Published in: 20. ITG-Fachtagung Mobilkommunikation, ITG-Fachbericht, VDE-Verlag, Berlin, 2015

# Data-Driven Routing for Delay-Tolerant Networks

Frank Nordemann*, Thorben Iggena*, Ralf Tönjes*
*Faculty of Engineering and Computer Science, University of Applied Sciences Osnabrück, Osnabrück, Germany

***Abstract*—For Delay-Tolerant Networks (DTNs) many routing algorithms have been suggested. However, their performance depends heavily on the applied scenario. Especially heterogeneous scenarios featuring known and unknown node movements as well as different kinds of data lead to either poor delivery ratios or exhausted network resources.**

**To overcome these problems this paper introduces *Data-Driven Routing for DTNs*. Data is categorized according to its requirements into priority queues. Each queue applies an appropriate DTN routing algorithm that fits the data requirements best. Simulation results show that Data-Driven Routing allows high delivery ratios for time-critical data while saving network resources during the transfer of less time-critical data at the same time.**

***Index Terms*—Delay-Tolerant Networks (DTNs), Routing, Forwarding, Epidemic, Spray&Wait, Prophet**

## I. Introduction

The realization of communication in delay-tolerant scenarios requires routing algorithms specialized for discontinued communication paths and unknown node movement. Many routing algorithms focus on fast data delivery by utilizing every communication possibility available. As a consequence, the network load grows extensively due to large amounts of created and transfered data packet copies. Other algorithms aim at network resource saving and may lead to significant delivery latencies.

Most real-world DTN scenarios include a broad range of data. Some data might be *highly time-critical* but relatively small in terms of data size (e.g. error data), other data might be larger but *less time-critical* (e.g. log data). Looking at DTN routing / forwarding algorithms, this results in a trade off between *fast delivery* and *network resource saving*. Especially scenarios including highly mobile nodes with unknown movement behaviors are challenging. Currently, there is no possibility to achieve fast delivery for time-critical data while using a resource saving delivery for non-critical data at the same time.

This paper presents a novel concept of routing data in DTN scenarios, named *Data-Driven Routing*. The concept is able to deliver data quickly if needed and to save resources if the transfered data is not time-critical. This is realized by applying class-based prioritization of data packets and well-known DTN routing algorithms.

The approach is motivated by requirements of different real-world use cases. In many outdoor scenarios communication between cooperating machines is needed for work process optimization. However, often neither mobile communication nor ad hoc communication connceting all nodes is possible. Examples are agriculture, construction work and surface mining. In particular, the presented work aims at enabling communication between agricultural machines for the optimization of harvesting logistics in rural areas. The work is performed in the German national project KOMOBAR [1] together with agricultural machine manufacturers. Organizing and controlling a grain or corn harvest requires data about the current status of harvesters, tractors and trailers. Furthermore, data on machine faults has to be communicated as fast as possible to reduce downtime of broken harvesters.

The reminder of the paper is organized as follows: First, related work in the areas of data prioritization and DTNs is presented. Section III introduces the concept of Data-Driven Routing, followed by an evaluation using a grain harvest scenario in Section IV. The paper concludes summarizing results in Section V.

## II. Related Work

The following section summarizes related work relevant for Data-Driven Routing.

### A. Prioritizing Data

Ensuring the delivery of important, time-critical data ahead of less-critical or non-critical data requires mechanisms to prioritize data against each other. Different techniques for prioritization of data packets exist, mostly based on the *Class of Service* (CoS) approach: Data packets are categorized into a number of classes according to the requirements of the contained data [2]. When sending data to other nodes, one class / queue is handled after the other following a defined order of priority. *Weighted-Fair-Queuing* or *Priority Jumps* may be applied if needed. It is important to remember that no guarantee for channel characteristics, delivery latencies or bandwidth can be given in delay-tolerant scenarios.

### B. Delay-Tolerant Networking

Delay-Tolerant Networking describes the realization of communication in intermittent, discontinued and dynamic environments [3]. Data is saved temporally on intermediate nodes and forwarded to newly encountered nodes, following the *store-carry-forward principle* [4]. The *Bundle Protocol* [5] is the reference protocol for the realization of DTNs. It inserts a Bundle Layer between application and communication layers to implement hop-to-hop communication.

