Delay-tolerant On-demand Routing for Mobile Ad Hoc Networks

Christian Kretschmer, Stefan Rührup, Christian Schindelhauer
{kretschm, ruehrup, schindel}@informatik.uni-freiburg.de

Computer Science Technical Report No. 244
University of Freiburg, Germany
December 2008

Multihop routing is a cornerstone of communication in mobile ad hoc networks and intermittently connected networks. While ad hoc routing protocols target the connection establishment in a changing environment, communication in intermittently connected networks requires end-to-end delivery without established connections. We present DT-DYMO, a combination of ad hoc routing based on the established Dynamic MANET On-demand Routing protocol and mechanisms for in-network storage and node availability prediction. DT-DYMO uses a route discovery mechanism in order to find the destination or nodes which are likely to be able to deliver the message in the future. We present simulation results for highly mobile wireless networks with temporary network disconnection. The results show that the extensions in DT-DYMO can significantly increase the delivery rate in mobile scenarios and thus can overcome the limitations of traditional ad hoc routing, while providing faster delivery than forwarding schemes that rely only on node availability estimation.

1 Introduction

Mobile Ad hoc Networking and Delay-tolerant Networking are fields where challenging network environments are considered, either because of node mobility, unstable links and the lack of infrastructure, or because of temporary disconnection and high latency. In those networks the routing protocol provides the basis of network communication and makes end-to-end message transmission over multiple hops possible. In ad hoc networks, the network dynamics requires the routing protocol to either update routing tables proactively in order to react on the changing topology or to search for the destination on-demand. Dynamic MANET On-demand Routing (DYMO) [2] is one of the recent developments in ad hoc routing, which uses a reactive route discovery: When a packet has to be transmitted, the source floods the network with route requests, and the destination replies using the reverse path. During this process, a route is established, which is then used for the data transmission. Reactive ad hoc routing protocols such as DYMO can cope with mobility and network dynamics up to a certain extent, but if the network is only intermittently connected, the route establishment fails.

Scenarios with network disruption and high latency are considered in delay-tolerant networking [7]. A delay- or disruption-tolerant routing protocol should ensure message delivery even in cases where no continuous end-to-end route can be established. An often-mentioned example is a shuttle bus commuting between two villages which are cut off from the Internet and from the telephone network. If the bus is equipped with a WLAN router and a storage device, it can provide a message transport service between the two villages, though an end-to-end connection cannot be established. This scenario shows that temporary network disconnection does not imply that multi-hop routing is not possible, it means that routing simply adheres to another principle. Delay-tolerant routing follows the store-carry-forward principle [6] where nodes keep messages while moving around and trying to find a better relay to the destination. If nodes only meet sporadically and do not know their future meeting points, other routing techniques than the traditional shortest path selection have to be applied, e.g. the dissemination of multiple message copies in the network [17] or the estimation of the likelihood of being able to deliver the message in the future [11, 1].

DT-DYMO combines these two worlds: It provides fast ad hoc routing while being tolerant to delays and network disruption. In case of network disruption, it delivers the message to a suitable *message carrier*, which takes over the message for later delivery. Message carriers are selected by predicting the delivery likelihood based on past meeting points. By integrating store-carry-forward principles into ad hoc routing, DT-DYMO can provide fast delivery in highly mobile ad hoc networks and also increase the delivery rate in sparsely connected networks. It is also suitable for heterogeneous networks with mobile nodes and static mesh nodes and can thus provide convergence between mesh and ad hoc networks. DT-DYMO appears to be a best-of-two-worlds approach, but interestingly there seems to be a small trade-off between successful delivery and latency when combining multi-hop ad hoc routing and store-carry-forward routing in mobile networks. Our simulation results give further insight in this issue.

This paper is organized as follows: In the following section, we review related work. In Section 3 we describe the extensions to DYMO and how they were integrated into the new DT-DYMO protocol. Section 4 contains the simulative study with the discussion of the results. Finally, we give a conclusion in Section 5.

2 Related Work

2.1 Routing in Mobile Ad-hoc Networks

Routing protocols for Mobile Ad-hoc Networks enable multi-hop communication in networks with changing topologies. A changing topology requires to gather routing information that reflects the current state of the network. Proactive protocols such as DSDV [15] maintain routing tables by exchanging update messages on a regular basis. Reactive protocols such as AODV [14] and its successor DYMO [2] search for a routing path on demand.

