 50%) in the number of throwboxes (i.e., in hardware cost) to deploy the bus-based DTN.

Table II shows the effect of aggregation when only the 2550 stops in the metropolitan area of Turin are considered. Because

TABLE II
EFFECT OF AGGREGATING STOPS IN THE MAIN METROPOLITAN AREA

TX-RANGE	THROWBOXES	REDUCTION RATIO
1m	2550	0.00
50m	1552	0.39
100m	1084	0.57
150m	844	0.67
200m	687	0.73
250m	517	0.80

the stops are more closely placed in the metropolitan area, the aggregation mechanism with the same transmission range results in a larger reduction ratio for the number of wireless access points that need to be deployed.

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, when transmissions succeed all the time. 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 time of the algorithm remain feasible.

In future work, we also plan to extend our approach by releasing some or most of our assumptions discussed in Sec. III.

APPENDIX A
TIGHT BOUNDS ON THE PERFORMANCE OF MULTI-COPY ALGORITHMS

In this section we provide the proof for Theorem 1 of Section IV-D, which deals with the performance of the multi-copy MC-ONTIME algorithm. This algorithm computes the success probability of all paths in isolation and chose the k best paths (without considering the interaction between them). Theorem 1 comprises of the following lower- and upper-bounds.

Lemma 1: The MC-ONTIME heuristic always achieves at least $1/k$ of the success probability of an optimal k multicopy heuristic.

Proof: Let $p_1, \dots p_k$ be the success probability of the paths selected by the MC-ONTIME algorithm, such that p_i corresponds to the path selected at iteration i. Let $q_1, \dots, q_k$ be the success probability of the paths selected by the optimal algorithm, and by $Q_1, \dots Q_k$ the corresponding events of successful delivery (namely, $\Pr[Q_i] = q_i$). Note that by definition, $p_1 \geq \max_i q_i$. Thus,

$$\Pr[\text{GREEDY succeeds}] \geq p_1 \geq \frac{1}{k}\sum_{i=1}^{k} q_i \geq \frac{1}{k}\Pr\left[\bigvee_{i=1}^{k} Q_i\right] = \frac{1}{k}\Pr[\text{The optimal algorithm succeeds}],$$

where the third inequality is due to the union bound. ■

Lemma 2: There is a valid transportation graph for which MC-ONTIME achieves at most $\frac{1}{(1-\varepsilon)k}$ of the success probability of an optimal k multicopy algorithm, for arbitrarily small $\varepsilon > 0$.

Proof: Consider a transportation graph in which, from the source to the destination, there are $2k$ paths:

- k two-edge paths, which share their first edge. The probability to traverse this first edge is p while the probability to traverse the second edge is $1 - \varepsilon/4$.
- k single-edge paths, such the probability to traverse the edge is $p(1 - \varepsilon/2)$.

Assume $p = \varepsilon/((k-1)(1-\frac{\varepsilon}{2})^2)$. The MC-ONTIME algorithm will choose the first k paths, since $p(1-\varepsilon/4) > p(1-\varepsilon/2)$. Since all these paths need to traverse the first edge, the probability that MC-ONTIME succeeds is at most p.

On the other hand, the optimal algorithm will do better than the algorithm that chooses the last k paths. The inclusion-exclusion principle (a.k.a Bonferroni inequality) yields that the success probability of the optimal algorithm is at least

$$kp(1-\varepsilon/2) - \binom{k}{2} p^2 (1-\varepsilon/2)^2.$$

This implies that the ratio between the success probability is at most

$$\frac{p}{kp(1-\frac{\varepsilon}{2}) - \binom{k}{2}p^2(1-\frac{\varepsilon}{2})^2} = \frac{1}{k(1-\varepsilon)}.$$

■

APPENDIX B
BUS MOBILITY MODELS IN TRANSPORTATION

This investigation of the transportation literature is mainly based on the overviews in [36], [37].

Some works provide probability distribution for arrival time or lateness or delay, based on empirical studies (e.g. [38]–[42] or on model simplification (e.g., [43], [44]). Most studies use a skewed distribution for the lateness, since it is more likely to be behind schedule than ahead. Lognormal or gamma random variables are the most common assumptions (see the summary table in [36]).

