Evaluation of DTN Routing Algorithms in Scheduled Public Transport Networks

José Irigon de Irigon(✉), Felix Walter(✉), and Thomas Springer(✉)

Faculty of Computer Science, Computer Networks Group,
Technische Universität Dresden, Dresden, Germany
{jose.irigon,felix.walter,thomas.springer}@tu-dresden.de

Abstract. Public transport networks (PTN) can serve as a basis to establish low-cost communication solutions by using Delay- and Disruption-tolerant Networking (DTN) technologies. Since vehicles move according to a schedule, DTN protocols able to leverage topological information are expected to perform well in such a setup. Anyway, it has not been evaluated, if deterministic protocols perform best or if opportunistic protocols like PRoPHET or Spray and Wait can outperform them in some scenarios if appropriate parameter tuning is performed. In this paper, the performance of state-of-the-art DTN routing protocols, namely Epidemic Routing, Spray and Wait, PRoPHET, MaxProp, and CGR is compared with respect to their use in PTN. The performance comparison takes delivery probability, average latency, buffer utilization, and network overhead into account. The ONE was extended to simulate vehicle movement according to tracks and the schedule of the PTNs of Helsinki, Freiburg, and Prague. Our evaluation results demonstrate that protocol parameters should be selected carefully to achieve the best performance in different scenarios. The most efficient parameterization of protocols is described, and their influence on different performance metrics is discussed.

Keywords: DTN · Routing · Public transport systems · Performance evaluation

1 Introduction

Delay- and Disruption-tolerant Networking (DTN) enables *store-carry-forward* data transmission in *challenged networks*, which may face disrupted end-to-end paths or vast delays on individual links. For that purpose, application data are encapsulated in *bundles* and routed via the DTN overlay network. Applications for DTN explored in research are Interplanetary Internet, underwater networks, wireless sensor networks, and vehicular networks. Disruption-tolerant vehicular networks have been explored for various use cases to either enable connectivity in disconnected areas or to establish decentralized and cost-efficient communication alternatives in urban settings. In this work we focus on urban public transport

L. A. Grieco et al. (Eds.): ADHOC-NOW 2020, LNCS 12338, pp. 37–52, 2020.
https://doi.org/10.1007/978-3-030-61746-2_4

systems (PTS). PTS vary with respect to covered area, topology, number of lines, stop density, route length, trip frequency and number of used vehicles. As a result, the frequency of contacts and potential paths is highly dependent on the PTS topology and setup. Consequently, the characteristics of each PTS might influence which DTN protocol and parameter setup performs best. Even if deterministic DTN protocols can exploit PTS schedules to derive a contact graph and, thus, be expected to perform best, opportunistic approaches may outperform them due to parameterization, replication control and buffer management that might address specific PTS characteristics better.

In this paper we explore the performance of five DTN protocol versions in three urban public transport systems based on simulations with the ONE simulator [19]. The contribution of the paper is twofold. First, we examine parameter settings for each protocol to identify which protocol setup works best in each of the three urban PTS. Second, we compare the best performing setup of all five DTN protocols to identify the DTN protocol that works best in each PTS. For both series of experiments we investigate a set of performance metrics, namely delivery probability, latency and buffer utilization. The goal is to identify relations between these metrics and maximize one metric over others.

The paper is organized as follows: in Sect. 2 an overview about classes of DTN routing protocols and their major characteristics is given followed by a rationale for the selection of the particular protocols explored in this paper. As related work, DTN protocol performance evaluations are discussed in Sect. 3 with a focus on DTN over PTS. Section 4 introduces the approach for investigating the selected DTN protocols in three urban PTS. The evaluation results are discussed in Sect. 5. The paper concludes with a summary of our results and an outlook to future research in the field of DTN over PTS.

2 Background

Due to the inherent disruptions and delays in challenged networks, DTN routing algorithms are fundamentally different from those leveraged in IP networks. They may employ different levels of knowledge concerning the network topology. While some *opportunistic* algorithms perform reactive decisions based only on current connectivity, others estimate a utility metric from the history of contacts, which may also be exchanged transitively. *Deterministic* algorithms, such as Contact Graph Routing (CGR), leverage a schedule of future episodes of connectivity (*contacts*), which has to be provided in advance. In this paper, we compare the performance of five routing algorithms that make different assumptions of the application scenario and the provided information.

Epidemic Routing [27] is an opportunistic routing protocol often used as a baseline for comparison. The algorithm replicates messages to all reachable nodes, as long as they do not already carry a copy of the message. The latter is determined by the regular exchange of *summary vectors* containing unique message identifiers. Though the authors propose to enforce a maximum hop count, the number of replicas is not limited.

In contrast, *Spray and Wait* [26] limits the number of message replicas to a fixed value. Every time a message is forwarded, the algorithm decreases the number of copies in the buffer by the transmitted number of copies. The authors propose two techniques for calculating the latter; by setting it to half of the copies in the local buffer (*Binary* Spray and Wait) or always transmitting only a single copy (*Source* Spray and Wait).