As an example, the route discovery of DYMO works as follows: First, the source node starts to flood a route request (RREQ) through the network. Each relaying node appends its information to this message, such that all intermediate nodes can learn the route to the source. Once the RREQ reaches the destination, it responds with a route reply (RREP), which is sent via unicast on the reverse path back to the source. Again, the intermediate nodes append their information to the RREP and record the route to the destination. Once the RREP reaches the source, each intermediate node knows how to forward the data packet to the destination.

Besides proactive and reactive algorithms that establish topology information or use a route discovery, there are geographic routing protocols [9] which make routing decisions based on location information, provided that nodes are equipped with positioning capabilities. We do not use position information here.

2.2 Delay-tolerant Networking

In mobile ad hoc networks it is often assumed that the network remains connected during message delivery. As this is not always the case, algorithms have been developed that can cope with network disruption and long delays and do not need a continuous end-to-end path (see [13, 19] for an overview). Within the Internet Research Task Force, the Delay Tolerant Networking Research Group[1] has developed the DTN Bundle protocol [6], which forms an overlay network connecting DTN-capable nodes, so-called *endpoints*. Data is transferred from endpoint to endpoint using traditional transport and network protocols such as TCP/IP. If an endpoint is not available, the data packets (bundles) are stored until the link or the route to the next endpoint becomes available. The Bundle protocol specification [16] does not define an algorithm for routing over multiple endpoints, but provides the framework for bundle forwarding in a *store-carry-forward* style, including fragmentation, storage, naming and addressing. The Bundle protocol can be set up to use static routes or extended by a dynamic routing algorithm, e.g. by DTLSR [4], a link state routing that uses an estimated delay as metric.

As the future link availability cannot always be predicted, it is also possible to spread messages in the network to increase the chances of successful delivery when a route becomes available. In order to avoid flooding the network, randomized algorithms for information dissemination were developed [3, 8]. Later, epidemic routing was proposed as routing algorithm for intermittently connected ad hoc networks [17]. Epidemic routing is a per-contact forwarding, where nodes hand over packets to other nodes they encounter. Based on the idea that the mobility in real networks is not completely random, PRoPHET [11] and MaxProp [1] use an estimation of the delivery probability to select next hop nodes. We use PRoPHET's mechanism to calculate delivery probabilities. It is based on the following idea: Whenever two nodes meet, they increase the encounter probability or delivery probability for each other. They also exchange their probability vectors for other nodes and update their own vectors. The update after encounter increases the probability values; they are decreased over time. The exact mechanism is explained in Section 3.3.

Estimating delivery probabilities helps in finding suitable relay nodes or message carriers. It is also important to decide how many message carriers are selected and how long a message

[1] www.dtnrg.org

should be kept. The more nodes are involved in packet dissemination, the higher the delivery probability, but the larger the required buffer capacity. If this capacity is limited, situations can occur where new packets have to be rejected or buffered ones have to be discarded. MaxProp uses a buffer management that gives buffered packets priorities depending on their delivery likelihood. A similar mechanism is proposed for PRoPHET, where the least probable messages are dropped first.

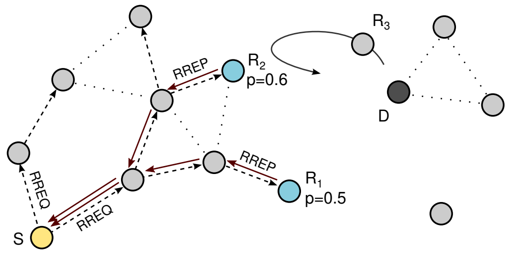
Similar to our work, Ott et al. [12] combine ad hoc and delay-tolerant routing. They describe concepts and the architecture of the integration of AODV and the DTN bundle protocol, thus making DTN-based communication available to ad hoc networks. Their work is based on the DTN Bundle protocol reference implementation and uses AODV as an additional routing protocol. AODV determines the availability of an endpoint, discovers other DTN-capable nodes within the route discovery phase, and provides endpoint-to-endpoint communication within the ad hoc network. This also gives the application layer the option to choose between end-to-end and DTN-based communication. We do not use the DTN framework, but integrate delay tolerance mechanisms directly into the ad hoc routing protocol. This is less flexible from an application point of view, but avoids using the TCP/IP connection over each wireless link in the ad hoc network. Furthermore, we focus on performance issues instead of an architectural framework.