About the statistical dependency of these quantities, contrasting effects hold. In general once a bus with low headway is late at a given stop, it is difficult to recover its lateness. In fact, for lines with low headways, passengers usually do not regulate their arrival on the basis of the schedule. Hence, passenger arrival can be assumed to be a Poisson process. When a bus is late, the longer waiting time at following stops causes an increase in the number of passengers who board (and later alight) resulting in longer dwell times and higher and higher delay en-route. Therefore, lateness and delay are positively correlated in such cases: high lateness at a stop results in increased delay over the subsequent segment [38]. This phenomenon does not always occur on buses with higher headway. In fact, passengers now tend to arrive just before the scheduled departure time of desired bus. Hence, late buses do not board significantly more passengers than on-time buses. Furthermore, since higher headway buses often have slack built into their schedule, there is opportunity to recover some of the lost time [45]. Penalties to drivers for being excessively late encourage them to catch up to the schedule. Thus, the delay in a segment is negatively correlated with the lateness at the start of the segment. Because of these two phenomena, the delay on a bus line segment can either be negatively or positively correlated with the lateness at the start of the segment, depending in large part on the line headway. Moreover, we observe that the lateness of a bus also has consequences on following buses on the same line and direction. A late bus boards more passengers, and so it leaves less of them for the following bus. This effect would lead to a negative correlation between the lateness of consecutive buses. At the same time in many cases transport agency policies or traffic conditions make overtaking impossible or quite rare. Hence a bus that is significantly late would cause also the consecutive ones to be late.

Regarding dwell time, this can be a significant part of the total service time (up to 16% of the total service time according to [37]). This time clearly depends on the number of passengers boarding and alighting (empirical formulas are proposed in [46] and [47]), but also on the crowding, fare types [40], payment modalities, bus design (separate/common doors for boarding and alighting), mode (i.e. bus or metro lines) and service type[5] [48]. Also, the contribution of dwell time to lateness correlation is not immediate. For example a large dwell time can be due to a large number of passengers boarding or alighting. In the first case the alighting at following stops will in general large, in the second will be small.

APPENDIX C
GENERATION OF TUNABLE SYNTHETIC TRACES

The real traces from *GTT (Gruppo Torinese Trasporti)* transportation network provide data about 26 lines that operate in the city. The variety in the traces allow us to model properties like headway, travel time and lateness. However, real traces cannot be used in performance evaluation because they cover only a small fraction of the lines in Turin, and the number of possible paths between a source and a destination using only this subset of lines is drastically reduced. In this case, we expect different routing algorithms to select the same path, and it would not be possible to really evaluate the potential gain -but also the computational complexity- of our approach in comparison to simpler ones. Moreover, we are interested to evaluate the best performance achievable by a bus-based DTN, so we want to consider a massive deployment of WiFi-enabled buses and stops in the metropolitan area. For these reasons, we have decided to rely on synthetic traces describing the mobility of all the buses in the network. In this section, we explain how these synthetic traces have been generated on the basis of our statistical analysis of the real data, discussed in Section III.

For each vehicle we generate the sequence of the arrival times at the different stops in its trajectory according to Eq. (1):

$$t(v, s_k) = \tau_0 + l(v, s_0) + \sum_{i=0}^{k-1} tt(v, s_i, s_{i+1}),$$

where τ_0 is the scheduled departure time of the vehicle from the first stop; the lateness at the first stop, $l(v, s_0)$, and the travel times, $tt(v, s_i, s_{i+1})$, are assumed to be independent random variables (see Section III). For the lateness at the first stop we have assumed a triangular distribution with support $[-2, +2]$ minutes, that resembles the empirical distribution we observed. Travel times are assumed to have a truncated lognormal distribution, as it is common in transportation literature. The truncated lognormal distribution is completely characterized by its mean (M_{TT}), its variance (V_{TT}) and its maximum value (th). We have assumed that these quantities depend only on the corresponding scheduled travel time (STT). By trying to match the moments of the empirical distribution and the lognormal distribution, we have identified the following empirical relations:

$$\begin{aligned} M_{TT} &= 0.7(STT + 0.5) \quad [\text{min}] && (2)\\ V_{TT} &= \frac{M_{TT}^2}{4} \quad [\text{min}^2] && (3)\\ th &= 2M_{TT}. && (4) \end{aligned}$$

Occasionally we have been forced to increase artificially th in order to be able to guarantee the headway constraint (see below).