The *Probabilistic Routing Protocol using History of Encounters and Transitivity* (PRoPHET) is specified in an IRTF Internet Draft [1]. Its forwarding decisions are based on the *delivery predictability* metric, which is derived for every known node from the history of encounters. During a contact, the value is adjusted due to the expectation of reaching the corresponding neighbor again. An aging factor is employed to decrease the metric over time if no further contacts are observed. Additionally, the delivery predictability values are disseminated to other nodes, allowing for transitive estimations. This way, bundle replication can focus on neighbors with a high probability to reach the destination. After a bundle has been delivered, acknowledgments are spread throughout the network, allowing nodes to clear their buffers of the bundle.

MaxProp [4] applies a similar transitive probability estimation technique as well as acknowledgments. Further, it extends the buffer management and queuing order by prioritizing messages based on their age and the number of traversed hops. This way, messages newly-introduced into the network are disseminated faster than older messages which were already forwarded via several hops.

The deterministic approach taken by *Contact Graph Routing* (CGR) [5] is vastly different: A *contact plan* has to be provided to the algorithm in advance, from which a *contact graph* is derived. This representation of the time-varying topology is used to calculate routes to every destination via a shortest-path algorithm. For every bundle, a viable route is selected and the bundle is proactively scheduled for transmission to the next node on the path. The implementation used in this work follows the standard recommended by the consultative committee for space data systems's (CCSDS), Schedule-aware bundle routing (SABR) [6].

The goal of our work was to find the most suitable routing algorithm for PTNs. By our selection, we expect to get an overview of the performance of approaches based on different levels of topological information: While Epidemic Routing as well as Spray and Wait rely solely on properties of the messages, PRoPHET and MaxProp leverage historical data and transitive information for their utility metrics, and sCGR calculates routes using a schedule provided in advance. Although the latter can exploit PTN schedules and, thus, is expected to perform well in scheduled networks, we perform parameter tuning of the opportunistic protocols with the goal to analyze whether they can compete with or even outperform the deterministic approach that needs an increased amount of initial information.

3 Related Work

Public Transport Systems have been proposed as disruption-tolerant data carrier in rural areas to provide educational means [16,25], economic development [13] and latency insensitive connectivity [17,22,25]; in urban areas they have been additionally proposed to offload traffic from mobile networks [14,20] and environmental monitoring [7,15]. Simulation is commonly used to estimate the performance of proposed routing algorithms in those scenarios. The faithful reproduction of the target mobility is essential to produce useful results. However, the accurate reproduction of complex scenarios is challenging, computationally intensive, and requires proper parameterization.

An extensive survey of tools and models for opportunistic networks is found in [9]. In the attempt to reproduce mobility faithfully, a set of available traces have been recurrently used to assess the performance of novel routing algorithms. However, traces are rarely available for each desired scenario; they are expensive to produce, inflexible (regarding to scaling and parameterization), and cannot be used in scenarios that do not yet exist. We could identify a few works in the DTN literature that, based on open data information, were able to create realistic PTN scenarios and compare their routing performance:

The *Routing in Urban Public Transport Systems* (RUTS) [10] reproduced the bus network of Braunschweig based on geographical information from OpenStreetMap [24] and exporting it to the *Simulation of Urban MObility* (SUMO) [21] to be combined with line definition and stop locations. The authors used a custom micro-mobility simulator to make a synthetic trace later imported to the ONE simulator. Based on the information extracted from timetables, a routing algorithm was proposed that forwards messages according to the encounter likelihood. In RUTS, differently from the current approach, buses are not assigned to a specific route and may serve a different line at every shift. As we notice in the evaluation, this difference seems to cause a considerable difference in routing algorithms based in contact history.

The *Urban Routing Backbone Simulator* (URBeS) [12] proposed the use of an open data provider (Google Transit Feeds - GTFS) in the creation of realistic scenarios. However, the authors do not make it clear whether the real trajectory between bus stations is considered since this information is optional and missing in most GTFS feeds. URBeS proposes a probabilistic routing algorithm based on the number of encounters to exchange messages from buses to a destination bus line (instead of a specific bus). In its evaluation, a period of 10 h is considered (8 a.m. to 6 p.m.); Epidemic Routing achieves the highest delivery ratio, but it was considered unsuitable due to the number of replicas per message.

Zimmermann *et. al* provide behavioral analysis of the bus network in Aachen through a simulated scenario created with the support of the transportation provider and technical operator. [28] The simulation considers the time-span of six hours from 7:30 to 13:30 on Mondays; besides, they assume that every bus and bus stop (station) are equipped with an 802.11 module, allowing bus-to-bus as well as bus-to-station communication. The paper highlights the impact of infrastructure elements to the number of contacts. Unfortunately, only Epidemic Routing was considered, ignoring the influence of those elements in the determinism of future contacts and the ability of different routing algorithms to explore this information. Different use-cases that use static communication modules to support communication over vehicles with opportunistic mobility have been mentioned in [2], but we could not find a performance assessment of DTN routing algorithms in a PTN network that allow vehicle-to-vehicle and vehicle-to-station communication. Finally, [23] highlights that the performance of a routing algorithm is highly dependent on its parameterization. Unfortunately, novel routing algorithms are often compared to inappropriate algorithms using only standard parameterization.

