

DTN Routing in Urban Public Transport Systems

Michael Doering, Tobias Pögel, Lars Wolf
Institute of Operating Systems and Computer Networks
Technische Universität Braunschweig
Braunschweig, Germany
[mdoering|poegel|wolf]@ibr.cs.tu-bs.de

ABSTRACT

Communication is crucial to the coordination and efficient operation of public transport systems. However, deployment of infrastructure based communication systems is very expensive. Delay tolerant vehicular networks are a promising alternative since only very few infrastructure elements are required. This paper presents a DTN routing algorithm for urban public transport systems. Beginning with an analysis of node mobility, system characteristics are derived and exploited to improve routing performance. To increase realism in the performance evaluation and comparison a new approach is taken for the generation of mobility traces. A map based on real cartographic data is combined with line definitions, stops and timetables of real public transport systems. A micromobility simulator then produces large scale mobility traces which are fed into a DTN simulator. We compare various DTN routing schemes with our algorithm. Moreover, the impact of disturbances in the public transport system on the routing performance is examined. The results show that our routing algorithm can outperform previously proposed algorithms even if 20% of all vehicles are behind schedule.

Categories and Subject Descriptors

C.2.2 [Computer Systems Organization]: Computer-Communication Networks—*Network Protocols*

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

DTN, Routing, Simulation, Public Transport, Mobility, RUTS, The ONE, V2V, V2I

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHANTS'10, September 24, 2010, Chicago, Illinois, USA.
Copyright 2010 ACM 978-1-4503-0139-8/10/09 ...\$10.00.

1. INTRODUCTION

Communication systems have a big impact on the efficiency and user-friendliness of public transport. Vehicle dispatching, traffic management, arrival prediction, dynamic passenger information and dynamic rerouting are just a few examples for processes that require communication. The increasing integration of information technology into the management and operations of public transport leads to a growing demand of wireless data transfer. Cellular data services provide sufficient range but relatively low data rates. Furthermore, the deployment and operation of a dedicated cellular infrastructure require relatively high investments. For the operator of public transport, permanent cellular access costs might be inhibiting. An alternative is direct vehicle-to-vehicle communication based on IEEE 802.11a/b/g/n/p. These WLAN technologies are capable of significantly higher data rates but also have a lower range, so it is unlikely that each vehicle is constantly in the radio range of other vehicles. This problem could be solved by roadside infrastructure, which would again require high investments. A delay tolerant network (DTN) is a more economical approach since for many applications a (slightly) delayed delivery of data can be accepted.

In a DTN end-to-end connections are not required. Instead the protocol data units, referred to as bundles, may be stored until a connection to the next hop is available. Then the bundle is forwarded and stored again, and so on, until it finally reaches its destination. Choosing the 'right' sequence of next hops to minimize delay and maximize reliability is the routing algorithm's task. In this paper we propose a DTN routing algorithm that takes advantage of the characteristics of public transport systems. First we analyze mobility and the characteristics of contacts in public transport systems. Based on this the design of a specialized routing algorithm and the implementation are described. For the evaluation we present a new approach centered on a chain of simulation tools generating very realistic mobility traces. Finally we compare various routing algorithms using these traces.

1.1 Related Work

Several generic DTN routing protocols have been proposed that can be applied to urban public transport networks. FirstContact routing [3] follows a hot-potato approach and forwards a message to the next available node until the destination is reached. A random node is selected if more than one node is in radio range. In the most implementations the nodes relay their bundles only once per contact

with another node. This reduces permanent loops and a high network load. Typically FirstContact requires only a minimal message buffer on the nodes but causes relatively high delays. Epidemic routing [8] is a simple but effective routing method based on flooding. Messages are forwarded to all neighbors if the receiver is not directly reachable by the sender. Each neighbor maintains a buffer for flooded messages. Upon meeting another node a summary vector is exchanged to prevent unnecessary transfers of messages that both nodes have already buffered. Messages are discarded in a first-in-first-out manner if the buffer's capacity runs low. Message lifetime is limited by a hop counter. Epidemic Routing achieves a low latency but causes a high network load.

Spray and Wait [7] tries to reduce the overhead of message copies by prohibiting non-source nodes to create message copies. Only the source node creates a limited number of L copies. The routing is divided into two phases: In the spray-phase the nodes spray and forward messages and in the wait phase they wait for a direct transmission to the destination node. Whenever a node discovers another node which has not yet buffered the message, the source node starts with the spreading and forwards one or $L/2$ message copies to the new neighbor. The spreading is stopped as soon as nodes keep only one copy. Afterwards both nodes change to the wait phase and stand by for a direct delivery.