The arrival times at the stops could be calculated independently for each bus generating the random variables $l(v, s_0)$ and $tt(v, s_i, s_{i+1})$ (whose parameters can all be evaluated from the schedule using (2) and (3)) and summing them according to (1), or equivalently in the following iterative way:

$$\begin{cases} t(v, s_0) = \tau_0 + l(v, s_0), \\ t(v, s_i) = t(v, s_{i-1}) + tt(v, s_{i-1}, s_i). \end{cases} \qquad (5)$$

This procedure can produce synthetic traces where a bus can overtake another bus serving the same line, for example when the second is particularly late. This happens with higher probability the smaller is the headway of the line in comparison to the scheduled travel times. Being that we do not observe this phenomenon in the real traces, we want to avoid it also in the synthetic ones by introducing the *headway constraint*: buses belonging to the same line arrive at each stop in the same order.

[5]Service type can be rapid, limited, local, or combined depending on the vehicle speed, and the distance between consecutive stops.

To satisfy such constraint whilst matching the empirical values for the first two moments (M_{TT} and V_{TT}) for the travel time, we have introduced a "virtual queue" for each different distribution considered, hence one for each possible scheduled travel time value in the transportation network, and one for the lateness at the first stop. Initially such queues are all empty. The arrival times of buses belonging to the same line are generated orderly according to their schedules. Let $t(v_j, s_k)$ be the arrival time of the j^{th} bus of a particular line at the k^{th} stop. We want to guarantee that $t(v_j, s_k) > t(v_i, s_k)$, for each $j > i$ and for each k. All the random variables are generated and used as described above as long as $t(v_j, s_k) > t(v_{j-1}, s_k)$. If this is not the case, a larger value is needed and the last random variable is generated again until the condition is satisfied. If the random values generated and not used would be simply discarded, the actual distribution would be stochastically larger than the intended theoretical distribution. For this reason, all the values generated but not used are put in the corresponding queue and we try to recycle them later. In fact, once a queue is no more empty, when a value from the corresponding distribution is needed in order to generate a new arrival time, e.g. $t(v_l, s_h)$, it will be taken among those in the queue, that are large enough to guarantee that the bus order is respected, i.e. that $t(v_l, s_h) > t(v_{l-1}, s_h)$. If the associated queue is empty, a new value is generated as described above (and possibly added to the queue if it is not large enough). If the number of values in each queue at the end of the trace generation is small, the distortion between the actual and the theoretical distributions is small. In our case, we have observed that at the end of each trace generation only a small number of values (less than 0.7%) remains in the queues in comparison to the total number of generated random variables.

Acknowledgment

The authors would like to thank both dr. Fabrizio Arneodo and dr. Rafael Aguilar from 5T (Tecnologie Telematiche Trasporti Traffico Torino) for the traces they provided us and for their availability to explain and discuss the details of Turin public transportation system.

References

[1] "UMass DieselNet Research," http://prisms.cs.umass.edu/dome/umassdieselnet.

[2] 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*, San Francisco, CA, USA, September 2008, pp. 81–91. [Online]. Available: http://prisms.cs.umass.edu/mcorner/papers/RMB_mobicom.pdf

[3] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "RFC 4838: Delay-Tolerant Networking Architecture." [Online]. Available: http://tools.ietf.org/html/rfc4838

[4] 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*, 2008, pp. 2711–2716.

[5] 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.

[6] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," in *Proc. IEEE INFOCOM*, April 2006. [Online]. Available: http://prisms.cs.umass.edu/brian/pubs/burgess.infocom2006.pdf

[7] A. Balasubramanian, B. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 373–384, 2007.

[8] 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.

[9] 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.

[10] C. B. Bin and C. M. Choon, "MobTorrent: A Framework for Mobile Internet Access from Vehicles," in *Proceedings of INFOCOM*, 2009.

[11] S. Guo, M. H. Falaki, U. Ismail, E. A. Oliver, S. U. Rahman, A. Seth, M. A. Zaharia, and S. Keshav, "Design and implementation of the kiosknet system (extended version)," David R. Cheriton School of Computer Science, University of Waterloo, Tech. Rep. CS-2007-40, 2007.

[12] S. Guo and S. Keshav, "Fair and efficient scheduling in data ferrying networks," in *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*. New York, NY, USA: ACM, 2007, pp. 1–12.

[13] 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*. Montreal, Quebec, Canada: ACM, 2007, pp. 51–60, tBR. [Online]. Available: http://portal.acm.org/citation.cfm?id=1288107.1288115

[14] ——, "Routing in a cyclic mobispace," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. Hong Kong, Hong Kong, China: ACM, 2008, pp. 351–360, tBR. [Online]. Available: http://portal.acm.org/citation.cfm?id=1374665

[15] ——, "An optimal probabilistic forwarding protocolin delay tolerant networks," in *MobiHoc '09: Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2009, pp. 105–114.