4 Methodology and Evaluation Setup

We use a simulation approach based on a mobility model that reproduces the vehicle motions in PTN. To get a performance overview in distinct networks, we selected PTNs that vary in terms of the number of traffic lines and stations as well as the covered area. As a first step, the parameters of *PRoPHET* and *Spray and Wait* were tuned to identify those versions that perform best in the three PTNs. In addition, we explored if a single parameter setup performs best in all three PTNs or if parameters should be tuned for every scenario. As a second step, we compared the performance of the best performing versions of the five selected routing protocols to explore which of them performs best in the selected PTNs and if there is a single protocol that performs best in all PTNs or if the results differ in the three PTNs. In the following, we describe the selected PTN, preparation of scenario data, and the simulation setup.

For the evaluation, we selected three PTSs with different characteristics, which are illustrated in Fig. 2. The metro network of Prague, with three traffic lines serving 61 stations, is the smallest with respect to traffic lines and stations but covers the largest area of 230 km^2. The tram network of Freiburg and Helsinki cover smaller areas (63 km^2 in Freiburg and 43 km^2 in Helsinki) but contain a larger number of traffic lines and stations. Table 1 illustrates the parameters of the three PTS scenarios.

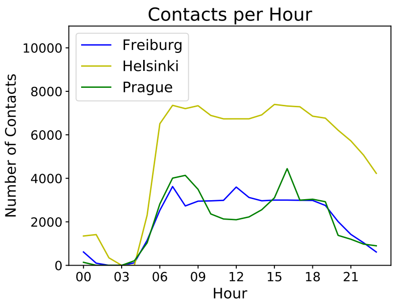
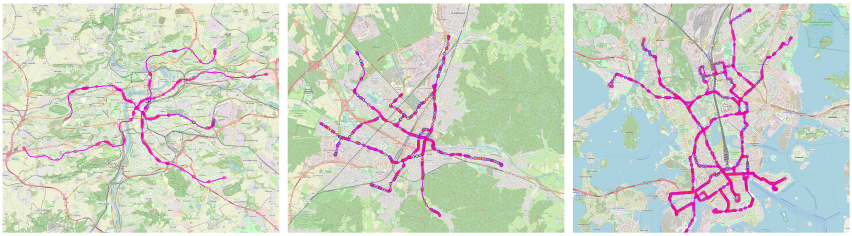


Fig. 1. Contacts over time

Table 1. Evaluated scenarios

	Prague Metro	Freiburg Tram	Helsinki Tram
GTFS Version	23.03.20	01.08.19	21.03.20
Area (km^2)	230	63	43
# Stations	61	89	213
# Traffic Lines	3	5	9
# Sim. Vehicles	106	117	116
# Trips in a day	1550	1458	2262



(a) Metro Prague (b) Tram Freiburg (c) Tram Helsinki

Fig. 2. Scenarios

Our scenario data is based on the General Transit Feed Specification (GTFS) and extended with geographic information from Open Street Maps with support of the map matching tool Pfaedle [3]. Route descriptions with station coordinates and vehicle schedules for each transit line were extracted and translated to a format suitable to feed the mobility model of the simulation tool. We used the ONE extended by a mobility model that reproduces real vehicle movement in PTN to generate scenario data for the simulations.

A trip represents the movement of a vehicle along a route of a particular traffic line according to a given timetable. It is defined by its departure time, departure station and arrival station. Usually, two stations are repeatably used as departure and arrival stations for a traffic line. However, routes vary in real PTNs. In [12], Gaito et. al. proposes the exclusion of all routes that are not *closed*, i.e., trips served by vehicles departing from or arriving at stations other than the first or the last of a route taken as a model. Based on the selected scenarios, we verified that the exclusion of those routes causes a considerable change in the network capacity. The metro line A of Prague, for example, defines 140 trips (70 in each direction) in a day whose route ends one station earlier. In this work, we propose the concept of *reference route* as a list of stations to be used as a model. Trips are defined as a pair of indices to the reference route and departure time. Only routes whose head or tail is not contained in the reference route are ignored. The number of routes represented depends on the proper choice of the reference route. Considering any route from the timetable as a candidate, we defined the reference route as the candidate capable of representing the greatest number of trips. Our simulation was able to represent over 95% of all trips defined in the schedule by using reference routes.

As the simulation starts, the mobility model instantiates a scheduler that is responsible for assigning trips to vehicles according to the timetable. Every vehicle queries the scheduler for its first trip, setting its location to the departure station. From this moment on, a vehicle must be located at the departure station at the departure time in order to be assigned to a trip (there are no jumps between stations or vehicular relocation). The scheduler works as follows: at startup, it reads from the disk all trips based on the timetable. Then, it chooses a vehicle V that has no trips assigned and adds it to the list of trips to serve, calculating its arrival time (AT) and arrival station (AS). Next, it selects the first trip departing from AS after AT, repeating this process for all trips V is able to serve. Finally, the next idle vehicle is chosen, and the process starts over until all trips are served.