3 Delay-tolerant routing extensions in DT-DYMO

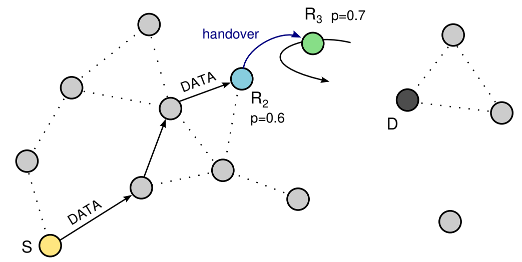
DT-DYMO is based on the reactive routing mechanisms of the DYMO protocol and contains additional mechanisms in order to become delay- or disruption-tolerant. During DYMO's route discovery, route requests are only answered by the destination node. In situations, where the destination node is not available, but another node close to or with frequent contact to the destination, the message can be forwarded to this node. It will serve as a *message carrier* and deliver the message to the destination later. DT-DYMO follows this strategy and uses message carriers, if the destination is unavailable. After taking over a message, the message carrier uses point-to-point handovers based on delivery probabilities to pass the message to a more suitable message carrier or to the destination. Figure 1 illustrates this example.

3.1 Routing modes and message handling

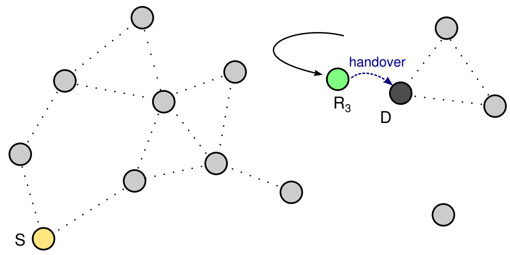
DT-DYMO supports two forwarding modes: *multi-hop* forwarding (based on DYMO's route discovery) and *store-carry-forward* using point-to-point handovers based on delivery likelihood. Routing messages are handled as proposed in the DYMO specification, with the following exceptions: Route requests are answered, if a node qualifies as a message carrier, i.e. it has a high delivery probability w.r.t. the target. In turn, multiple replies arrive at the source, out of which the most suitable message carrier is chosen. If a message carrier receives a DATA packet, it switches the packet to store-carry-forward mode and stores it in its queue for later delivery. DT-DYMO's routing messages contain additional fields for message carrier selection (see Table 1). The original fields remain such that DT-DYMO routing packets can be handled like original DYMO packets by relay nodes.



(a) Source node S issues a route request with minimum probability 0.5, which is answered by R_1 and R_2.



(b) S selects R_2 as message carrier and sends the data packet. Then R_2 meets R_3, which has a better probability, and hands over the packet



(c) R_3 carries the message, meets D and is finally able to deliver the message.

Figure 1: Routing example for DT-DYMO

3.2 Message Carrier Discovery

In order to select a suitable message carrier, we specify a minimum delivery probability in the route request for a potential message carrier. All nodes whose delivery probability for the destination exceeds this value answer with a route reply. If such node is not the destination itself, it forwards the message again. It is also possible to stop forwarding RREQs once a node is eligible for carrying the message – this reduces the total message overhead of the route discovery, but increases the delay. This is because the replying nodes may prevent the RREQ to be forwarded to more suitable nodes or to the destination itself. Therefore we do not suppress the RREQ forwarding.

After the route replies reach the source, a suitable message carrier is selected to which the data packet is sent afterwards. Here, the source has the following choices: It can select the best node with the highest delivery probability for the destination, a random message carrier, or distribute the message to multiple nodes.

If the route request was not successful, i.e. if the source did not receive any reply, it switches the packet to store-carry-forward mode and becomes a message carrier by itself. It will store the message in its queue until it meets another node with a higher probability for delivering this message. As an option, it can start the route request again, which is also an option in the DYMO specification. In order to increase the chances of finding a message carrier, the minimum carrier probability is decreased in each retry until the retry limit is reached or the probability is so low that a handover to a neighboring node is preferred.