[16] 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.

[17] S. T. Waller and A. K. Ziliaskopoulos, "On the online shortest path problem with limited arc cost dependencies," *Networks*, vol. 40, no. 4, pp. 216–227, 2002.

[18] J. S. Provan, "A polynomial-time algorithm to find shortest paths with recourse," *Networks*, vol. 41, no. 2, pp. 115–125, 2003.

[19] S. D. Boyles, "Reliable routing with recourse in stochastic, time-dependent transportation networks," Master's thesis, University of Texas at Austin, 2006.

[20] 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.

[21] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," *Econometrica*, vol. 47, no. 2, pp. 263–291, 1979.

[22] Y. M. Nie and X. Wu, "Shortest path problem considering on-time arrival probability," *Transportation Research Part B: Methodological*, vol. 43, no. 6, pp. 597 – 613, 2009.

[23] S. D. Boyles, "Operational, supply-side uncertainty in transportation networks: Causes, effects, and mitigation strategies," Ph.D. dissertation, University of Texas at Austin, 2009.

[24] H. Frank, "Shortest Paths in Probabilistic Graphs," *Operations Research*, vol. 17, no. 4, pp. 583–599, 1969.

[25] P. B. Mirchandani, "Shortest distance and reliability of probabilistic networks," *Computers & Operations Research*, vol. 3, no. 4, pp. 347 – 355, 1976.

[26] E. Miller-Hooks, "Optimal routing in time-varying, stochastic networks: algorithms and implementations," Ph.D. dissertation, University of Texas at Austin, 1997.

[27] 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*, vol. 146, no. 1, pp. 67 – 82, 2003.

[28] Y. Fan, R. Kalaba, and J. Moore, "Arriving on time," *Journal of Optimization Theory and Applications*, vol. 127, no. 17, pp. 497–513, December 2005.

[29] Y. Fan and Y. Nie, "Optimal routing for maximizing the travel time reliability," *Networks and Spatial Economics*, vol. 6, no. 3, pp. 333–344, 2006.

[30] Y. a. Nie, "Arriving-on-time problem: Discrete algorithm that ensures convergence," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1964, no. 1, pp. 193–200, 2006.

[31] S. Vivas and G. R. no, “Mobile device programming for finding optimal path on a fast-bus system,” in *IEEE SIEDS*, 2006, pp. 196–201.

[32] J. Boyan and M. Mitzenmacher, “Improved results for route planning in stochastic transportation networks,” in *In Proc. of Symposium of Discrete Algorithms*, 2001, pp. 895–902.

[33] M. Datar and A. G. Ranade, “Commuting with delay prone buses,” in *In Proc. of Symposium of Discrete Algorithms*, 2000, pp. 22–29.

[34] 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.

[35] F. Baccelli and P. Bremaud, *Elements of Queueing Theory: Palm Martingale Calculus and Stochastic Recurrences*. Springer, 2010.

[36] M. Dessouky, A. Singh, and R. Hall, “Transit ITS Simulator (TRANS-ITS): Design Document,” University of California, Berkekey, Tech. Rep. UCB-ITS-PWP-97-26, 1997.

[37] R. L. Bertini and A. M. El-Geneidy, “Modeling Transit Trip Time Using Archived Bus Dispatch System Data,” *Journal of transportation engineering*, vol. 130, pp. 56–67, 2004.

[38] M. A. Turnquist, “A model for investigating the effect of service frequency and reliability on bus passenger waiting times,” *Transportation Research Record*, vol. 663, pp. 70–73, 1978.

[39] W. Talley and A. Becker, “On-Time performance and the exponential probability distribution,” *Transportation Research Record*, vol. 1198, pp. 22–26, 1987.

[40] R. Guenthner and K. Hamat., “Distribution of Bus Transit On-Time Performance,” *Transportation Research Record*, vol. 1202, pp. 1–8, 1988.

[41] P. N. Seneviratne, “Analysis of On-Time Performance of Bus Service Using Simulation,” *Journal of Transportation Engineering*, vol. 116, pp. 517–531, 1990.

[42] J. G. Strathman and J. R. Hopper, “Empirical Analysis of Bus Transit On-Time Performance,” *Journal of Transportation Engineering*, vol. A27, pp. 93–100, 1993.

