Increasing Reality for DTN Protocol Simulations

Ari Keränen and Jörg Ott
Helsinki University of Technology, Networking Laboratory
{akeranen,jo}@netlab.tkk.fi

Abstract— **Powerful personal devices provide the basis for ad-hoc networking among mobile users. Delay-tolerant Networking (DTN) enables such communication in spite of low node density—to reach an infrastructure network as well as for direct information exchange between peers. Numerous DTN routing protocols have been developed, and their analysis has shown different performance depending on the (human) mobility assumed—ranging from simple to complex mobility models to a variety of real-world traces. We have designed a simulation environment that allows incrementally adding bits of reality to mobile (DTN) simulations, running different DTN routing protocols, and interactively visualizing the results. It interfaces to various trace formats on the input and different other simulator engines (ns2, dtnsim2) on the output side. Using this simulator, we analyze the characteristics of communication opportunities and compare four different DTN routing protocols under increasing reality conditions.**

I. INTRODUCTION

Today's powerful personal devices enable mobile users to communicate via infrastructure networks (cellular, WLAN) as well as to form ad-hoc networks. Particularly modern mobile phones are virtually always turned on and offer sufficient processing power, storage capacity, and battery lifetime. But interpersonal mobile ad-hoc networking cannot rely on a sufficient dense node distribution to establish end-to-end paths between two peers wanting to communicate (and the performance of multihop wireless communication seriously degrades with an increasing number of wireless hops) [1]. Therefore, the concept of Delay-tolerant Networking [2], [3] is being widely applied to such mobile environments for applications that do not need instant end-to-end interaction. DTNs offer asynchronous communication in which arbitrarily sized messages are sent by the originator and stored by intermediate nodes (and possibly physically carried) until a suitable next hop or the final destination becomes available to forward the message. This is also referred to as *store-carry-and-forward* paradigm.

Depending on how the mobile nodes move, how dense the node population is, and how far sender and receiver are apart, message delivery times may vary substantially from a few minutes to many hours or days. Communication performance heavily depends on the routing and forwarding algorithms in use and how well their assumptions match the actual mobility patterns: no ideal routing scheme has been found so far.

Simulations play an important role for analyzing the behavior of (not just) DTN routing protocols at larger scales beyond the reach of experimentation and supply a controlled environment with repeatable conditions. They also allow quicker assessment of protocol properties without the manifold overhead of full implementations. With typically sparsely distributed nodes, DTN simulations abstract from the details of the wireless communication characteristics and make the simplifying assumption that two nodes can communicate when they are in range of one another. This allows focusing on the evaluation of the DTN protocols—an approach we follow in this paper, making conservative assumptions about the achievable data rate, the radio range, and thus the resulting transfer volume.

The typically sparse node population, however, makes the mobility model even more important for assessing the performance of DTN routing protocols, typically measured in terms of message delivery probability, delay, and overall protocol overhead. Many synthetic mobility models have been suggested to support easy protocol evaluation. These range from simple ones (such as random walk or random waypoint) to group mobility and community-based models (see [4] for an overview) to complex vehicular models [5] to, albeit with a different focus, even complete models of small populations and their activities [6]. In DTN protocol evaluations, these mobility models are complemented by the use of real-world traces: human mobility patterns are monitored by means of WLAN access points or Bluetooth contacts observed by mobile nodes (such as motes or phones) [7], [8].

While synthetically generated node mobility models allow for fine-tuning in many respects, they usually cover only selected mobility characteristics. In contrast, real-world traces often have only coarse temporal (e.g., scanning intervals in the order of minutes) or spatial (e.g., only covering a campus area) resolution.

In this paper, we start out from synthetic mobility models and then incrementally add real-world aspects constraining and varying node motion characteristics for evaluating a set of DTN routing protocols. Our contribution is threefold: firstly, we introduce a new modular simulation tool, *ONE*, encompassing a mobility generator, DTN simulator, and visualizer (figure 1) that can also interface to existing other simulation tools and real-world traces. The mobility generator allows constructing city scenarios in which we stepwise add reality by using real-world street maps, different classes of nodes (pedestrians, vehicles, buses [9], stationary nodes such as throwboxes [10]) with different radios, schedules, routes, motion preferences, among others, and allowing importing mobility data from other simulations (such as TRANSIMS [6]).

Secondly, we analyze the mobility and contact characteristics arising from the enhanced mobility schemes. Finally,

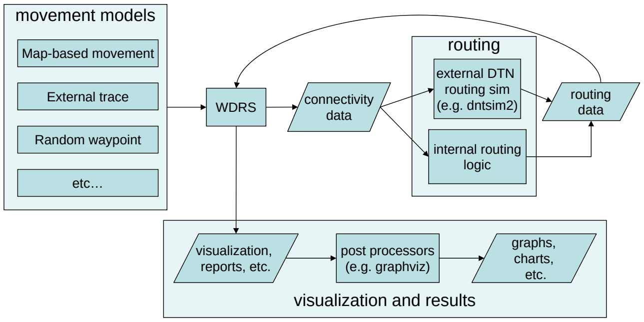


Fig. 1. Schematic overview of the ONE simulation environment

we use the our DTN simulation engine to evaluate the performance of different DTN routing protocols when increasing "reality" of our simulations.

This paper is structured as follows: in section II, we review related work on DTNs and (related) simulations for mobility. We introduce our simulation environment *ONE* in section III based upon which we evaluate four different DTN routing protocols in section IV and the impact of stepwise approximating reality. We conclude this paper in section V with a brief assessment and point out future work.