### C. DTN Routing Algorithms

A key issue in delay-tolerant routing is the lack of knowledge about the future structure of the network. Nodes have no or just little data about future node contacts and communication opportunities. Depending on the level of knowledge, DTN routing is categorized into *deterministic* or *stochastic routing* [6] [7]. This paper addresses stochastic scenarios including highly mobile, dynamic nodes and time-restricted node movement repetitions.

*Epidemic Routing* [8] is a pragmatic approach of transferring data through a DTN. Since no data about communication paths is available, data packets are transfered to every node encountered in the network. Correspondingly, the network is flooded with packet copies which spread quickly across the network. Epidemic Routing may lead to congestion situations depending on the network load.

*Spray&Wait Routing* [9] is an optimization of Epidemic Routing and tries to minimize the amount of created packet copies. The algorithm features a configuration value $n$ specifying the maximum number of copies allowed for every data packet. The source of a data transfer is allowed to create packet copies. Alternatively, in *Binary Spray&Wait* the source shares the number of packet copies with encountered nodes till no copies are left.

*Prophet Routing* [10] introduces encounter probability values for the destination of a data transfer. Every time a node is encountered, the probability for this node is increased. Probability values are exchanged prior to the actual data transfer. If the probability for the destination is higher at the newly encountered node, packets are copied or forwarded. The Prophet algorithm reduces the number of copies, but not every possible communication path through the network is used any longer.

Beside Epidemic Routing, Spray&Wait and Prophet Routing, many more routing algorithms exist. *MaxProp* [11] and *Nectar* [12] both try to optimize packet distribution, duplication and deletion using delivery-likelihoods and neighborhood indexes respectively. *Biological* [13] and *social* [14] inspired DTN routing algorithms exist. Detailed data about more DTN routing algorithms can be found in the literature [15] [16].

## III. Concept of Data-Driven Routing

None of the existing approaches selects and combines routing algorithms based on the requirements of the data to be transfered. Data-Driven Routing allows delay-tolerant scenarios to deliver time-critical data ahead of other, less critical data. Most important, it ensures best possible latency values when needed while saving network resources.

### A. Prioritize Data

In many DTN scenarios important data is defined by data that is time-critical (cf. Section I). Data-Driven Routing *does not* require a certain CoS mechanism nor it is defining its own mechanism for data prioritization. Instead, Data-Driven Routing is *compatible* to existing prioritization mechanisms which are based on classes or queues.

A simple class-based prioritization mechanism is used to *demonstrate* Data-Driven Routing in this paper. The mechanism features three classes from *High* to *Low*, where data of class *High* represents the most time-critical data:

- Class *High*: Important data / time-critical data (e.g. machine fault notifications)
- Class *Middle*: Normal data (e.g. status data)
- Class *Low*: Background data (e.g. tracing data)

Data of class *High* is transfered first when communication opportunities occur. Data of class *Middle* is next, followed by data of class *Low*. Again: Other prioritization mechanisms (e.g. including Weighted-Fair-Queuing or Priority Jumps) may be used if needed.

### B. Combining Routing Algorithms

Copy-based routing algorithms try to use every chance of finding a communication path through the network by flooding it with packet copies (cf. Section II). This approach will quickly lead to an overloaded network, unable to deliver any data. History-based algorithms such as Prophet reduce the number of packet copies by the calculation of encounter probability values. The learning of reliable communication routes may take time and not every communication possibility will be used. Furthermore, Prophets algorithm is unable to handle highly mobile and dynamic nodes. This points out that some DTN routing algorithms may have their strength in finding communication paths as fast as possible and their drawbacks in saving network resources. Others might save network resources efficiently but have drawbacks in fast delivery.

DTN scenarios apply a DTN routing algorithm that seems to fit the scenario requirements best. However, a single routing algorithm is not able to fulfill the requirements of every prioritization class. *In other words:* Existing DTN algorithms are not able to deliver time-critical data quickly without exhausting network resources in dynamic scenarios. There is no option to transfer time-critical and non-critical data differently.

Ensuring fastest delivery of time-critical data (data of class *High* in the example prioritization scheme) while saving network resources at the same time requires a *combination of DTN routing algorithms*. The combination of algorithms is the *key concept* to optimize a DTN scenario for all kinds of data relevant in the scenario. Data-Driven Routing is realized by assigning every prioritization class a DTN routing algorithm that *firstly* fits best to the requirements of the class data. *Secondary*, the requirements of the overall scenario are taken into account. If classes have similar requirements, the same routing algorithm may be used.