Spray and Wait routing spreads messages quickly to the first nodes contacted. This can be problematic if exactly those nodes cover only a limited area. In this case far destinations may not be reached. Therefore a follow-up approach has been proposed: Spray and Focus [6] generates L tokens for every new message. These determine the number of possible copies and forwardings. Similar to Spray and Wait routing, the nodes forward $L/2$ tokens when the message is known on only one side. When all tokens have been used, the node goes into the focus phase. In contrast to the Spray and Wait routing, messages can still be relayed. It calculates a utility function which decides about the transfer. For example this utility function could be derived from the expected duration of a contact.

PROPHET [5] is a probabilistic routing protocol using a delivery prediction based on historic encounters. The forwarding strategy is similar to epidemic routing, but the resource utilization is reduced by forwarding only a few message duplicates to nodes with a high delivery predictability. Like in epidemic routing the nodes exchange summary vectors on each encounter. These vectors contain the identifiers of messages in the nodes' buffers as well as updated predictability information. PROPHET can be applied to routing in public transport systems but does not exploit their specific characteristics.

MaxProp Routing [1] is optimized for DTNs with fast moving nodes. It is based on flooding but prioritizes messages with the lowest costs caused by forwarding the message to its destination. The costs reflect the delivery probability. In addition, each node maintains vectors with an estimation of the probability of meeting every other node. At each contact, the vectors are updated and re-normalized. Moreover, the nodes exchange their vectors. Then the potential paths to the destination node is calculated. The costs result from the encounter probabilities. Like PROPHET, MaxProp can be applied to public transportation systems. However, both routing algorithms rely on historic encounters and do not

proactively exploit a public transport system's properties such as timetables and lines.

2. DTN-ROUTING IN URBAN TRANSPORT SYSTEMS

In this section we present the design of a DTN routing algorithm for public transport systems. Our approach aims at a specialized algorithm that actively exploits the characteristics of public transport systems to achieve a better efficiency. The drawback of this approach is that the specialized routing cannot be applied to such a wide range of applications as the more generic approaches like e.g. epidemic routing. On the other hand, DTNs are typically deployed in very specific scenarios, so a potential performance gain is often more important than generic routing.

2.1 Analysis and Characteristics of Urban Public Transport Systems

In this paper we focus on urban public transport systems with vehicles operating on surface infrastructure. Buses and trolleys running on roads or rails are the most common vehicles in such transport systems. These vehicles move on lines along stops. The arrival and departure time at the stops is defined by a timetable, which also defines a frequency of service on a given route. Usually the frequency of service changes several times a day, according to a static plan based on the average number of passengers at a given time. For example frequency of service may be five minutes during rush-hour and 30 minutes at late evening. It should be noted that timetables are hard to maintain since unpredicted disturbances may happen such as traffic congestion, vehicle failure, traffic accidents or blocked roads. Another important property is that there is no fixed assignment of vehicles to routes. Usually each vehicle is dispatched to a random route at the beginning of a shift. Moreover, during one shift vehicles frequently change routes in order to increase usage rate by reducing deadheads and waiting times.

A possible contact occurs whenever two vehicles are in each others radio range. Vehicles move on lines and follow a timetable, therefore potential contacts are roughly predictable. However, there are always minor variations, e.g. caused by traffic lights or changing numbers of passengers getting on and off at stops. In certain situations these minor variations make the contact prediction very difficult. For example a variation of just a few seconds can prevent a potential contact between vehicles of two different lines crossing at an intersection. For this reason it is important to assess the probability and the duration of a possible contact. Since all vehicles move on predetermined lines a prediction based on the length of joint segments can be made. However, vehicles can also move in different directions on the same line. A contact is most likely if both vehicles enter a joint segment simultaneously on both ends and move in opposite directions. But these contacts can also be expected to have a shorter duration than contacts of vehicles moving in the same direction, since vehicle speeds in joined segments are very similar. Therefore, vehicles driving in the same direction with a low distance will maintain longer contacts.

2.2 General Approach

The aims in the design of Routing in Urban Public Transport Systems (RUTS) are resource-friendliness as well as timely

and reliable delivery as far as possible in a public transport vehicular DTN. In addition, the modular design allows for changes and extensions in the future.

Urban public transport has some specific characteristics, e.g. timetables and network maps that can be exploited for routing. If this context information is available on each DTN node an efficient routing is possible. First the possible contacts between the vehicles are determined and stored in a routing-graph. Based on this routing-graph different ways from the sender to the receiver can be identified. Each way represents a possible routing path for a bundle. Due to external disturbances and local conditions, contacts have different performances and probabilities. Therefore, every involved contact in the routing paths has to be evaluated. This result is a rating for every possible routing path. Now, the bundle can be passed on the most suitable path. Multiple bundle copies can be used to increase the delivery probability and to reduce latency. After the routing path for a bundle has been determined, it is stored in the bundle. Now, the routing path can be followed unless there is a discrepancy which occurs when a planned contact fails. Only in this case a new path has to be calculated.