II. RELATED WORK

In this paper, we focus on communication performance in delay-tolerant ad-hoc networks made up of mobile nodes. Delay-tolerant Networking [2] is increasingly applied to enable communication in challenging networking environments, including sparse sensornets and opportunistic mobile ad-hoc networks. The DTNRG architecture [3] proposes a *bundle* layer as an overlay to bridge different (inter)networks. Nodes communicate via asynchronous messages of arbitrary size which are exchanged using the store-(carry-)and-forward paradigm. Messages have a finite TTL and are discarded when the TTL expires. They may also get dropped from a node's queue due to congestion, yielding a best-effort service. Application protocols need to tolerate the delays resulting from the challenged environment and the risk that messages are not delivered in time or not at all. Typical performance metrics for evaluating DTN protocol performance are hence message delivery probability and latency.

Numerous routing and forwarding schemes have been proposed over the past years (refer to [11] and [12] for overviews). Different mechanisms are usually applied depending on whether the network is primarily of mobile ad-hoc nature (e.g., mobile devices carried by humans) or is based upon a (fixed or mobile) infrastructure (e.g., space networks, bus networks). Obviously, mixed networks exist as well (e.g., mobile users supported by infrastructure nodes).

The primary difference between various DTN routing protocols is the amount of information they have available to make forwarding decisions [13]. Ad-hoc DTNs usually apply variants of reactive protocols. Flooding protocols such as epidemic routing [14] do not use any information. Predictive protocols such as PRoPHET [15] use past encounters of nodes to predict their future suitability to deliver messages to a certain target whereas other protocols also exploit further (explicitly configured) schedule and context information per node [16]. Furthermore, they differ in their *replication strategies*, i.e., how many copies of a message they create which, in turn, has a direct impact on the load incurred on the network. Some protocols generate just a single copy [17] (e.g., First Contact [13], Direct Transmission/Delivery [17]), others a fixed number limited by the sender [18] [19] while epidemic [14] and probabilistic [15] routing potentially create an "infinite" number of messages.[1] Scheduling strategies govern in which order messages are passed when a communication opportunity occurs between two nodes. Finally, queue management strategies define when and which messages are deleted, e.g., if congestion occurs.

For evaluating the performance of DTN routing protocols, manifold settings have been used, mostly including some type of node mobility. Mobility has been created (a) from synthetic mobility models, (b) taken from traces obtained from real-world measurements, and (c) by evaluating code in the real-world. While a few testbeds for (c) exist (such as DieselNet [9]) their flexibility is usually limited, large-scale operation "expensive", and their use is typically limited to those running the testbed. Such testbeds may also be used to obtain real-world traces (b) which can then be made available to other researchers.

Various projects have collected traces of contacts (peers, times, durations, and possibly positions) between Bluetooth devices [7], between users and/or wireless access points [8], among others. The CRAWDAD project [20] provides a repository where numerous real-world traces are available. These traces offer insights into real-world interactions between mobile users from different angles and constitute a valuable data source for validating (or invalidating) the mobility and connectivity characteristics obtained from synthetic models.

But also real-world traces have their limitations as—so far—the population analyzed in these traces is naturally very limited and may thus bias the results. Furthermore, the time granularity is often limited in order not to drain mobile device batteries too quickly: e.g., the Haggle iMotes uses sensing intervals of 5 min so that many contact opportunities may easily go undetected and contact durations can only be assessed equally coarsely.[2] Finally, the results cannot be arbitrarily scaled up, thus limiting what we can evaluate.

The only option for flexible and scalable simulations is thus (a) model-based synthetic mobility generation. As noted above, the models range from simple ones such as *Random Waypoint (RWP)* to group-based models to community models with major points of interest [4] to vehicular ones taking street maps into account (e.g., [5]). Node velocity and pause times may be adjusted to match pedestrians, vehicles, or other node types. Specific models for vehicular networking furthermore

---

[1] Source and network coding schemes allow for more efficient message replication but have not been investigated for this paper.

[2] This may also be considered a feature as end user devices wanting to communicate may suffer similar power constraints.

consider additional constraints from simple road setups to real-world maps on one hand and simple non-interfering vehicles to vehicular interaction (distance, speed) based upon traffic flow models on the other.

In other areas (e.g., for epidemic spreading studies or traffic planning), more complex simulation models have been created mimicking the behavior of the population of an entire city [6]. Depending on the precise setting, the latter may not have the proper focus for evaluating ad-hoc interpersonal communications: TRANSIMS, for example, allows modeling a population and their interaction *at* certain locations or *in* public transportation vehicles, but does not include details on the way *between* such locations, which limits the suitability of the generated mobility data of pedestrians. In the case of TRANSIMS, detailed vehicle information could be made available and has been used for investigating MANET protocols [21].

In summary, while we have a plethora of tools available, testbeds, real-world traces, or simulation models address only one particular subsets of users: traces and testbed are limited in their coverage and granularity, vehicular mobility models do not consider pedestrians and vice versa, constraints of maps are often not taken into account, schedules not always considered, and so on. Most importantly, while all facets can probably be addressed by different models, traces, and setups, these usually come from different areas and thus do not allow incremental addition of further features to stepwise increase the degree of reality of the simulations in a controlled environment. We take one more step into this direction by introducing ONE as a simulation environment that supports flexibly modeling node mobility in the environment of any city—which we use afterward to assess the impact of increased reality on DTN (routing) protocol performance.

III. ONE: SIMULATION ENVIRONMENT

The simulator is a customizable Java program that is capable of 1) generating node movement using different movement models, 2) routing messages between nodes with various DTN routing algorithms and sender and receiver types, and 3) visualizing both mobility and message passing in real time in its graphical user interface. ONE can import mobility data from real-world traces or other mobility generators. It can also produce a variety of reports from node movement to message passing and general statistics. All movement models, routing algorithms, and report modules are loaded dynamically based on the given configuration files so that extending the simulator with new modules and changing the modules used in different scenarios is made easy for users and developers.