RREQ	
HopLimit	
TargetNode.Address	destination node
OrigNode.Address	sender of the RREQ
OrigNode.SeqNum	
minDeliveryProb*	delivery prob. threshold
AdditionalNodes	list of intermediate nodes

*) DT-DYMO extension fields

RREP	
HopLimit	
TargetNode.Address	sender of the RREQ
OrigNode.Address	destination or msg. carrier
OrigNode.SeqNum	
SearchedNode*	actual destination
deliveryProb*	originator's probability for the destination
AdditionalNodes	list of intermediate nodes

Table 1: Content of DT-DYMO Routing Messages (without IP/UDP header).

Depending on the message carrier selection and the routing mode, we use the following variants in the simulations:

- *Classic*: The message carrier with the highest delivery likelihood is selected. After an unsuccessful route request, the message is switched to store-carry-forward mode.
- *Fast-or-never*: If a route request is unsuccessful, the packet is dropped and the application layer informed. This can be used, if the application requires a minimum delivery probability and an immediate response after an unsuccessful route request.
- *5 message carriers*: The source selects up to 5 message carriers starting with the best probability. This implies duplication of the data message.
- *Random message carrier*: The source selects a random message carrier among the nodes that sent a route reply.

3.3 Estimation of delivery probabilities

If the destination is temporarily unreachable, message carriers are selected based on their likelihood to deliver the message in the future. The prediction of these probabilities is based on historic data about meeting with other nodes. PRoPHET [11] and MaxProp [1] use such mechanism, and DT-DYMO is not limited to any specific method to calculate delivery probabilities.

For the implementation, we have chosen PRoPHET's scheme for calculating delivery probabilities, which is described in the following. Whenever a node A meets node B, it adjusts its delivery probability for B as follows.

$$p(A,B) = p(A,B)_{\text{old}} + (1 - p(A,B)_{\text{old}}) \cdot P_{\text{init}}$$

where $P_{\text{init}} \in [0..1]$ is a fixed parameter. A also receives B's probability vector and adjusts its probabilities to reach other nodes. A's probability for a node C from B's probability vector is then computed by:

$$p(A,C) = p(A,C)_{\text{old}} + (1 - p(A,C)_{\text{old}}) \cdot p(A,B) \cdot p(B,C) \cdot \beta$$

where β is a fixed parameter to weight the *transitivity*. The probabilities are reduced over time, which is called *ageing*. Whenever a probability is accessed, it is updated using $p_{(A,B)} = p_{(A,B)_{\text{old}}} \cdot \gamma^k$, where γ is the aging parameter and k the number of time steps since the last aging.

The probability vectors are updated, whenever two nodes meet, i.e. whenever they are within mutual communication range and receive beacons from the other node. Other than specified in the PRoPHET protocol [10], we let the nodes include their probability vectors in the beacons and broadcast them on a regular basis, instead of a node-to-node exchange in addition to the beaconing. This reduces the communication overhead in cases where a node joins a group of other nodes, because it does not need to communicate via unicast with all new neighbors. However, the updates by beacons are done asynchronously and can lead to fluctuating probabilities. In order to avoid packets to bounce between two nodes because of fluctuations, a packet handover is performed only if the receiver's delivery probability for the destination exceeds the one of the node holding the packet by a *handover threshold*.

4 Simulation Results

We performed simulations using the simulator OMNeT++ with the INET framework [2]. The implementation of DT-DYMO is based on the DYMO implementation for OMNeT++ in [5]. We use an IEEE 802.11 Mac Layer in ad hoc mode, and OMNeT's free space propagation model on the physical layer. The application layer generates messages to random destination nodes. As we focus on intermittently connected networks without continuous end-to-end paths, we do not use an additional transport layer.

The playground size is 1200m $\times$ 850m. In order to create sparse and dense networks, we used different transmission ranges depending on the scenario. In the simulation results we specify the *average network density* as the number of neighbors a node has on average during the simulation. Beacons are generated within an interval of 30 seconds. Data packets are generated every 30 seconds after an initial phase of 10 min. Data is collected over 10 simulation runs.

We compare DT-DYMO with the original DYMO, and the PRoPHET protocol, from which the calculation of delivery probability is borrowed. For a fair comparison, DYMO uses the same number of route request retries. PRoPHET is started with the same parameters for the calculation of probability vectors and uses the simplified probability vector exchange of DT-DYMO.