2.3 RUTS Procedure

At first, the number of bundle copies has to be determined. Copies can increase the delivery speed and rate, because as a result of unpredicted disturbances it may happen that not every bundle can be routed on its calculated path. We define n as the number of bundle copies (including the original bundle). n is usually small, and $n \geq 1$. Consequently, $n = 1$ if no bundle copies are used. For each new bundle the transmitting node sets the initial value of n into an additional field in the bundle to save storage and transmission capacity for 'real' copies. As long as $n > 1$, the bundle may be split by a DTN node to follow a previously calculated path for each bundle copy. This means that a bundle is transmitted with an adjusted value of n to another node whereby multiple transmissions can be saved. Now, both nodes have the same bundle and the sum of available copies of both parts corresponds to the value before. Over the whole bundle lifetime, it must be ensured that the number of copies in the DTN network is always $\leq n$. In addition to the splitting, it is still possible to forward bundles to follow the calculated paths.

In order to send the bundles and their copies on different routing paths, possible ways have to be detected. For this k routing paths must be identified where typically $k > n$ to get a higher choice because each routing path can have a different quality. Furthermore, $k \leq n$ if not enough paths exist. To find the necessary paths RUTS uses the available timetables and the network map from the urban public transport system. This context information can be used to calculate possible contacts between vehicles. Each possible contact is stored in a special routing-graph. In the first step the search starts from the sender DTN node. All possible contacts are inserted into the routing-graph. Now, the search will be resumed for all nodes contained in the routing-graph. The search is iteratively continued and it is stopped if at least k potential paths are available. In section 2.4 the graph is explained in more detail.

Now, the k fastest paths from the transmitter to the destination must be found in the routing-graph. As already mentioned the paths vary in quality such as different contact probability and duration between the vehicles. Therefore,

the paths are evaluated and the n best ways are selected for forwarding. In addition a score value for each of the k -path is calculated. It reflects the estimated time required for the transport from the sender to the receiver, the number of required hops and the line characteristics for each DTN node. All these data can be obtained from the routing-graph and the available context information. For the line characteristics the driving directions and the kind of contact is crucial. In section 2.5, this issue is addressed in detail.

The score for each k -path is determined as follows: TT indicates the calculated transmission time from the transmitter to the destination, RPT defines the rating for a possible transmission and NT declares the number of bundle transmissions to reach the destination.

$$Score_k = TT + \frac{\sum(RPT)}{NT} \quad (1)$$

Each parameter can be parameterized to control its impact on the score. In addition, the rating-models for the possible DTN node transmissions are interchangeable. After each of the k paths has been rated, the bundles are sorted according to the scores and distributed on the most suitable paths.

2.4 Routing-Graph functionality

As described above, the routing-graph is used to determine possible routing paths. Figure 1 shows an example for such routing-graphs and the resulting paths whereas the nodes $A - H$ symbolize specific DTN nodes for any scenario. Therefore, the DTN nodes must be clearly distinguishable. In this graph, a bundle has to be transmitted from node A to node H . The directed edges represent expected contacts between the vehicles, so that e.g. a contact between the nodes A and C is possible. These contacts result from the analysis of the context information by comparing the positions and schedule time of the vehicles. As a special feature, the routing-graph stores the time when a node has been inserted into the graph, except for the destination node. In the example, the creation time of each node can be seen in the timeline and starts always with $t = 0$. Typically, the times represent minutes because the timetables are usually defined in minutes, too. This additional knowledge is needed to determine the weights from the edges. Here, the weights are the difference between the times of the timeline from two corresponding nodes. These represent the time which will probably elapse before it comes to the next appropriate contact. Later, the predicted time to transfer a bundle can be read in the sum of the weights from a routing path.

At the beginning in figure 1, the graph contains only the source (A) and the destination (H) node. The transmitter node is located in the area of $t = 0$ because the search starts always with this time step and represents the current time. The destination node is outside the timeline because it can not be assigned a fixed value. Instead, the currently considered time step is used for the calculation of the weights.

Based on the timetables and the network map, new possible contacts are iteratively searched for each time step and all nodes in the graph, except for the destination node. The search is terminated as soon as the destination node has at least k incoming edges. In the illustrated example, at least two potential paths should be sought ($k = 2$). For illustration the example shows 3 paths.

Once a possible new contact has been found for any node in the graph, the respective node is inserted into the graph.