[43] P.-A. Andersson and G.-P. Scalia-Tomba, “A mathematical model of an urban bus route,” *Transportation Research Part B: Methodological*, vol. 15, no. 4, pp. 249–266, August 1981. [Online]. Available: http://ideas.repec.org/a/eee/transb/v15y1981i4p249-266.html

[44] R. W. Hall, “Vehicle Scheduling at a Transportation Terminal with Random Delay en Route,” *Transportation Science*, vol. 19, pp. 308–320, 1985.

[45] R. Hall, M. Dessouky, A. Nowroozi, and A. Singh, “Evaluation Of ITS Technology For Bus Timed Transfers,” University of California, Berkekey, Tech. Rep. UCB-ITS-PRR-97-37, 1997.

[46] H. Levinson, “Analyzing transit travel time performance,” *Transportation Research Record*, vol. 915, pp. 1–6, 1983.

[47] R. Guenthner and K. Sinha., “Modeling Bus Delays due to Passenger Boardings and Alightings,” *Transportation Research Record*, vol. 915, pp. 7–13, 1983.

[48] L. T. and N. H. Wilson, “Dwell time relationships for light rail systems,” *Transportation Research Record*, pp. 287–295, 1992.

PLACE PHOTO HERE

Utku Günay Acer (M’09) received his B.S. degree in telecommunications engineering with honors from Sabancı University, Turkey in 2004 and his M.S. and Ph.D. degrees from the Electrical, Computer and Systems Engineering Department of Rensselaer Polytechnic Institute in 2005 and 2009, respectively. He is currently a researcher with Alcatel-Lucent Bell Laboratories, Antwerp, Belgium. From 2009 to 2010, he was a post-doctoral researcher with the National Research Institute of Informatics and Control (INRIA), Sophia-Antipolis, France.

PLACE PHOTO HERE

Paolo Giaccone (M’98) received the Dr.Ing. and Ph.D. degrees in telecommunications engineering from Politecnico di Torino, Italy, in 1998 and 2001, respectively. He is currently an Assistant Professor in the Department of Electronics, Politecnico di Torino. During the summer of 1998, he was with the High Speed Networks Research Group, Lucent Technology-Bell Labs, Holmdel, NJ. During 2000-2001 and in 2002 he was with the Information Systems Networking Lab, Electrical Engineering Dept., Stanford University, Stanford, CA. His main area of interest is the design of network algorithms, the theory of interconnection networks, and the performance evaluation of telecommunication networks through simulative and analytical methods.

PLACE PHOTO HERE

David Hay (M’09) received his BA (summa cum laude) and PhD degree in computer science from the Technion-Israel Institute of Technology in 2001 and 2007, respectively. He is currently a senior lecturer (assistant professor) at the Rachel and Selim Benin School of Computer Science and Engineering, Hebrew University, Jerusalem, Israel. Prior to joining the Hebrew University, David Hay was with IBM Haifa Research Labs (1999-2002), Cisco Systems (2006),Ben-Gurion University of the Negev (2007-2008), Politecnico di Torino (2008-2009), and Columbia University (2009-2010). His main research interests are algorithmic aspects of high-performance switches and routersin particular, QoS provisioning, competitive analysis, and packet classification.

PLACE PHOTO HERE

Giovanni Neglia (M’02) is a researcher in Maestro team at INRIA. He received a master’s degree in electronic engineering and a Ph.D. degree in telecommunications from the University of Palermo, Italy, in 2001 and 2005, respectively. In 2005 he was a research scholar at the University of Massachusetts Amherst, visiting the Computer Networks Research Group. Before joining INRIA as a permanent member in September 2008, he was a postdoc at the University of Palermo (from June 2005) and an external scientific advisor in Maestro team (from September 2006). His research focuses on modeling and performance evaluation of computer networks and proposals of new mechanisms to improve their performance. To this end he applied different mathematical tools (Markov processes, control theory, fluid models, game theory, gradient-based and metaheuristic optimization).

PLACE PHOTO HERE

Saed Tarapiah received his M.Sc and Ph.D degrees in telecommunications engineering from Politecnico di Torino, Italy, in 2007 and 2011, respectively. He received his B.Sc degree in computer engineering from An-Najah National University, Palestine, in 2005. He is currently an Assistant Professor in the Department of communication Engineering, An-Najah National University, Palestine. In 2007 he was member of the AROMA project at Telecom Italia labs, TIM Turin, Italy. During 2009-2010 he was with MAESTRO Team, at INRIA Sophia Antipolis - Mditerrane Research Centre. His main area of interest is, Network Coding in Ad Hoc Networks, routing in Delay Tolerant Networks, and the encryption and security in Ad Hoc networks.