At depart time, a vehicle calculates the speed using the expected time and distance to the next station as defined in the trip received by the scheduler and starts moving at a constant speed. At each station, it waits a configurable amount of time. In our experiments, a random waiting time between 10 and 15 s is used. As a vehicle arrives at the end of a trip, a vehicle queries the scheduler for the next available trip. If no further trips are defined, or in case the vehicle has to wait longer than 20 min, it turns off its radio.

In this work, both vehicle-to-vehicle and vehicle-to-station communication is considered. However, only stations are chosen as source and destination of bundles. Bundles are created at a constant pace; when a bundle is created, two stations are randomly chosen as source and destination. We assume that every vehicle and station is equipped with an 802.11p module. The amount of time a vehicle stops at a station is a configuration parameter. Our evaluation considers a stopping time of 30 s and a communication range of 80 m line of sight. In our evaluation we varied the message size and transmission capability, since they are application dependent and vary greatly depending on the use-case. Our evaluation considered two message sizes (1 MB and 5 MB) and transmission speed of 8 Mbps and 40 Mbps.

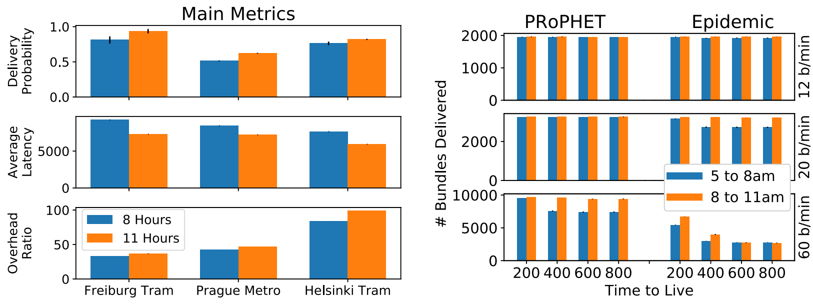
Due to the lack of information on vehicle reallocation, some lines end up with a disproportionate number of vehicles. On the one hand, the creation messages for vehicles that are out of service masks the evaluation of the routing protocol. A similar effect occurs in choosing vehicles serving their last trip as the bundle destination. On the other hand, ignoring the message creation when the chosen vehicle is out of service changes the frequency with which messages are created across simulation runs.

5 Evaluation

For the evaluation of different routing algorithms and metrics, we defined three classes of network load: a *low load* class with bundle size of 1 MB, 12 bundles created per minute and transmission speed of 40 Mbps; an *average load* class with bundle size of 1 MB, 20 bundles created per minute and transmission speed is 8 Mbps; finally, a *high load* class with 5 MB bundles created at every second and transmission speed of 8 Mbps.

5.1 Defining the Simulation Period

To deal with the known limitations of the ONE simulator's scalability, we first explore in two experiments whether an eight-hour period is capable of producing usable results.



(a) Varying scenarios (b) Varying load and TTL

Fig. 3. Setting simulation period (Color figure online)

In the first experiment, we measured the performance of PRoPHET Routing in different scenarios for the early eight and eleven hours of a day to verify whether the different setups produce comparable results. 10 runs for every city were performed varying β from 0.1 to 1.0 in steps of 0.1. Figure 3a contains the measured values for delivery probability, average latency, and overhead ratio. The increase in simulation time shows a consistent rise in delivery probability and overhead ratio, and a decrease in average latency. In the second experiment, we compared the number of delivered bundles for PRoPHET and Epidemic in the same periods simulating the PTS of Freiburg, while varying bundle time to live (TTL) and network load. The goal of this experiment was to verify the influence of the warm up time on probabilistic routing performance. The results of three runs are presented in Fig. 3b. On top, a low load variation that creates 12 bundles every minute is shown; in the middle the average load variation is presented, that creates 20 bundles per minute; at the bottom, in the high load variation, a bundle is created every second. We ran every routing algorithm with four different TTLs varying from 200 to 800 min, as specified in the x axis. The bars represent the absolute number of messages delivered in two consecutive periods separately: from 5 a.m. to 8 a.m. (blue) and from 8 a.m. to 11 a.m. (orange).

Under *low load* (left), there is a minimal difference in the number of delivered bundles between the two periods observed for TTL equal to or higher than 400. This difference increases considerably for Epidemic under *average load*. During the first hours of the day, bundles are generated at a constant pace but cannot be exchanged due to the lack of vehicular mobility. During the first trips, vehicles collect and distribute bundles stored along the route. Epidemic Routing is the most affected since PRoPHET can better utilize the contact time by avoiding the waste of contact time with bundles with low delivery probability. A further increase in the transmission load eventually reaches the limits of PRoPHET routing that starts to present a similar tendency, even though being able to cope with congestion better than Epidemic. The second experiment shows that

Epidemic had an increasing number of delivered bundles in the second period, similarly to PRoPHET.

This result excludes the assumption that the improvement over time noticed in the first experiment was caused (at least exclusively) by the learning process. The tendencies observed in the main performance metrics were consistent, and the congestion effects were noticeable in both periods. Therefore, we conclude that the outcome of simulation realized in those scenarios during the first eight hours of a week-day are able to produce meaningful material to the performance assessment of different DTN routing algorithms. Consequently, in the following experiments the first eight hours of a week-day are simulated.