In a simulation setting, any number of types of (mobile) nodes—referred to as a *node group*—may be defined. A node group shares a common set of simulation parameters like speed and pause time distributions, message buffer size, and radio range, among others. Different node groups can also utilize different movement model modules and, to some extent, also different routing algorithms so that also heterogeneous environments may be created and explored.

A. Mobility Modeling

Currently, ONE supports the basic Random Waypoint [22] mobility model, arbitrary mobility models by using externally generated movement data, and different map-based movement models. All map-based movement models get their input data using files formatted with a subset of the Well Known Text (WKT) format. WKT files can be edited and generated from real world map data using Geographic Information System (GIS) programs such as OpenJUMP [23]. With map-based movement models, the nodes move using roads and walkways from the map data. In addition, different node groups can be set to use only certain parts of the map, thus allowing to distinguish between cars and pedestrians so that the former do not drive on pedestrian paths or inside buildings.

In the simple random map-based model, *MBM*, nodes move to randomly determined positions on the map but follow the roads as defined by the map data. The shortest path map-based movement model, *SPMBM*, uses the same map data but instead of wandering randomly around the map, nodes use Dijkstra's shortest path algorithm to calculate shortest paths from the current location to a randomly selected destination. Additionally, map data can contain Points of Interest (POIs). POIs are places on the map area and, for each node group, separate probabilities can be defined for choosing a POI from a certain group for node's next destination. These POIs can be used to model e.g. tourist attractions, shops, restaurants etc.

Finally, some nodes may have pre-determined routes in the map that they follow. This route-based movement model uses the same map data but, instead of selecting destination map nodes in a random manner, nodes always select the next destination on the route they are currently traveling. This mode of movement is useful for modeling e.g. bus and tram routes. Both POIs and routes can be defined using any WKT-compatible GIS-program.

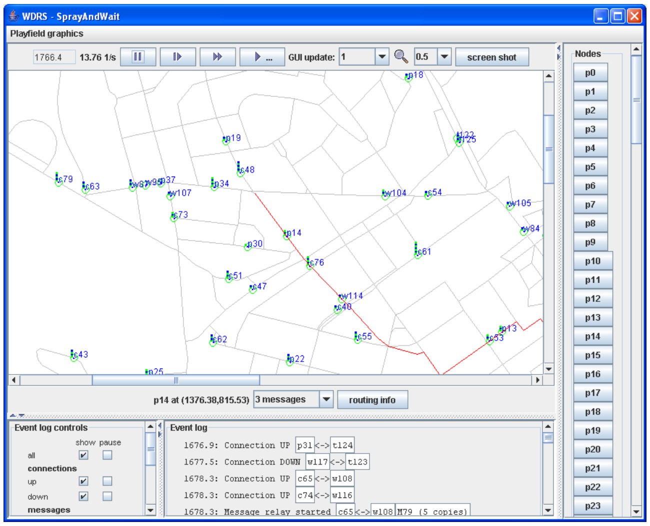
B. Routing simulation

For message routing, several built-in routing modules are available. Alternatively, external simulators such as *ns2* or *dtnsim2* [24] can be used. For external programs, a separate (but usually simple) parser is needed to convert message events to a form that ONE understands. We have designed a basic one for importing the output of dtnsim2. If ONE is used to create contact schedules for an external router, the resulting routing decisions can be inspected in the GUI synchronized with node movement.

Built-in routing modules include *direct delivery* [17], *spray and wait* [18] (normal and binary), *epidemic* [14] and *PRoPHET* [25]. All protocols transfer messages to the final recipient in case of meeting it, but differ on the way how other messages are handled. The direct delivery protocol does not transfer any messages but the ones for which the contact is also the final recipient of the message. Epidemic routing tries to forward all those messages that the other node does not have. Messages are exchanged in random order until the connection breaks (because of mobility) or until both nodes have all the messages. Spray and wait creates only a certain number of copies of each message to be transferred to other nodes. In the normal mode, the number of copies left is reduced by one on each transfer and in binary mode the number is halved each time. Only messages that have more than one copy left are forwarded to other nodes than the final destination. PRoPHET uses information about the previous contacts to predict how good candidate a node is to deliver the message to the recipient. In the current implementation, only the messages for which the other node has a higher probability of delivery are transferred (GRTRMax described in [25]).

C. Visualization

ONE is able to visualize results of the simulation in two ways. If the user wishes, the whole simulation is shown in real-time within the GUI as shown in figure 2. Node locations, current paths, connections among nodes, number of messages carried by a node, etc. are visualized in the main window. If a map-based movement model is used, also all the map paths are shown in the GUI. An additional background image (e.g. a raster map or a satellite image of the simulation area) can be also shown in the background. The view can be zoomed and the speed of the simulation can be adjusted interactively. The GUI also produces a log of events in the simulation and these events can be filtered by their type. The whole simulation can be configured to to pause when a certain kind of event occurs. A single node can be selected for closer inspection by selecting it from a list or from a log message. This allows retrieving further information about the messages node is carrying and about the routing module's state.



```
1676.9: Connection UP p31<->t124
1677.5: Connection DOWN w117<->t123
1678.3: Connection UP c65<->w108
1678.3: Connection UP c74<->w116
1678.3: Message relay started c65<->w108 M79 (5 copies)
```

Fig. 2. ONE Screenshot (map data copyright: Maanmittauslaitos, 2007)

Another way to visualize simulation results is to generate images from simulation reports. ONE contains report modules that generate adjacency graph and message path graph files that can be directly fed to Graphviz [26]. Adjacency and message path graphs visualize the connections among hosts throughout the simulation and provide an intuitive view of node interactions. Other included report modules generate numeric statistics of node connectivity, movement and message transfers. The simulation package includes tools that can be used to extract and combine statistic from multiple different scenarios and to create plots using, e.g., gnuplot [27].