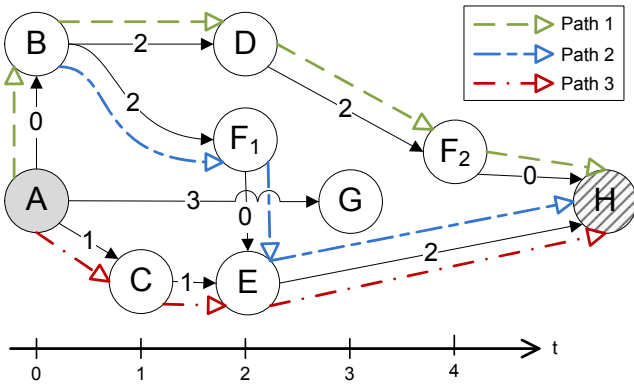


Figure 1: Example for a RUTS Routing-Graph

The position in the graph is dependent on the current considered time step. In addition, a directed edge must be inserted, from the node which found the possible contact to the newly added node. The weight results from the currently considered time step and the time step from the associate member node:

$$Edge_{Weight} = T_{Current} - T_{MemberNode} \quad (2)$$

Over the whole time it must be ensured that the graph is cycle-free to avoid invalid ambiguities. This requires a cycle test after inserting a new edge. If necessary, the last inserted contact must be removed.

The next issue is that nodes can have multiple incoming edges. These appear if more than one possible contact to a node exists. In figure 1, the node F is an example for this because the nodes B and D have a possible contact to this node. This can lead to ambiguities and invalid paths because a differentiation between possible contacts at different times is no longer feasible. Therefore, nodes must be split as soon as a second incoming edge is inserted. In the example the node F has a first contact at $t = 2$ with node B and a further at $t = 4$ with node D . After a node has been split, the further search for new contacts is performed only once. Therefore, not every single subnode must be searched individually. If new contacts are added, edges from each subnode towards the contact node are inserted. Multiple contacts between the same nodes at different times are possible. In such cases, only the first contact is considered because no additional benefits can be drawn from later ones. Normally, the search stops at k incoming edges of the destination node.

Moreover, gateways and relays can also be included in the routing-graph. Gateways are stationary nodes and allow transmissions to the backend system over a WAN. Thus, multiple backends can communicate directly with each other. Relays are also stationary but have no WAN connection. Through the usage of relays and gateways the delivery probability and reactions to external failures such as breakdowns and delays are improved. Once a node has a possible contact to any gateway, all gateways are inserted into the routing-graph as nodes with a minimum weight for the edge because an immediate transfer is possible. Relays are added individually because they have no direct connection among each other. Thus, the relays and gateways are involved in the further search.

Finally, the current delay from the transmitter node can be included in the search for possible contacts. This can be

deduced from the position of the vehicle and the timetables. Therefore, the identified delays must be considered while looking for new possible contact options by adjusting the own timetables with the delay.

2.5 Weighting of a possible bundle transmission

As already described in section 2.3, the probabilities of all possible contacts between vehicles must be identified and weighted. Afterwards, the determined value is considered in the score. Currently, RUTS uses a simple weighting model and differentiates between two use cases. The first case is shown in figure 2, the intersection of two lines. The course of the driving direction is insignificant. As there is no additional dynamic context information available such as the duration of previous contacts, a low contact probability has to be assumed. Thus, the weight for this possible bundle transmission is 1 which can be parameterized with an additional factor to regulate the impact on the score.

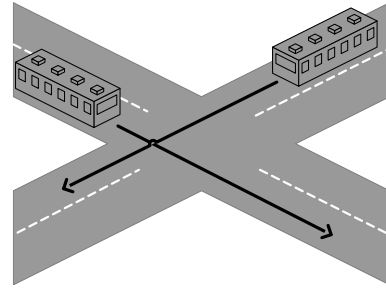


Figure 2: Bundle transmission in an intersection

Overlapping segments of the lines is much more complex. In this case lines overlap on a section of the network map. The length of the overlapping segments and the direction of the vehicles is of relevance. Figure 3 shows an example with vehicles which drive towards and after the contact they drive away from each other. The other possibility is that vehicles drive in the same direction. Both possibilities have a different contact probability and expected contact duration. The direction of the vehicles can be derived from the timetables and the network map. In both scenarios the weight for the score is determined by the length of the overlap. The length equals the number of edges in the routing-graph which cover the overlap.

The presented weighting model allows a simple differentiation between the probabilities of possible bundle transmissions. Moreover, it can later be supplemented by additional dynamic context information such as the average contact duration and contact probability which are determined from recorded contacts.