4.1 Mobile Ad hoc Networking Scenario

In this scenario all the nodes are placed randomly and start moving continuously according to the random waypoint model without pause time. The speed is always non-zero as recommended in [18]. The parameters are given in Table 2.

Figure 2 shows the delivery rate for different network densities. In sparse networks (density ≤ 5), where DYMO can deliver less than 15% of the messages, DT-DYMO and PRoPHET have a delivery rate of around 70% and more. With increasing density, DYMO and DT-DYMO reach a delivery rate of 85%, PRoPHET's rate is even higher by up to 10%. In the simulations we could detect packet losses on multi-hop routes due to high mobility. These link errors cannot

[2] www.omnet.org

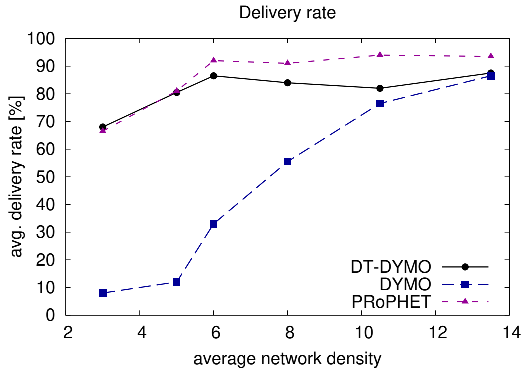


Figure 2: Delivery rate comparison for different network densities.

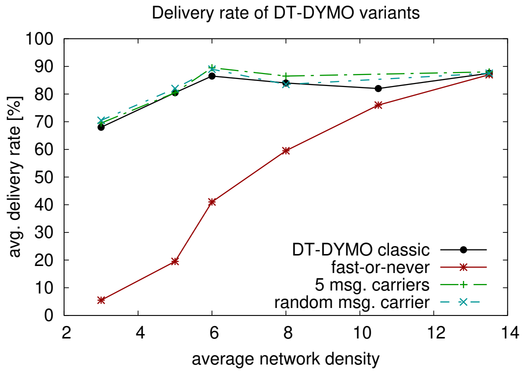


Figure 3: Delivery rate of DT-DYMO variants for different network densities.

simulation time	25 min
max. pkt. lifetime	15 min
mobility model	random wp.
speed (min/max)	6–7m/s
pause time	0
number of nodes	20
network density	variable: 3 – 13.5 (avg.)

p_{init}	0.8
γ (aging)	0.7
β (transitivity)	0.3
handover threshold	0.05
min. carrier probability	0.2
max. carrying time	15 min

Table 2: Parameters for the mobile ad hoc network scenario

be detected fast enough, and the source node has no chance to react after route discovery. This problem cannot be solved by simply adding a transport layer retransmission strategy based on end-to-end acknowledgements, as network disruption might prevent their delivery. A better alternative are negative acknowledgements after a retransmission timeout on the MAC layer. In contrast to multi-hop routing, the point-to-point forwarding used in PRoPHET is more reliable, which explains the higher delivery rates. But as the delivery probabilities do not always reflect the shortest path, PRoPHET's message delivery usually takes longer. This is visible in the histograms for end-to-end delay in Figure 4.

In sparse networks, DYMO delivers around 5% of the messages very fast, but most of the messages cannot be delivered. PRoPHET can deliver nearly 70% of all messages, but much slower than DYMO. DT-DYMO delivers a few messages very fast and most of the messages a little earlier than PRoPHET. In dense networks, both DT-DYMO and DYMO prove the advantage of reactive routing. Over 80% of the messages are delivered within the first 0.5 sec. PRoPHET delivers the majority of the messages slower within 5 – 50 sec. Hence, DT-DYMO is as fast as DYMO in dense networks and even slightly faster than PRoPHET in sparse networks, without loosing as many messages as DYMO.

Furthermore, DT-DYMO is able to route messages without having accurate delivery probabilities for distant nodes. In our simulations, we allow the probabilitiy vectors to be established in the initialization phase, so that PRoPHET can benefit from that. For DT-DYMO being successful it is only necessary to find a node in vicinity of the destination with accurate values.