5.2 Spray and Wait: *Binary* vs. *Source*

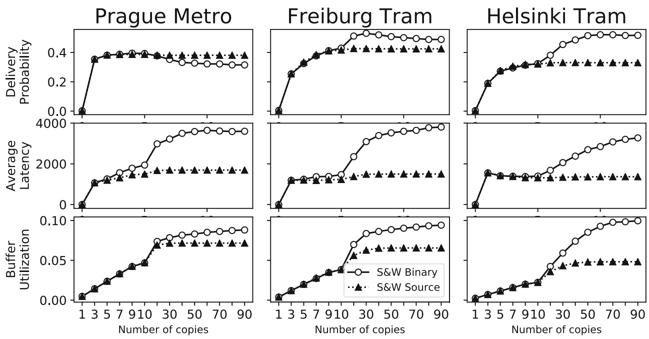


Fig. 4. Spray and wait average load: 1 MB size, 20 bundles/min, 8 Mbps

Spray and Wait Routing assumes that the highest possible delivery likelihood is achieved by Epidemic, at the price of energy and latency. The goal of this section is threefold: first, verify whether these assumptions hold for public transportation systems; second, assess the importance of the right parameterization as well as the factors that affect its performance; third, compare both spray variations (*Binary* vs. *Source*) to verify whether one of them is more suitable for public transport systems. Spray and Wait variants have a single parameter: the number of replicas. In our experiments, we vary this number from one to nine in steps of two and from 10 to 90 in steps of 10 s. We analyzed the results for delivery probability, average latency, and buffer utilization for the three scenarios. Figure 4 illustrates the behavior for a network configuration under *average load*. Average latency represents the period between message creation and delivery. Figure 4 depicts the average latency based on bundles created after 5 a.m. Buffer utilization is presented in percentage of the total storage capacity. Our experiments utilized 1 TB storage, excluding congestion due to lack of storage. However, we consider storage occupancy an important metric, since proposed use cases frequently consider limited storage capabilities.

In *average load* and *lower load* scenarios, *Binary* spray performed better than *Source* with respect to delivery probability (DP). However, the observed gain in DP in the city of Freiburg and Helsinki comes at the cost of latency and buffer utilization (caused by an increasing number of transmissions). For example, in the city of Freiburg, a simulation with 30 copies achieved the highest DP (53%), 23.5% better than the result obtained using ten copies. However, this performance improvement also increases the average latency in 110% and buffer utilization in 178.5%. In the *high load* scenario, an increase in the number of bundles also decreases DP due to congestion. The use of 30 copies in our experiments caused 45% DP lost in Prague, 50% in Freiburg, and 56% in Helsinki with the best number of copies three (Prague), three (Freiburg), and five (Helsinki). In general, *Source* Spray and Wait is not the right candidate for complex PTS. It only forwards one bundle at a time to be delivered directly to the destination. Since, in our scenarios, bundles are created exclusively by stations, they are only delivered if the destination is reachable in two hops. However, in *high load* scenarios, *Source* Spray performed similarly to *Binary* spray, sometimes slightly better. Based on these results, only *Binary* Spray and Wait is used in the assessment of different DTN routing protocols in Subsect. 5.4