2.6 RUTS Extensions

We introduce further extensions to make the routing procedure more efficient. At the beginning of each contact, announcement messages are exchanged to identify the set of bundles already buffered on the other node. After bundles copies are distributed, it may happen in the case of unpredictable disturbances that two nodes making contact hold the same bundle. Instead of transferring the bundle again, an meta-bundle containing an update message is transferred. This updates the number of bundle copies (n) on the other

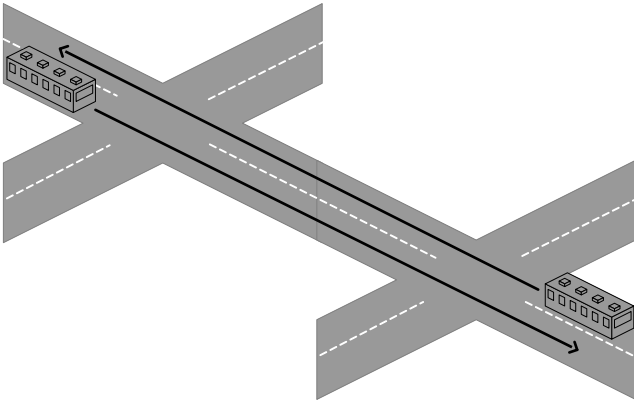


Figure 3: Bundle transmission by an overlapping

node and makes the transmission of the whole bundle unnecessary. Moreover, a node is informed of successfully delivered bundles. For this, a flooding mechanism is used with a hop-limit. The announcement message is also used to identify a vehicle and inform other nodes about the current state of a vehicle, e.g. its direction and delay.

A priority can be assigned to each bundle and determines the number of bundle copies and the transfer order. The transmission of bundles is in the order of priorities and within the priorities it is ordered by the bundle age. The bundle age is in ascending order to allow a delivery as fast as possible and to release resources as soon as possible.

Further optimizations are for the delivery to a gateway. This is worthwhile because the probability of a successful delivery is very high when the gateway is on the path of a vehicle. Only heavy disturbances can prevent the delivery because the amount of hops is minimal. Therefore, one bundle copy is kept permanently on the node until it makes direct contact with the gateway. Often this is not the fastest way to transfer a bundle, but it is more reliable and saves resources. All other bundle copies are sent as supplied before and may reach the destination faster but with higher uncertainty.

2.7 Implementation

We have implemented our routing protocol for ‘The ONE’¹ [4], a DTN simulator featuring several previously proposed routing protocols that we use for performance comparison. For RUTS, it is necessary to make some context information accessible on every DTN node, such as the network map and timetables. The timetables or other dynamic context information are stored in a database. For the network map and the routing-graph, the free Java Graph Library ‘JGraphT’² is used and allows for simple management of the graphs. Furthermore, it is included a simple search for the k-shortest ways and a check for cycles in the graph.

The routing paths are evaluated to calculate the score. For this purpose, the possible bundle transmissions are analyzed and rated. It is necessary to determine the type and quality of the encounter in consideration of the directions of each vehicle. This is provided through an evaluation of context information.

¹<http://www.netlab.tkk.fi/tutkimus/dtn/theone/>

²<http://jgrapht.sourceforge.net/>

For the management of the number of copies and the priorities of a bundle, the simulator allows to define additional fields in a bundle so that they can be used for this purpose. Further functionality has been added to update the number of bundle copies before it is transmitted and to restore it if a transfer fails.

Finally, in addition to the data bundles, the announcement and update bundles are needed. Again, the available additional fields in the bundles have been used to transmit the necessary data.

3. EVALUATION

3.1 Methodology

A common approach to the simulation-based evaluation of routing algorithms in mobile ad-hoc networks (MANETs) is the use of mobility models. These models allow reproducible large scale simulations in controlled scenarios. However, simple models such as random walk, random waypoint and levy walk generate almost unrestricted and unrealistic node movements. Map based models are better suited to describe vehicle movement in public transport networks, where vehicles generally move on a predefined route along roads or rails. An example is the map based working day movement model [2], which also features a bus submodel. Another approach to model mobility especially in DTNs is the extraction of contact and inter-contact times from real life traces. Zhang et al. [9] use traces obtained from the UMass DieselNet testbed to implement a generator of synthetic traces. However, the influence of disturbances like e.g. road congestion is very hard to model with the aforementioned approaches.

Therefore we chose a different approach to produce more realistic mobility traces. Several existing tools were integrated and we developed a tool to import timetables, lines and locations of bus stops. The resulting toolchain uses real cartographic material and performs a large scale micromobility simulation with real lines and timetables to generate very realistic vehicle movements. These are then imported into a DTN simulator and are used for the evaluation of various routing algorithms, including our own.