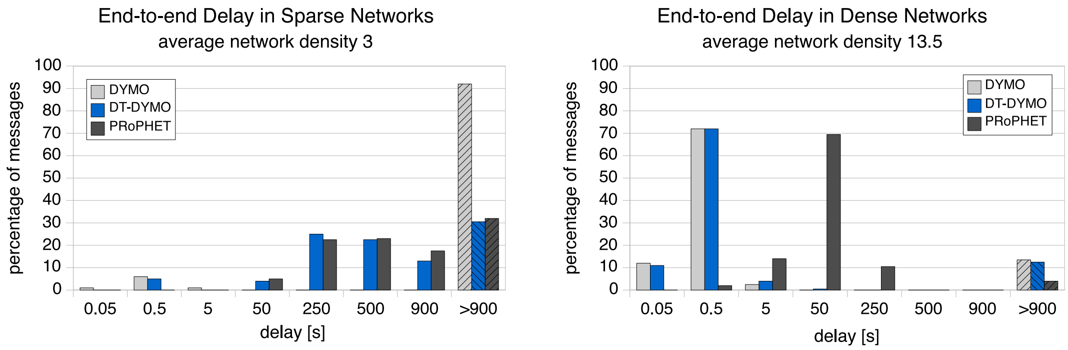


Figure 4: End-to-end delay histogram for DYMO, DT-DYMO and PRoPHET in sparse and dense mobile networks. The bars show the fraction of messages that could be delivered within the denoted time. Messages exceeding the lifetime limit (>900) are never delivered.

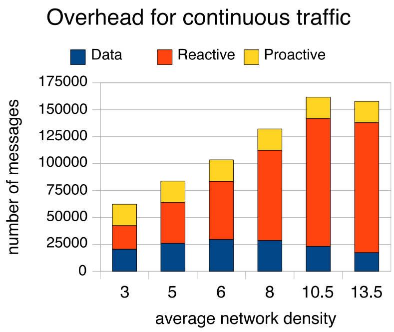


Figure 5: Overhead of DT-DYMO for proactive (beacons), reactive (RREQ/RREP) and data messages.

Regarding the variants of DT-DYMO (Figure 3), there is no significant difference between different choices of the message carrier (best, 5 or random). The fast-or-never delivery strategy works similar to the original DYMO and performs similar in sparse networks, where unsuccessful route requests occur often and cause the deletion of the message.

The message overhead is shown in Figure 5 for continuous packet generation. In comparison to the reactive route discovery, the proactive beaconing, which is added in DT-DYMO to exchange delivery probabilities, does not create a large overhead. Of course, the picture is different if only a few data packets are routed. The major part of the traffic is due to reactive route requests and replies, and this part increases with higher network density, because larger parts of the network are accessible and affected by propagating the route request.

4.2 Mesh Network Scenario

In this scenario the network consists of static mesh routers and mobile nodes, using an 802.11 MAC layer and DT-DYMO as routing protocol. The static nodes form two clusters, which are spatially separated and not connected (see Figure 6). The mobile nodes are commuting between

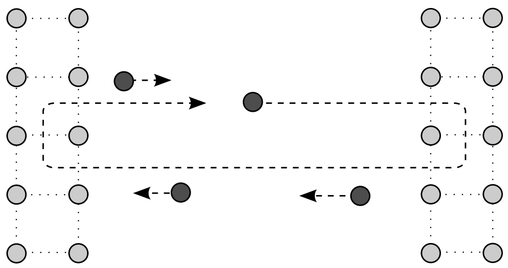


Figure 6: Mesh network scenario. Mobile nodes (dark) are carrying messages between two clusters of static mesh nodes.

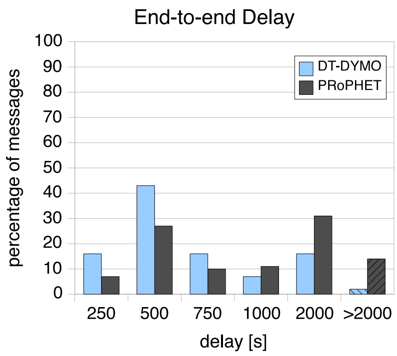


Figure 7: End-to-end delay histogram for DT-DYMO and PRoPHET in the mesh network scenario. DYMO cannot deliver any message in this scenario.

the two clusters on rectangular trajectories. The static nodes in the clusters are placed at a distance of 100 m while the transmission range is about 125 m. Messages are sent from the left cluster to random nodes in the right cluster. Simulation parameters are given in Table 3.