D. Interfaces and Configuration

In addition to visualization and statistics, the reporting interface can be also used to interface with other programs. The connectivity pattern that is determined by node mobility and radio range can be reported in a form that is suitable for dtnsim2 (or any other simulator that understands the same syntax). For any other syntax, writing a suitable reporting module takes usually only a few minutes. This allows ONE's mobility models to be easily utilized by other simulators, too. If reporting link connection status is not enough, plain movement data can also be exported from the simulator. A reporting module for creating mobility traces that are suitable for *ns2* simulations is already available. The behavior of report modules can be adjusted further using the simulator configuration files.

The simulator is configured using simple text-based configuration files that contain the simulation, user interface and reporting parameters. Many of the simulation parameters can be configured separately for each node group but groups can also share a set of parameters and only alter the parameters that are specific for the group. The configuration subsystem also allows defining an array of values for each parameter hence enabling easy sensitivity analysis: in batch runs, a different value can be chosen for each run so that large amounts of permutations are easily explored. The messages that are created during the simulation are specified in an external file to enable flexible definition of creation patterns. Besides defining the created messages' parameters, also expected replies (if any) size can be defined and this way also request-reply scenarios can be investigated. A tool for creating simple message distributions is included in the simulator package.

IV. SIMULATIONS

For our simulations, we assume interpersonal communication between mobile users in a city using modern mobile phones or similar devices that are typically always turned on. We assume up to 20 MB of free RAM for buffering messages in these phones (which could be expanded to some extent by flash memory cards, but their capacity may be used up by permanent data). Users travel on foot or in cars. Dedicated supporting communication devices (with higher storage capacity) are built into trams or operated as standalone devices (throwboxes). All users are assumed to have Internet access at home or in their hotel so that messages will get delivered over night (and buffers emptied). We define a day to last for 12 hours (e.g., from 9:00 to 21:00) and are interested in the connectivity opportunities and message delivery during such a day.

A. Mobility and Connectivity

We have chosen part of the Helsinki downtown area (4500×3400 m) as depicted in figure 3. In addition to normal roads, we have added to the map data some paths to parks and shopping malls for pedestrians and tram routes for trams. We run our simulations with 100 and 500 nodes spread out across the above area. We use several different types of mobile nodes: *Pedestrians* move at random speeds of 0.5–1.5 m/s with pause times of 0–120 s. *Cars* are optional and, if present, make up 20% of the node count (the rest are pedestrians); they move at speeds of 10–50 km/h, pausing for 0–120 s. 0, 2, 4, or 6 *trams* run as speeds of 7–10 m/s and pause at each configured stop for 10–30 s. All random variables are uniformly distributed across the respective (inclusive) intervals. Finally, we add up to six stationary *throwboxes*.

As mobility models, we choose the four ones introduced in section III-A: we use RWP (for which we use the entire area, ignoring that parts of the map section are covered by the Baltic Sea), MBM, and SPMBM as described for pedestrians and cars and add trams based upon a sequence of points of interest. Additionally, we define two scenarios using different POIs for which we subdivide the nodes into five groups and create four POI groups (*west* containing 3, *parks* 11, *central* 4, and *shops* 22 POIs):

- POIs1: One node group runs MBM, three choose their next destination with a probability $p = 0.1$ for each of the four POI groups, the last remaining one (which can be either a car group or a pedestrian group) only chooses from the POI groups that are accessible by car (avoiding indoors and parks); otherwise, a random target is selected.
- POIs2: We define a preferred POI group for four of the node groups. A node chooses a POI with $p = 0.4$ from its preferred POI group, with $p = 0.1$ from each other POI group, and otherwise a random target.

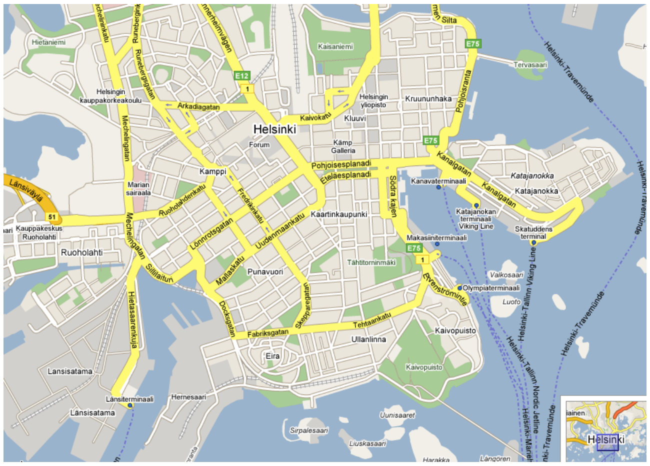


Fig. 3. Helsinki simulation area

For connectivity, we assume Bluetooth (10 m range, 2 Mbit/s) and a low power use of 802.11b WLAN (30 m range, 4.5 Mbit/s)[3] and do not account for peer detection, initialization, and autoconfiguration.

We have generated node mobility traces for all the permutations of the mobility models, radio ranges, and cars as described above for 100 and 500 nodes. Because of the observed similarities between the different non-random map-based models, we have investigated the impact of trams and throwboxes only for SPMBM. For every combination, we have chosen seven runs using different random seeds and report the mean values.[4]

To characterize the impact of adding the different degrees of mobility, we have analyzed the inter-contact times, the contact durations, and the total connectivity time between any two nodes.

We observe that the node density and the radio range do not fundamentally affect the connectivity distribution: with five times the number of nodes, the contact frequency increases but the relative distribution of contact durations and inter-contact times remains unchanged. As expected, increasing the radio range from 10 m to 30 m, simply shifts the distribution of contact durations by a factor of three. We restrict most of our reporting on contact characteristics to only the 10 m radio range and 500 nodes. We also observed that including 2, 4, or 6 stationary throwboxes changes the results only marginally compared to the case with only pedestrians and hence leave them out of the reported contact figures.