3.2 Simulation Toolchain

The aim is to create a most realistic scenario by integrating as many properties as possible from the urban public transport to a DTN simulator. These are the roads, rails, lines, timetables, stations and the number of vehicles. In the following the process of creating a scenario is described, drawing on the example of the city of Braunschweig in Germany.

First, the cartographical material from OpenStreetMap³ of the city area of Braunschweig are exported to the SUMO tool. SUMO⁴ is a generally recognized and flexible open source traffic simulator. With the corresponding tool ‘net-convert’ it is possible to convert the exported xml-file to a SUMO network and configuration file. Now these files are supplemented by the corresponding lines, stops, vehicles and their timetables from the urban public transport. For this purpose we developed the tool ‘BusNetGenerator’ which allows to add the required data in a simple way. It offers a

³<http://www.openstreetmap.org/>

⁴<http://sumo.sourceforge.net/>

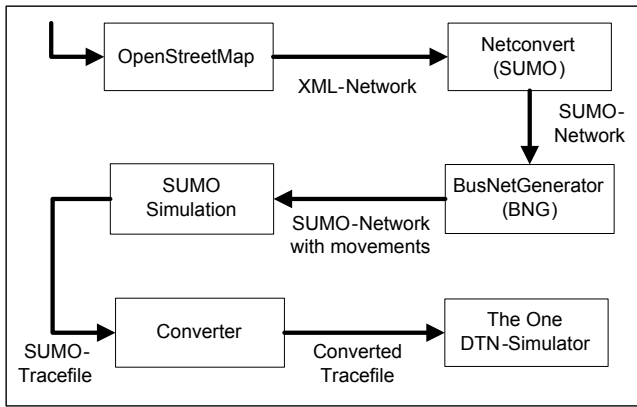


Figure 4: Used Toolchain to develop our scenario

GUI with Google-Maps to place the stops in the network. Subsequently, the lines can be created through a list of stops and each required vehicle is assigned to one of the lines. Finally, the timetables of the vehicles are integrated. The outcome is a new movement file for the SUMO simulator. Now SUMO can generate trace files with different properties and environmental conditions. Thereby traffic lights can be considered, intersection-based right-of-way rules and different vehicle types. Thus, highly realistic traffic simulations can be created, including disruptions in traffic flow. In the next step the movement files must be converted for ‘The ONE’, a DTN simulation tool which supports different routing protocols. Moreover, it is easily expandable by own statistics and routing operations. The conversion adapts the coordinate systems and matches the syntax for the input reader from the DTN simulator. After the last step a realistic simulation and comparison of the performance of routing protocols in urban public transport is possible.

3.3 Simulation Setup

The scenario has been simulated for 5 hours. In the first hour the vehicles were successively integrated into the simulation in a realistic way. During this startup period the vehicles are not instantly available but start serving the lines from the terminals. It takes some time until the network is fully populated and for this reason the startup period is ignored in the evaluation. The same applies for the shutdown period which is the last hour of the simulation. This is necessary because paths are calculated with timetables in the future.

The setup is as follows. The scenario has 54 stops and 13 lines which are defined by a series of stops. To each line one or several vehicles are assigned, 28 vehicles are simultaneously on the lines. The vehicular nodes have a local copy of the timetables and the network map. For a connection with a backend system two stationary gateways are available. Furthermore, there is a relay which is also stationary. In contrast to a gateway it has no WAN-connection. Through the usage of relays and gateways, the delivery probability and reactions to external intermittent failures such as breakdowns and delays are improved.

RUTS was implemented in Java for ‘The ONE’ simulator. Additional reports have been implemented to record all simulation results. Further adjustments were made to provide

a continuous connection between the gateways and to avoid communication between unused vehicles.

For the data exchange between nodes, the simulator can generate bundles in a constant interval at a random node. The destination node for each bundle is also random. We have investigated various intervals. For this purpose all information about the bundle creation were recorded and reused so that all routing methods have been tested under the same conditions. Finally, the parameterization has been optimized. In addition to the original bundle two additional copies are used. In the routing-graph five possible paths are determined. Thus, $n = 3$ and $k = 5$.

3.4 Simulation Results

For the evaluation various performance criteria are used to compare the different routing protocols. First, the number of successfully delivered bundles is considered as well as the bundles that never reached their destination because all bundle copies were discarded. Moreover, bundles may exist that have not yet been delivered because the simulation was terminated.

The next criterion is the network load. It increases with the number of transmitted bundles and may prevent the transmission of other important bundles. In addition, it has a direct impact on the memory requirements on the nodes. Furthermore, the average time to transport a bundle from the source to the destination is investigated. The number of delivered bundles has a direct influence on this value. Hence, these values must be examined jointly. Finally, the average memory consumption on the nodes over the entire simulation is of concern. This value has an impact on the number of discarded bundles and the number of transmitted bundles. Consequently, this has far-reaching implications.