simulation time	60 min	p_{init}	0.8
max. pkt. lifetime	15 min	γ (aging)	0.75
mobility model	static & rect.	β (transitivity)	0.3
speed (min/max)	6–7m/s	handover threshold	0.05
pause time	0	min. carrier probability	0.1
number of nodes	30	max. carrying time	15 min

Table 3: Parameters for the mesh network scenario

A scenario like this is exemplary for delay tolerant networking. The mobile nodes are able to carry messages, but ad hoc routing without in-network storage is not possible, except for the unlikely case that the mobile nodes form a chain bridging the gap between the two clusters. Therefore, DYMO could not deliver any message. By contrast, DT-DYMO and PRoPHET can deliver most of the messages, as Figure 6 shows, although there is a significant delay. The problem in this scenario is to pass the packet to a suitable message carrier. But the nodes coming from the destination direction often have a higher delivery probability than those going towards the destination, which can cause a detour. Nevertheless, DT-DYMO benefits from ad-hoc routing among the static mesh nodes and thus can deliver messages faster than PRoPHET. Though the static mesh routers could be also connected by an adaptive link state routing protocol, this example shows, how reactive ad hoc routing combined with store-carry-forward routing can provide connectivity in a heterogeneous network.

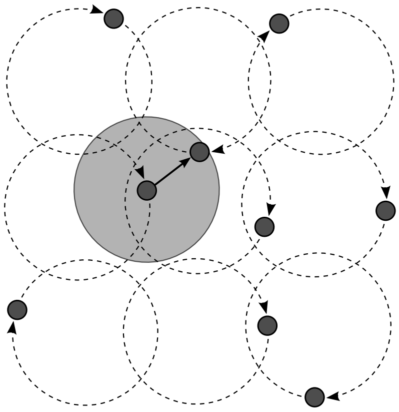


Figure 8: Local mobility scenario. Nodes are connected occasionally while moving on circular trajectories.

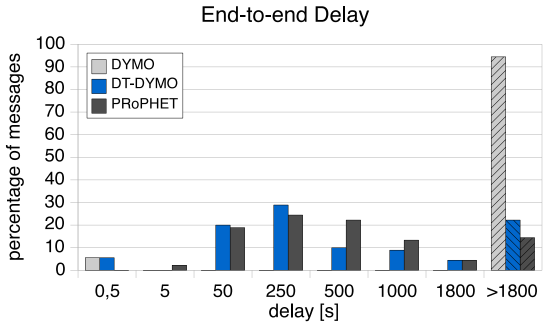


Figure 9: End-to-end delay histogram for DYMO, DT-DYMO and PRoPHET in the local mobility scenario.

4.3 Local Mobility Scenario

In this scenario, we let 9 nodes move on circular trajectories, which are aligned in a 3×3 placement as shown in Figure 8. The transmission range is chosen such that communication links can be established only sporadically. However, the nodes meet with the same neighbors again which gives a better basis for estimating delivery probabilities than in case of random movement. Scenarios of this kind are the domain of protocols such as PRoPHET, which rely on delivery probability estimation.

simulation time	60 min
max. pkt. lifetime	30 min
mobility model	static & rect.
speed (min/max)	6–7m/s
pause time	0
number of nodes	9

p_{init}	0.8
γ (aging)	0.8
β (transitivity)	0.6
handover threshold	0.01
min. carrier probability	0.1
max. carrying time	30 min

Table 4: Parameters for the local mobility scenario

The end-to-end delay histogram in Figure 9 shows that DYMO could deliver some messages in a few cases where an end-to-end connectivity was possible, but most of the messages got lost. DT-DYMO could also deliver these messages very fast. Otherwise the performance is similar to PRoPHET with a slight advantage on delivery times and a few more messages that got lost. Sometimes a very long delivery time can be observed though the network is small and movement pattern seems to be simple. However, the delivery probabilities are adapted when nodes meet, and the random connectivity between the nodes causes fluctuations in the probability vectors leading to disadvantageous next hop selections and packets being forwarded hop by hop without progress.

5 Conclusion

With DT-DYMO we have combined on-demand ad hoc routing with extensions for enabling message delivery in highly mobile and intermittently connected ad hoc networks. DT-DYMO has the advantage, that a message is forwarded as far and as fast as possible by multi-hop routing, while not only looking for the destination but also for suitable message carriers. If the destination is reachable, the message can be immediately delivered and acknowledged. If this is not possible, forwarding continues using nodes, which probably encounter the destination node in the future. The extensions are carefully integrated, such that the discovery of message carriers also works in heterogeneous networks with nodes running the original DYMO protocol.