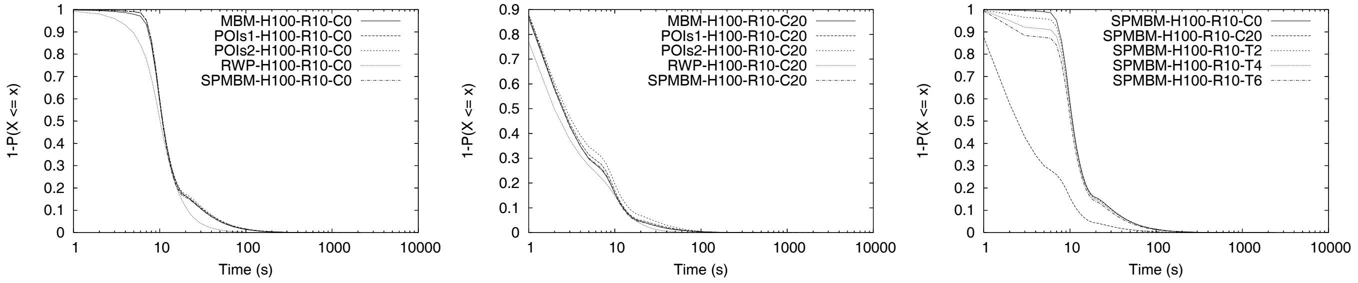
We plot the relative distributions of inter-contact times and contact durations in figure 4 for all mobility models without (a and d) and with cars (b and e) and show the impact of trams for the SPMBM model (c and f). We observe that RWP leads to shorter contact durations because nodes move into arbitrary directions rather than along roads; all other mobility models are fairly similar. The inter-contact times are greatly reduced if nodes move randomly on a map (MBM, d), simply because going back and forth makes them meet more often. Adding cars shows a reduction of the contact durations as short contact times increase due to the difference in node velocity (b). Adding cars also shows that inter-contact times for random mobility models diverge from non-random ones (e): MBM decreases the overall inter-contact times whereas RWP increases them.

For SPMBM, adding trams (we expect buses to have a similar effect) decreases the characteristics of short-term contact durations with an increasing number of trams (c) because, again, trams create many contacts which are of short-term nature as do cars. The more trams or cars are around, the stronger the relative importance of short-term contacts grows. Trams and cars decrease the inter-contact times and thus improve connectivity (f) as they move faster and thus get in contact with each other and with pedestrians more often.

---

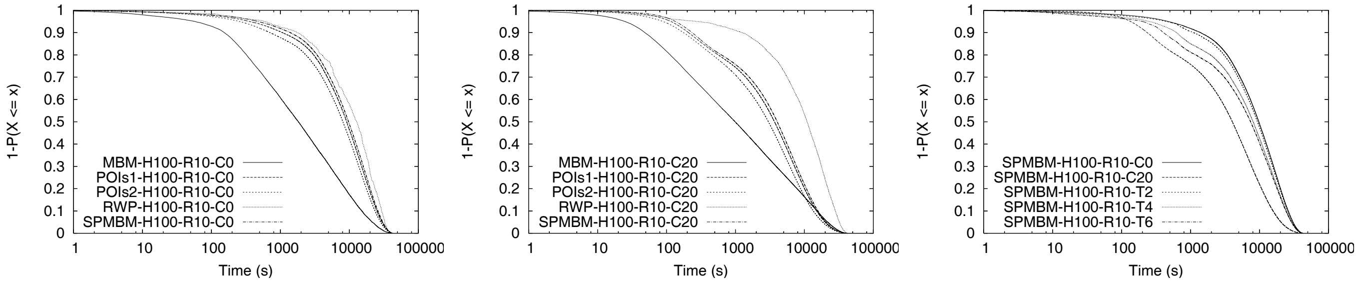
[3] Measurements reported in literature and performed by the authors yielded TCP net data rates of 3–7 Mbit/s, so that 4.5 Mbit/s is a reasonably conservative approximation. We choose low transmission power to avoid extensive interference if such devices get deployed widely and, also accounting for obstacles, thus limit our resulting radio range to 30 m.

[4] We have run all simulations with different time granularity of 1.0 s and 0.1 s and note that the results are similar for our settings.


(a) All contact durations, no cars

(b) All contact durations, with cars

(c) SPMBM contact durations


(d) All inter-contact times, no cars

(e) All inter-contact times, with cars

(f) SPMBM inter-contact times

Fig. 4. Mobility characteristics

These effects become more pronounced if their ratio increases as we have seen from the traces with just 100 nodes.

Figure 5 shows the significant differences in the total contact durations. RWP[5] and MBM yield a by far lower total contact duration than the more deterministic mobility models. Adding points of interest increases contact periods further (POIs1) and so does concentrating node group activities (POIs2). From SPMBM (but also from POIs1 and to some extent POIs2), we also observe that diversity in motion velocity due to trams (T2, T4, T6), cars (C100), and throwboxes (B2, B4, B6), does not alter the total contact durations significantly. Additional trams or throwboxes yield an increase in total contact duration because nodes are added, cars slightly reduce the contact times as they pass each other more quickly. Overall, the total contact times are maintained, but they shift towards shorter periods occurring more frequently.

When comparing the total contact times between runs of 100 and 500 nodes, we observe that the total contact duration grows by an (expected) factor of 21–28, i.e., roughly by the square of the node count increase. For increasing the radio range from 10 m to 30 m, we observe that the map-based mobility models show a similar increase (factor 3.0–3.9) whereas RWP increases roughly by the square (factor 9.0–9.4), which can be explained by the fact that RWP nodes move in all directions whereas map-based nodes usually only pass each other along the roads. For the map-based models, the factor larger than 3 can be explained by intersections and free spaces where nodes have more degrees of freedom.