5.3 PRoPHET

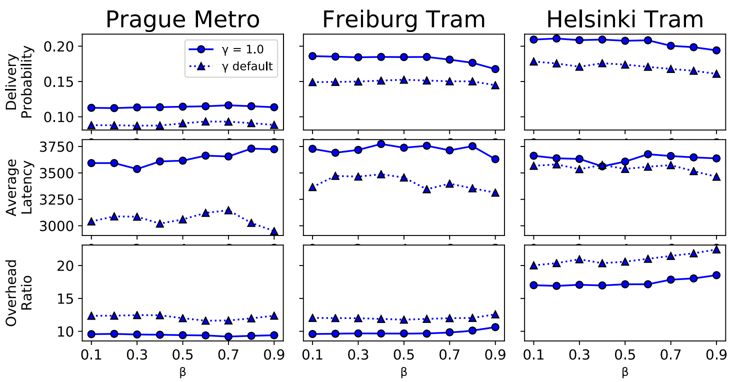


Fig. 5. Tuning PRoPHET

PRoPHET is a probabilistic routing algorithm that maintains a table with the predictable likelihood of future encounters to each neighbor. It has two parameters: γ defines how fast the expected probability decreases over time (aging); β defines a reducing factor to the expected probability at each extra hop (transitivity). In this section, we analyze the importance of the right tuning for PRoPHET, i.e., how aging and transitivity affect the performance of different configurations and how the design of the simulation framework may affect the outcome of the routing algorithm. We performed an initial set of runs varying γ and verified that, as long as the schedule is respected, the best outcome is achieved, ignoring the aging factor ($\gamma = 1.0$). This behavior is understandable since vehicles visit the same set of stations with high likelihood. Moreover, when aging is taken into account, the lack of communication during the night affects the first morning trips negatively, since vehicles suppose that future communications are rare, based on the nocturnal behavior. For each scenario and parameterization set (low, average, and high load), two sets of 10 experiments were performed: in one set, γ was set to its default value (0.999885791), and in the other set γ was set to 1.0. In each set, we varied β from 0.1 to 1.0 in steps of 0.1. The result for the *high load* run is illustrated in Fig. 5. Setting the γ to 1 increased the delivery probability in every scenario for all βs and caused a decrease in the overhead ratio, i.e., the number of copies per message delivered. Although an increase in average latency seem to be associated with an increase in DP, experiments show that *average load* configurations still achieve better DP (70% for γ of 1 against 64% for default value, while a similar average latency was achieved. In Helsinki, tuning the right β resulted in an increase of 12% in DP, showing the importance of the proper routing parameterization in PTS scenarios, as Oliveira et al. have shown for opportunistic networks in [23].

Finally, the chosen routing algorithms takes into account vehicles (tram, bus) instead of lines. We believe that a small modification in the routing algorithm that decide based on the line identifier (instead of device identifier) and to learn about the line statistics could overcome the effects of the lack of knowledge about the device behavior between trips. Consequently, we identified nine individual setting for β and γ for each of the three scenarios and network load options that produced the best performance results. These settings will be used in the comparative evaluation of DTN protocols.

5.4 Performance Comparison Between Different Routing Algorithms

This section compares the performance of the routing algorithms Epidemic, Binary Spray and Wait, PRoPHET, MaxProp, and CGR. An extensive tuning for MaxProp, as we have done for PRoPHET and Spray and Wait, was not performed because the simulation of MaxProp is computing-intensive and time-consuming. Instead, we used the default configuration for MaxProp for all simulation runs. A second challenge was to simulate CGR in the ONE. The ONE has known scalability limitations [9], and our implementation was not able to simulate the tram network of Helsinki. Our approach to circumvent this problem was to export the trace generated by the ONE as input for the *aiodtnsim*, an asyncio-based DTN simulator [11] to simulate CGR in each scenario. *aiodtnsim* was validated against the ONE and produced comparable results for Epidemic and Spray and Wait. Figure 6 summarizes the results of each routing algorithm under different loads. Under *low load*, all routing algorithms with exception to Spray and Wait presented high delivery probability and similar outcome (all results were within 0.8% of their mean in each scenario). Spray and Wait did not perform better because the maximum number of copies was limited to 90.

Increasing this number should allow Spray and Wait to have a DP similar to Epidemic.

Under *average load*, CGR presented the highest DP in each scenario, followed by PRoPHET; additionally, CGR achieved the least average latency in Freiburg and Helsinki. An attentive reader might find curious the fact that the buffer utilization of Spray and Wait changes drastically. Notice that the illustrated bar corresponds to the best parameterization (number of copies) per scenario: 9 copies in Prague, 30 in Freiburg, and 90 in Helsinki. The lowest DP was achieved by Epidemic routing in the three scenarios. Although MaxProp does not stand out for a high DP, its drop policy's effect is perceptible in the resulting buffer utilization, especially in Freiburg and Helsinki.

In *high load* scenarios, DP values are much lower for all protocols than in low and *average load* scenarios. In Prague and Freiburg the highest DPs were achieved with *Binary Spray and Wait*: 16.72% and 19.15% respectively; the downside is that in Prague, it also had the second-highest average latency. The second highest DP had CGR with 12.06% and 18.37%. In Helsinki, CGR achieved the highest DP (31.41%) and the least latency, followed by PRoPHET and Spray and Wait with 21% DP. MaxProp and Epidemic achieved the lowest DP.

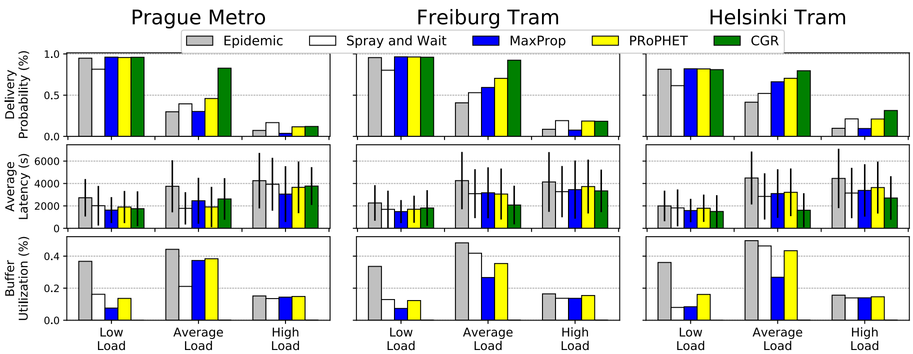


Fig. 6. Routing algorithm comparison

Epidemic Routing, despite its simplicity, was able to achieve high DP in the three scenarios under *low load*, but DP falls rapidly as soon as load increases. Besides, it presented the highest latency and buffer utilization through all experiments. Spray and Wait achieved an overall low latency, but also low DP for *low* and *average load*; in *high load* scenarios it stand out, given that it is configure with the right number of copies. The maximum DP under low load was achieved with the maximum number of copies (90 in our experiments); under *average load*, the highest DP was achieved with 9, 30, and 60 copies in Prague, Freiburg, and Helsinki respectively; finally, under *high load* the number of copies fall to 3, 3, and 5. Despite using its default configuration, MaxProp achieved the highest DP in all cities under *low load* (Prague: 96.11%, Freiburg: 96.5% and Helsinki:

82.01%), the lowest latency in Prague and Freiburg, and the second least buffer requirements, due to its ability to delete acknowledged messages and buffer management. PRoPHET was among the three highest DPs in each scenario and stood out in scenarios for the *average load* without the requirement of a precise contact plan, as in CGR. It stands in the average regarding average latency and buffer utilization. Finally, CGR achieved the highest DP in general, especially under *average load*: 82.74%, 92.40%, and 0.7963% compared to 45.93%, 70.41%, and 70.46% achieved by the second-highest DP (PRoPHET) in the cities of Prague, Freiburg and Helsinki, respectively. Additionally, the average latency of CGR was the lowest in Freiburg and Helsinki for average, and *high load* and its buffer utilization was the least in each experiment since it is a single copy approach.

Considering another perspective, it is possible to classify the routing algorithms according to the amount of information they exchange to support the routing algorithm. The most straightforward algorithm is Epidemic, which does not need to change any information. It is a good choice if resources are abundant and the load remains low; however, regardless of the load, Epidemic presents the highest average latency and buffer utilization in all scenarios. Besides, PD of the Epidemic declines rapidly as the load increases, due to congestion. Therefore, our experiments lead to the conclusion that the Epidemic should be avoided whenever resources are limited, or there is no guarantee that the load will remain low. Spray and Wait control the number of replicas to minimize congestion; our experiments show that this approach can effectively increase the DP and decrease latency, especially under *high load.* However, defining the number of replicas is challenging: it is statically configured in Spray and Wait, but the right number depends on the communication and mobility behavior. We tested all variations in our experiments and chose the best outcome, but this cannot be done in reality. An alternative is to exchange information at runtime to adapt the number of replicas, as proposed in [8]. MaxProp's buffer management and acknowledgments proved to be effective in reducing buffer utilization. Additionally, it takes into account the delivery likelihood based on transitive information about contact opportunities exchanged at runtime. Its DP was, in average, lower than PRoPHET, but we must consider that it used its default configuration. PRoPHET has achieved in average the second highest DP routing exclusively based on information about contact opportunities, which seems to be an effective routing strategy in PTS networks. This assumption is reinforced by the result of CGR that reached the highest PD and the lowest storage usage based on precise information about contact opportunities. These results lead us to believe that the theoretical results presented by Jain et al. [18] are valid to a scheduled scenario as PTS, reaffirming the premise that algorithms capable of exploiting the environmental information accurately are expected to achieve higher performance. A detailed study to consider the effects of delays and unexpected route modifications caused by accidents or disasters in the outcome of those routing algorithms is proposed as future work.

6 Conclusion and Future Work

In this paper, the performance of five DTN routing protocols was explored in three urban public transport networks, namely the metro network in Prague, and the tram networks in Freiburg and Helsinki. We incorporated route information and time tables from real public transport networks in the ONE simulator. In the first simulation step, we identified the time period of the first 8 h of a weekday as sufficient for our simulations. Next, a comparison between Source and Binary Spray and Wait shows that the latter variant is more appropriate for this setup. We highlighted the importance of proper parametrization and its trade-offs, based on an extensive set of runs with Spray and Wait and PRoPHET Routing. Finally, we compared the performance of five selected DTN protocols. Simulations revealed that the protocol performance was highly dependent on the network load. Epidemic Routing obtained poor performance metrics under average and high loads. Binary Spray and Wait was able to improve DP, reduce average latency and buffer utilization given that it used the right parameterization. MaxProp's approach to exchange acknowledgements and buffer management based on transitive probabilities exchange was able to achieve the second best performance in buffer utilization. PRoPHET achieved the second highest DP in average, behind CGR that, in general achieved the best overall performance. Since the right parametrization depends on the network load, our results lead to the conclusion that information about the network behavior at runtime should be taken into account. The communication behavior (creation of messages) in this work remained constant, and mobility respected the schedule. The assessment of the above mentioned protocols under variable load and unexpected mobility are planned as future work.

Acknowledgement. This work is partially funded by the German Research Foundation (DFG) within the Research Training Group Role-based Software Infrastructures for continuous-context-sensitive Systems (GRK 1907).