Most performance criteria have an impact on others. Moreover, some routing protocols obtain very good results in certain areas but drop back in other fields. Therefore it is important to consider the set of criteria as a whole.

For the evaluation, different bundle intervals were used. The interval defines how often new a bundle is generated on a random node. The evaluation shows the results for a large set of bundles. Here, the used bundle interval is 5 seconds. The analysis under heavy load is interesting because the resource requirements are significantly higher. Some of the routing protocols reach their limit already at this bundle interval while others still achieve good results.

In figure 5 the successfully delivered bundles are illustrated as well as the lost bundles and the bundles remaining in transit at the end of the simulation. These sum up to 100% and are equal to the amount of generated bundles. In the comparison of all routing protocols, RUTS can deliver the most bundles and also causes the lowest bundle loss. Also, the number of remaining bundles is low.

Figure 6 shows that RUTS occupies lesser storage capacity than nearly all other routing protocols. Only FirstContact takes slightly less memory in our simulation because it uses no additional bundle copies. However this causes a low delivery rate. All other protocols take significantly more memory resources.

Figure 7 compares the transmission times of successfully delivered bundles. For each simulation run a set of randomly chosen sources with random destinations is calculated. The same set is used for all routing algorithms. Therefore the results are reproducible and comparable. Due to the ran-

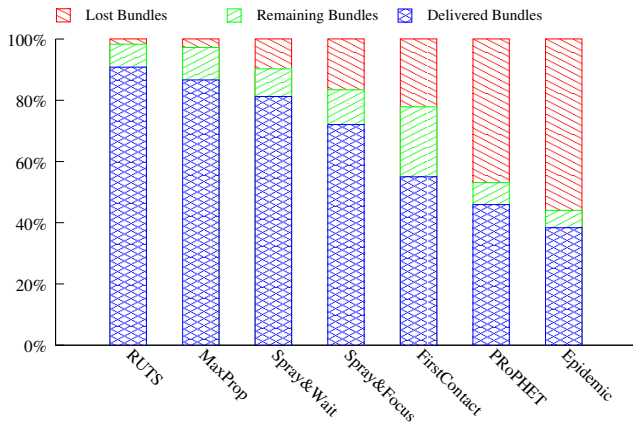


Figure 5: Lost, remaining and delivered bundles

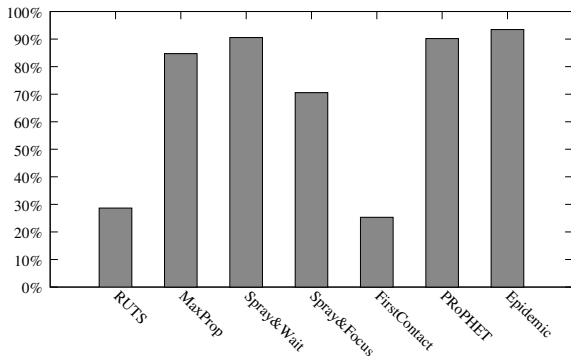


Figure 6: Average memory usage

dom selection of the source/destination pairs. It is possible that bundles are generated on nodes which are in contact with the destination. These bundles can be delivered immediately, so that the minimum transmission time is very low. Average and maximum transmission times can only be calculated for bundles which are delivered before the simulation ends. Therefore transmission time can only be interpreted in relation to the amount of delivered bundles, because a higher number of delivered bundles increases the probability that bundles with high transmission times are included. In this context RUTS achieves a low average transmission time, although it has the best delivery ratio.

To investigate the impact of disturbances in the public transport system several variations to the scenario were generated. 20% of the vehicles are chosen at random and are assigned a random delay of 1-5 minutes. Thereafter the result is calculated as the average of several simulations. The same procedure is used for potential breakdowns of some vehicles. For this purpose, 10, 20, 30 and 50 % of the vehicles are chosen at random and removed from the simulation. As a result, no bundles to the destination vehicles can be delivered and hence these remain longer in the network. On the other hand, the removed vehicles can no longer emit any bundles.