For DTN communication protocols, this means that, on one hand, mobile node diversity (as well as added fixed nodes) increases connectivity and thus should have a positive impact on communications. On the other hand, it leads to reduced contact duration which should thus favor communications using shorter messages or demand (reactive) message fragmentation as supported by the DTNRG bundle protocol [28].

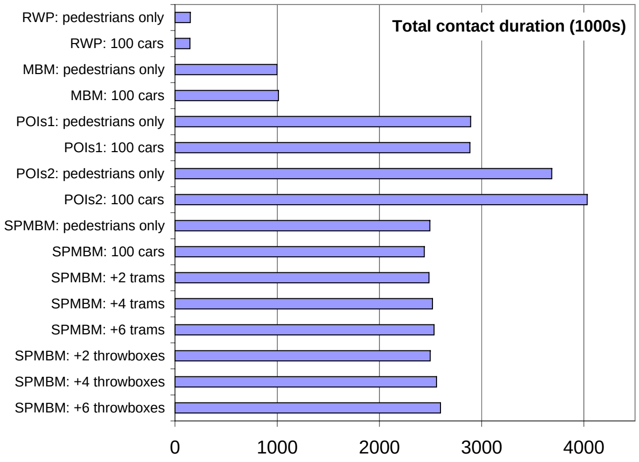


Fig. 5. Total contact durations (500 nodes, 10m radio)

B. DTN Routing Protocols and Message Passing

We have currently implemented four different routing protocols in ONE, all of which we consider in our evaluation.[6] DirectDelivery (DD) and Epidemic (E) only have one mode of operation. PRoPHET (P*nn*) can be parameterized with the transitivity parameter β, for which we choose $\beta = 0.25, 0.01, 0$ (0.25 being suggested in [25]). Spray and wait is available in

---

[5] Remember that RWP uses a larger area for nodes to move.

[6] We are in the process of implementing MaxProp but could not include its results in this paper.

the *normal* (SW*nn*) and the *binary* (SWB*nn*) variant, both of which we run with 6, 9, and 12 message copies. This results in a total of 11 routing protocol alternatives.

We use mostly mobile nodes (pedestrians, cars, trams) plus a few stationary throwboxes as discussed above. According to our above considerations, human users (as pedestrians and in cars) have either (lower end) mobile devices with 10 m radio range and 5 MB of buffer space or (higher end) 30 m radio range and 20 MB buffer space. For these settings, trams have 50 MB and 200 MB buffer space, respectively.

Mobile users (not the trams or throwboxes) generate messages to a random destination on average once per hour per node (uniformly distributed). We use message lifetimes of 3, 6, and 12 hours to obtain delivery during our simulation day (the originator could keep a copy, e.g., at the application layer and send it via the Internet during the night). We use message sizes uniformly distributed between 100 KB (elaborate text message with attachment) and 2 MB (digital photo). We run the simulations with all the mobility traces generated above (using the seven different random seeds) and report below on the one-way message delivery characteristics for a subset of these 1320 resulting permutations.

When analyzing our simulation results at large, we observe that increasing the message lifetime from 3 to 6 hours provides up to 30% gain in message delivery rate—particularly for DD because of the longer opportunity window to meet the target—but loss of up to 20% for epidemic and PRoPHET—due to buffer overflows. Further increasing the lifetime does not create additional gain (which is not surprising given our simulation time). As a result, we only consider 3 hour message lifetimes in the following which also is likely to match user expectations.

Substituting 20% of the pedestrian node with cars helps epidemic forwarding and PRoPHET, as these act as “messengers” and spread data easier. They do not help DD. SW and SWB are pretty much neutral except for the MBM model where up to 20% improvement were obtained with 12 message copies.

Looking at the parameters of the routing protocols, except when using RWP, PRoPHET performs best with $\beta = 0$; we assume that encounters between nodes are so frequent and circular that they distort transitive probability predictions. Therefore, we only consider PRoPHET with $\beta = 0$ (P0), effectively disabling the transitive predictions. Both normal and binary spray and wait perform best when only 6 copies of a message are used (except for RWP and MBM which are least realistic). Also, normal and binary spray and wait are largely (but not entirely) similar in their performance. SW seems to benefit from cars being around unless RWP or SPMBM are used. But the difference are rather small. Therefore, we use 6 copies for and consider both the normal (SW6) and the binary (SWB6) variants.

The above observations and particularly the impact of the mobility models on the protocol performance become apparent from figures 6, 7, and 8 which show the message delivery probability, latency, and the overhead per delivered message, respectively. We observe that all protocols perform better the more realistic the mobility models become (only DD does not achieve reduced latencies). Figure 8 shows that the overhead of heavily replicating protocols such as epidemic and PRoPHET grows with increasing reality whereas the overhead of spray and wait decreases. Generally, all protocols perform poorly under RWP, but also adding a map structure to random motion as with MBM only provides limited benefit.

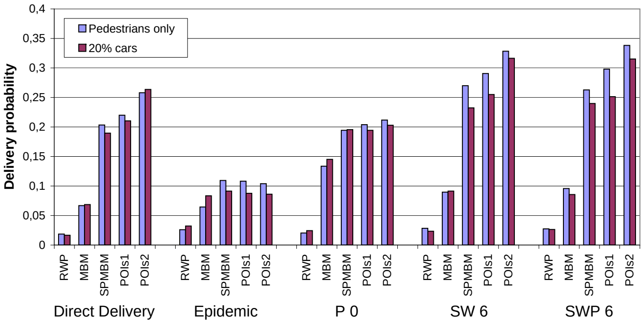


Fig. 6. Message delivery probability

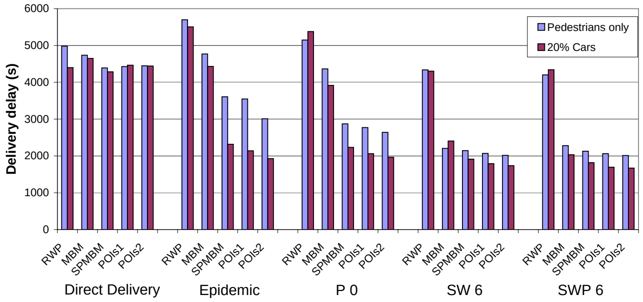


Fig. 7. Message delivery latency

Adding cars has a diverse effect on the performance. The above observations remain but details differ. Cars (and similarly trams) reduce the average delay as they can carry messages faster and thus increase connectivity. This is also reflected when looking at scatterplots of delivery delay over distance: messages from further away are delivered more frequently or in less time. However, with our large message