References

1. Lindgren, A., Doria, E.D.S.G.: Probabilistic Routing Protocol for Intermittently Connected Networks. Internet Engineering Task Force, October 2011. https://tools.ietf.org/html/draft-irtf-dtnrg-prophet-09. Experimental
2. Baron, B., Spathis, P., De Amorim, M.D., Viniotis, Y., Ammar, M.H.: Mobility as an alternative communication channel: a survey. IEEE Commun. Surv. Tutor. **21**(1), 289–314 (2019)
3. Bast, H., Brosi, P.: Sparse map-matching in public transit networks with turn restrictions. In: GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, pp. 480–483 (2018)
4. Burgess, J., Gallagher, B., Jensen, D., Levine, B.N.: MaxProp: routing for vehicle-based disruption-tolerant networks. In: Proceedings of the IEEE INFOCOM, May 2006
5. Burleigh, S., et al.: Contact graph routing in DTN space networks: overview, enhancements and performance. IEEE Commun. Mag. **53**(3), 38–46 (2015)

6. CCSDS members: schedule-aware bundle routing (2019). https://public.ccsds.org/Pubs/734x3b1.pdf
7. Christodoulou, S., Gizas, A., Asimakopoulos, G.: Dynacargo routing subsystem and its algorithms for efficient urban waste collection. In: Proceedings of the 11th International Conference on Practice and Theory of Automated Timetabling (PATAT-2016), Udine, Italy, pp. 45–61 (2016)
8. Cui, J., Cao, S., Chang, Y., Wu, L., Liu, D., Yang, Y.: An adaptive spray and wait routing algorithm based on quality of node in delay tolerant network. IEEE Access **7**, 35274–35286 (2019)
9. Dede, J., et al.: Simulating opportunistic networks: survey and future directions. IEEE Commun. Surv. Tutor. **20**(2), 1547–1573 (2018)
10. Doering, M., Pögel, T., Wolf, L.: DTN routing in urban public transport systems. In: Proceedings of the 5th ACM Workshop on Challenged Networks - CHANTS 2010, p. 55. ACM Press, New York (2010)
11. Walter, F.: A minimal framework for performing DTN simulations based on Python 3.7 and asyncio (2020). https://gitlab.com/d3tn/aiodtnsim
12. Gaito, S., Maggiorini, D., Rossi, G.P.: Leveraging Bus Mobility to Enable Communications in Urban Areas. Wired, p. 14 (2011)
13. Galati, A., Sazonovs, A., Olivares, M., Mangold, S., Gross, T.R.: Delay tolerant networking for the socio-economic development in rural South Africa. In: Gaggi, O., Manzoni, P., Palazzi, C., Bujari, A., Marquez-Barja, J.M. (eds.) GOODTECHS 2016. LNICST, vol. 195, pp. 195–202. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61949-1_21
14. Gao, Z., Meng, J., Wang, Q., Yang, Y.: Data offloading for deadline-varying tasks in mobile edge computing. In: Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications. Cloud and Big Data Computing, Internet of People and Smart City Innovations, SmartWorld/UIC/ATC/ScalCom/CBDCo, pp. 1479–1484 (2018)
15. Giannini, C., Calegari, P., Buratti, C., Verdone, R.: Delay tolerant network for smart city: exploiting bus mobility. In: AEIT 2016 - International Annual Conference: Sustainable Development in the Mediterranean Area, Energy and ICT Networks of the Future, pp. 1–6 (2016)
16. Grasic, S., Lindgren, A.: Revisiting a remote village scenario and its DTN routing objective. Comput. Commun. **48**, 133–140 (2014)
17. Guo, S., et al.: Very low-cost internet access using KioskNet. Comput. Commun. Rev. **37**(5), 95–100 (2007)
18. Jain, S., Fall, K., Patra, R.: Routing in a delay tolerant network. ACM SIGCOMM Comput. Commun. Rev. **34**(4), 145 (2005)
19. Keränen, A., Ott, J., Kärkkäinen, T.: The ONE simulator for DTN protocol evaluation. In: Proceedings of the Second International ICST Conference on Simulation Tools and Techniques (2009)
20. Komnios, I., Tsapeli, F., Gorinsky, S.: Cost-effective multi-mode offloading with peer-assisted communications. Ad Hoc Netw. **25**(PB), 370–382 (2015)
21. Lopez, P.A., et al.: Microscopic traffic simulation using sumo. In: The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE (2018)
22. Naidu, S., Chintada, S., Sen, M., Raghavan, S.: Challenges in deploying a delay tolerant network. In: Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM, pp. 65–71 (2008)

23. de Oliveira, E.C.R., et al.: Context-aware routing in delay and disruption tolerant networks. Int. J. Wireless Inf. Netw. **23**(3), 231–245 (2016). https://doi.org/10.1007/s10776-016-0315-2
24. OpenStreetMap contributors: planet dump (2017). https://planet.osm.org, https://www.openstreetmap.org
25. Pentland, A., Fletcher, R., Hasson, A.: DakNet: rethinking connectivity in developing nations. Computer **37**(1), 78–83 (2004)
26. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and wait : an efficient routing scheme for intermittently connected mobile networks. In: SIGCOMM, pp. 252–259 (2005)
27. Vahdat, A., Becker, D.: Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical Report CS-200006. Department of Computer Science, Duke University (2000). http://issg.cs.duke.edu/epidemic/epidemic.pdf
28. Zimmermann, T., Wirtz, H., Punal, O., Wehrle, K.: Analyzing metropolitan-area networking within public transportation systems for smart city applications. In: 2014 6th International Conference on New Technologies, Mobility and Security - Proceedings of NTMS 2014 Conference and Workshops, pp. 1–5 (2014)