In most cases DT-DYMO can deliver messages much faster than a forwarding scheme that is based on delivery probability prediction. In highly mobile and sporadically connected networks it yields a significantly higher delivery rate than ad hoc routing by the original DYMO protocol. Moreover, the storage requirements are low compared to epidemic routing schemes, which disseminate multiple copies of the data packet in the network.

However, despite the good performance, DT-DYMO could not solve all problems that occur in highly dynamic networks. The fast ad hoc routing over multiple hops is more prone to link failures than store-carry-forward routing from point to point. But relying solely on store-carry-forward routing increases the latency significantly. Moreover, delivery likelihood prediction does not come for free: a beaconing mechanism is necessary to gather the required information, and the quality of the estimation depends on the beaconing frequency. However, DT-DYMO uses the reactive route discovery first and is thus able to forward the message in vicinity of the destination where the delivery probability information is more up-to-date.

The delivery likelihood prediction mechanism of PRoPHET, which we have selected for our implementation, is controlled by parameters for initial probabilities and ageing. The quality of the prediction depends on how well these parameters reflect the dynamics of network, a factor that is usually not known in advance. Futher improvements could be made by designing an adaptive mechanism that reacts on network dynamics.

Finally, lost messages on multihop paths have can be recovered as long as an end-to-end connection exists. Otherwise, the source could be informed by a negative acknowledgement about the delivery failure. For these cases, an appropriate transport layer mechanism reacting on message loss could complement the routing protocol.

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] Ian Chakeres and Charles Perkins. Dynamic MANET On-demand (DYMO) Routing. IETF Internet Draft, draft-ietf-manet-dymo-14, June 2008.

[3] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *6th Annual ACM Symposium on Principles of Distributed Computing (PODC '87)*, pages 1–12, 1987.

[4] Michael Demmer and Kevin Fall. DTLSR: delay tolerant routing for developing regions. In *NSDR '07: Proceedings of the 2007 workshop on Networked systems for developing regions*, pages 1–6, 2007.

[5] Isabel Dietrich, Christoph Sommer, and Falko Dressler. Simulating DYMO in OMNeT++. Technical Report 01/07, University of Erlangen, Dept. of Computer Science 7, April 2007.

[6] Kevin Fall and Steven Farrell. DTN: An architectural retrospective. *IEEE Journal on Selected Areas in Communications*, 25(5):828–836, June 2008.

[7] Stephen Farrell, Vinny Cahill, Dermot Geraghty, Ivor Humphreys, and Paul McDonald. When TCP breaks: Delay- and disruption-tolerant networking. *IEEE Internet Computing*, 10(4):72–78, 2006.

[8] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Randomized broadcast in networks. *Random Structures and Algorithms*, 1(4):447–460, 1990.

[9] S. Giordano and I. Stojmenovic. Position based routing algorithms for ad hoc networks: A taxonomy. In C. Xiuzhen, X.H. Huang, D.-Z. Du, and X. Cheng, editors, *Ad Hoc Wireless Networking*, pages 103–136. Kluwer, 2004.

[10] Anders Lindgren and Avri Doria. Probabilistic routing protocol for intermittently connected networks, draft-lindgren-dtnrg-prophet-03. IETF Internet-Draft, November 2007.

[11] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.

[12] Jörg Ott, Dirk Kutscher, and Christoph Dwertmann. Integrating DTN and MANET routing. In *CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks*, pages 221–228, 2006.

[13] Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine*, 44(11):134–141, November 2006.

[14] Charles Perkins, Elisabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (AODV) routing. IETF RFC 3561, July 2003.

[15] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244, 1994.

[16] K. Scott and S. Burleigh. Bundle protocol specification. RFC 5050, draft-irtf-dtnrg-bundle-spec, November 2007.

[17] Amin Vahdat and David Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report CS-2000-06, Duke University, July 2000.

[18] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Random waypoint considered harmful. In *22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03)*, 2003.

[19] Zhensheng Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *IEEE Communications Surveys & Tutorials*, 8(1):24–37, 2006.