Figure 8 shows the differences between the previous results and the average results with random delays. On the left side the results for the delivered, lost and remaining bundles are compared and on the other one the average memory usage is

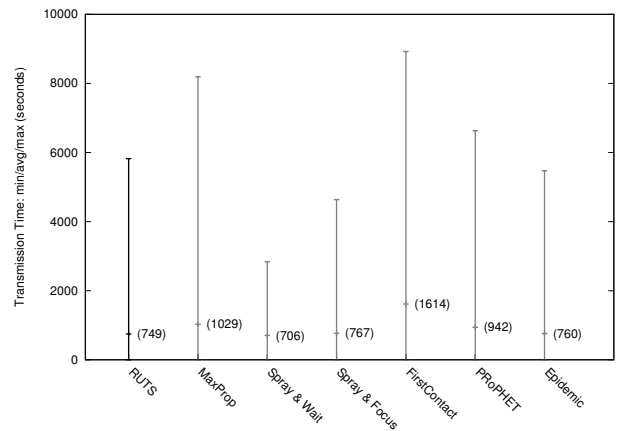


Figure 7: Transmission Time (min/avg/max)

contrasted. As you can see, the differences are very marginal (<1%). This is because RUTS creates a low but sufficient number of bundle duplicates which are routed via another path, so that it is robust against disturbances.

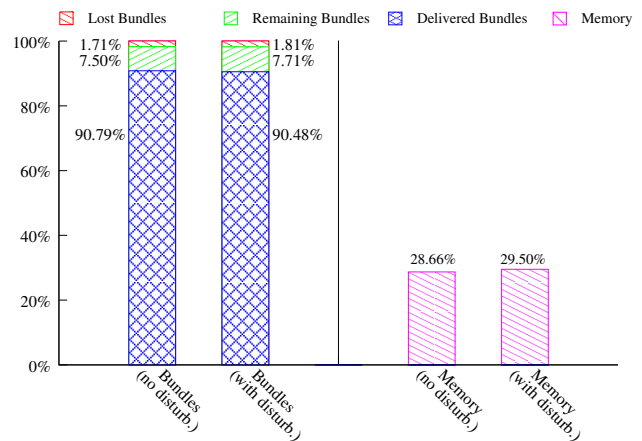


Figure 8: Impact of disturbances

The same also applies for the simulated breakdowns in figure 9. The lines converge with an increasing number of removed vehicles. Nevertheless, the RUTS can deliver successfully the most bundles under all conditions.

The performance of RUTS is very promising taking into account all performance criteria. In all investigated areas RUTS is at least similar to the best performing routing protocol. In most cases it archives the best results. RUTS can also deal well with a higher bundle load, whereas all other routing protocols struggle.

4. CONCLUSION

In this paper we presented RUTS, a DTN routing algorithm which exploits the characteristics of urban transportation systems. The evaluation results show that low resource utilization, low latency and high delivery rates are not conflictive objectives. One could argue that RUTS profits only from the known timetables, but the performance decreases only marginally even if 20% of the vehicles are be-

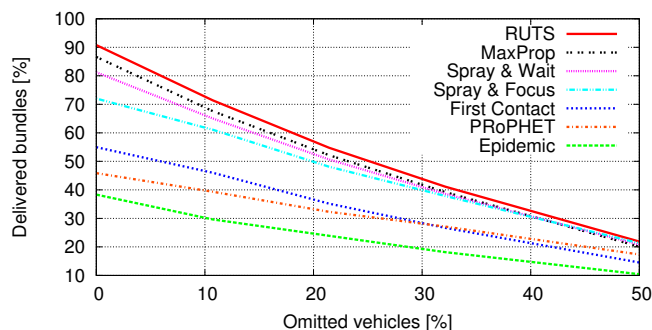


Figure 9: Impact of breakdowns

hind schedule or 10% fail completely. Another reservation against RUTS could be its high level of specialization. However, DTNs are deployed in very special scenarios. Generic approaches like epidemic routing work in almost any scenario, but the performance is often poor. In our opinion the increased performance of RUTS outweighs the advantage of a generic algorithm. We also presented our approach to generating complex mobility traces, going beyond simple models by integrating an existing map based micromobility simulator. The evaluation scenarios feature real cartographic data, lines, timetables and stations. One of the scenarios is a model of a planned large scale testbed. In future work the performance of the routing algorithms will be investigated in this real world testbed and compared to the simulation's results.

5. REFERENCES

- [1] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–11, 2006.
- [2] F. Ekman, A. Keränen, J. Karvo, and J. Ott. Working day movement model. In *MobilityModels '08: Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models*, pages 33–40, New York, NY, USA, 2008. ACM.
- [3] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 145–158, New York, NY, USA, 2004. ACM.
- [4] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA, 2009. ICST.
- [5] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.
- [6] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops'07. Fifth Annual IEEE International Conference on*, pages 79–85, 2007.
- [7] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, New York, NY, USA, 2005. ACM.
- [8] A. Vahdat and D. Becker. Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical Report CS-200006, Duke University, 2000.
- [9] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang. Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 195–206, New York, NY, USA, 2007. ACM.