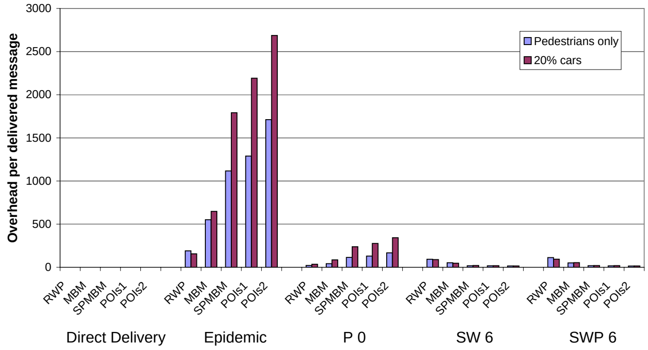


Fig. 8. Relative overhead per delivered message

sizes, the contact duration between a car and another node is often too short to communicate the message (we have indeed observed a bias towards the successful transmission of smaller messages). This changes is the node density is reduced as we observed in our simulations with 100 nodes: cars and trams contribute significantly to the overall connectivity, yielding an improvement also in the message delivery ratio. We also observe a positive trend (reduced latency, increased delivery ratio) for throwboxes, simply as additional relay nodes become available.

C. *Interactive DTN Messaging*

We finally looked at interactive messaging between pairs of nodes which is from an application perspective more meaningful than unidirectional communications. We model the exchange of short messages (Small-Small) to cover brief text messaging between users, small requests with large responses (Small-Large) to represent content queries (to users or web servers via gateways), and large requests with small responses (Large-Small) to model blog postings or image transfers with a quick reply or reception report. Small messages are 10–20 KB in size whereas large messages are 100 KB–2 MB, using a uniform distribution in both cases. We run these simulations with more extreme parameter sets (only 100 nodes, thus fewer messages, and TTL of 12 hours) to get more pronounced results for all mobility models and routing protocols but report the results only for SPMBM and SW6.

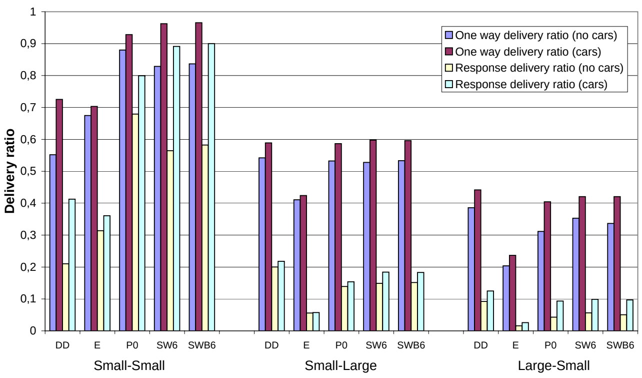


Fig. 9. Delivery ratio for request-response pairs

Figures 9 and 10 summarize the results. They confirm that smaller messages have much better chances of being delivery: with smaller (and fewer) messages generated, the buffer occupancy is lower and fewer messages are dropped (SW normal and binary did not drop any messages). Furthermore, the message exchange is quicker and thus these messages can particularly benefit from passing cars (or trams) as is clearly visible from both the delivery ratio and the latencies.[7]

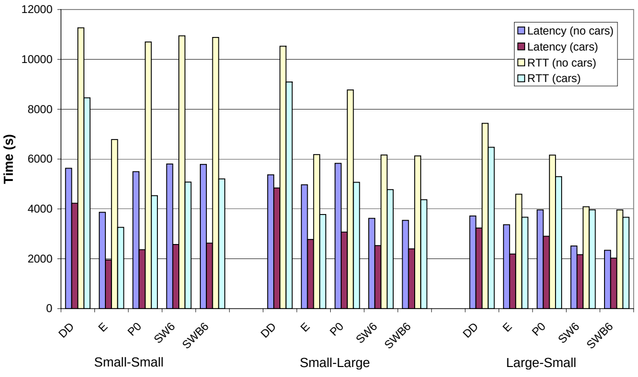


Fig. 10. Latency for request-response pairs

Looking at the latency and RTT in figure reffig.req-rsp-rtt emphasizes the importance of cars for efficient communication of small messages: they are able to reduce communication latencies by more than 50%. Overall, however, this scenario still demands some delay tolerance from the users having to wait about an hour for the response to a previously issued request. Yet, while further study is surely required, non-time-critical communications during the days appear well suitable given these results so far.

D. *Further Observations*

Beyond what we reported above, we found aspects of general nature worthwhile discussing. If the scenario includes nodes with high velocity, smaller messages have a higher probability of getting delivered than large messages. This can be seen from average delivered message sizes which are up to 20% smaller for scenarios with cars compared to scenarios with only pedestrian nodes. The effect is natural since smaller messages can utilize even the short connections that happen between fast and slow nodes and they also have a smaller change of being aborted due connection going down because their transfer time is shorter.

When comparing the results of message delivery probability using lower end and higher end devices, most of the routing protocol and mobility model combinations yield expected 50–350% increase in delivery probability with increased radio range and buffer size. However, with random waypoint the increase is within range of 400–2000%. This gives an indication that (overly) simple movement models may result in massively biased conclusions about the effect of simulation parameters.

Finally, we have observed that very heterogeneous buffer sizes can lead be problematic if contacts durations allow exchanges many messages. For example, a throwbox with a lot of storage may easily overwhelm a mobile phone user passing by, potentially causing instant message drops due to buffer overflows and/or future ones by filling up the buffer. Particular when using routing protocols that can potentially create an unlimited number of message copies, care must be taken when deciding which (and how many) messages to exchange.

---

[7] The average delivery ratios of *Large-Small* are less than *Small-Large* because the ratio is calculated based upon the total number of sent messages. If, however, a large request does not make it to the receiver, then no small response will be generated.

V. CONCLUSION

In this paper, we have presented ONE, an opportunistic networking evaluation system that offers a variety to tools to create complex mobility scenarios that come closer to reality than many synthetic mobility models. GPS Map data provides the scenario setting and node groups with numerous different parameters can be used to model a wide variety of independent node activities and capabilities. With its flexible input and output interfaces, ONE can incorporate real-world traces and feeds from other mobility generators as well as generate mobility traces for use by other simulators. Its extensible routing scheme currently includes four parameterizable DTN routing protocols. Its visualization component can be used for instant sanity checks, deeper inspection, or simply to observe node movements in real-time—which broadens its applicability beyond DTN studies.

While ONE is not complete at this point, it already supports gaining valuable insights into the behavior of DTN routing protocols when increasing the degree of reality. To this end, we have analyzed how contact patterns change when adding heterogeneity to a setting and we have observed how different DTN routing protocols react to such changes in the dynamics of their environment. This provides useful insights into DTN routing protocols which is important towards designing more robust and adaptive DTN routing protocols in the future—which is one of our next steps after incorporating more of the existing protocols into the simulator. In parallel, we continue to expand the functionality of ONE to include group mobility, vehicles with more than one person, heterogeneous radio links, and Internet access points and to further embrace real-world measurements and interface to large-scale simulation systems.

VI. ACKNOWLEDGMENTS

The work on ONE is funded by Nokia in the SINDTN project. The GPS map data has been kindly made available by Maanmittauslaitos as under the general TKK license.

REFERENCES

[1] Jörg Ott, Dirk Kutscher, and Christoph Dwertmann, "Integrating DTN and MANET Routing," in *ACM SIGCOMM Workshop on Challenged Networks (CHANTS)*, September 2006.

[2] Kevin Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," in *Proceedings of ACM SIGCOMM 2003*, August 2003, pp. 27–36.

[3] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H.Weiss, "Delay-Tolerant Network Architecture," Internet Draft draft-irtf-dtnrg-arch-05, Work in progress, March 2006.

[4] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[5] David R. Choffnes and Fabián E. Bustamante, "An Integrated Mobility and Traffic Model for Vehicular Wireless Networks," September 2005.

[6] "Homepage of the TranSims Transportation Analysis Simulation System," http://www.ccs.lanl.gov/transims/, 2007.

[7] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," in *ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN)*, 2005.

[8] Nathan Eagle and Alex (Sandy) Pentland, "Reality mining: sensing complex social systems," *Personal Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.

[9] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," in *Proceedings of IEEE Infocom*, April 2006.

[10] Wenrui Zhao, Yang Chen, Mostafa Ammar, Mark Corner, Brian Levine, and Ellen Zegura, "Capacity enhancement using throwboxes in dtns," in *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, October 2006.

[11] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 24–37, January 2006.

[12] L. Pelusi, Andrea Passarella, and Marco Conti, "Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks," *IEEE Comm. Magazine*, November 2006.

[13] Sushant Jain, Kevin Fall, and Rabin Patra, "Routing in a Delay Tolerant Network," in *Proceedings of the ACM SIGCOMM*, 2004.

[14] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Duke University, April 2000.

[15] Anders Lindgren, Avri Doria, and Olov Schelen, "Probabilistic routing in intermittently connected networks," in *The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR)*, 2004.

[16] Jeremy Leguay, Timur Friedman, and Vania Conan, "Evaluating Mobility Pattern Space Routing for DTNs," in *Proceedings of IEEE Infocom 2006, Barcelona, Spain*, April 2006.

[17] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *Proc. Sensor and Ad Hoc Communications and Networks SECON*, Oct. 2004, pp. 235–244.

[18] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *Proceedings of ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN)*, 2005.

[19] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-copy Case," *To appear in ACM/IEEE Transactions on Networking*, 2007.

[20] "Homepage of the Community Resource for Archiving Wireless Data At Dartmouth (CRAWDAD)," http://crawdad.cs.dartmouth.edu/, 2007.

[21] Gustavo Marfia, Giovanni Pau, Enzo De Sena, Eugenio Giordano, and Mario Gerla, "Evaluating Vehicle Network Strategies for Downtown Portland: Opportunistic Infrastructure and the Importance of Realistic Mobility Models," June 2007.

[22] David B Johnson and David A Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, Imielinski and Korth, Eds., vol. 353. Kluwer Academic Publishers, 1996.

[23] "OpenJUMP - The free, Java based and open source Geographic Information System for the World," http://openjump.org, 2007.

[24] "Homepage of the DTNSim2 Delay-tolerant Network Simulator," http://watwire.uwaterloo.ca/DTN/sim/, 2007.

[25] Anders Lindgren and Avri Doria, "Probabilistic Routing Protocol for Intermittently Connected Networks," Internet Draft draft- lindgren-dtnrg-prophet-02, Work in Progress, March 2006.

[26] "Graphviz - Graph Visualization Software," "http://www.graphviz.org/", 2007.

[27] "Gnuplot - An Interactive Plotting Program," http://gnuplot.info/, 2007.

[28] Keith Scott and Scott Burleigh, "Bundle Protocol Specification," Internet Draft draft-irtf-dtnrg-bundle-spec-08, Work in progress, December 2006.