## IV. Evaluation of Data-Driven Routing

Data-Driven Routing is evaluated in an agricultural grain harvest scenario using Epidemic, Spray&Wait and Prophet Routing.

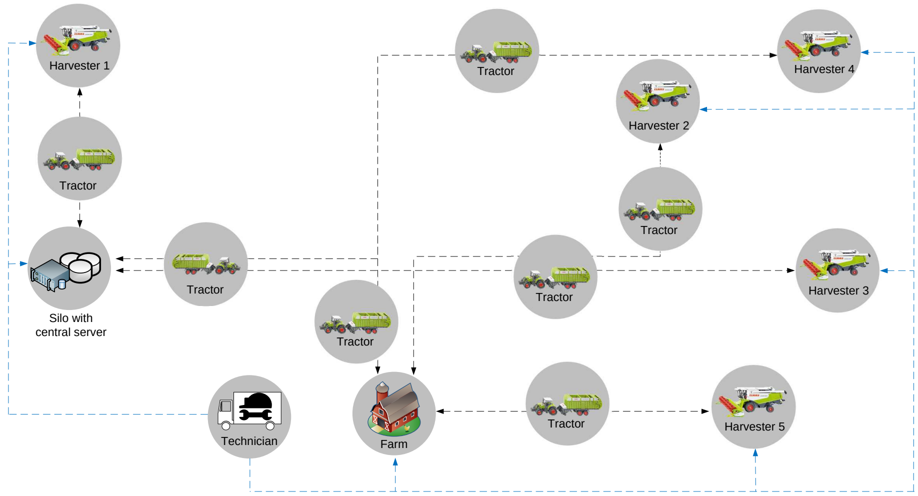


Fig. 1. Grain harvest scenario

### A. Simulation Scenario

Fig. 1 illustrates the setting of the grain harvest scenario. It consists of five harvesters working on different grain fields. Tractors pick up and transport the grain to a central silo or a farm. A technician is checking operation by moving between harvesters, silo and farm. Scenario runtime is $10hours$, representing a typical day of harvest. Tractors transport grain at a speed of $30km/h$ while the technician moves around at $50km/h$. Every time a tractor reaches a harvester, silo or farm, it waits a random time between $1.5min$ and $3min$ to load / unload grain. In contrast to tractors, the technician stays about $15min$ to $60min$ at a destination.

Table I shows details about sources, destinations and frequencies of the scenario data. Prioritization classes *High* to *Low* (cf. Section III-A) are used, defining machine fault data as most important and time-critical data. Picture data is used for documentation and represents non-critical data.

TABLE I
PRIORITIZATION CLASSES FOR SCENARIO DATA

| **Type** | **Direction** | **Freq.** | **Size** | **Priority** |
|---|---|---|---|---|
| Error data | Harvester to technician | $100s$ | $200Byte$ | High |
| Status data | Harvester to silo | $60s$ | $200Byte$ | Middle |
| Picture data | Harvester to silo | $20s$ | $50,000Byte$ | Low |

### B. Simulation Setup

The scenario is evaluated using OMNeT++ [20] and the INET framework [21]. DSDV is applied as MANET routing algorithm, UDP realizes all data transmissions. For the delay-tolerant communication, a basic Bundle Protocol as well as Epidemic, Spray&Wait and Prophet Routing have been implemented.

Four different routing configurations are applied for evaluation. The first two configurations form the evaluation basis by telling all prioritization classes to use *Epidemic* Routing and *Prophet* Routing respectively. In addition, two configurations represent Data-Driven Routing. The first one (*Epidemic-Prophet-Prophet*) applies Epidemic Routing to *High* class data and Prophet Routing for *Middle* as well as *Low* class data. The second Data-Driven configuration (*Epidemic-Spray&Wait-Prophet*) also applies Epidemic Routing to *High* class data. However, *Middle* class data uses Binary Spray&Wait Routing (four copies) and *Low* class data is configured to apply Prophet Routing (no copy threshold at $0.85$). If the delivery probability of the next node is above $0.85$, packets are forwarded instead of copied.

### C. Simulation Results

The simulation results focus on the analysis of the *delivery ratios*, the *data transfer latencies* and the amount of *created packet copies*. All values affect each other.

Fig. 2 presents the averaged delivery ratios of the different routing algorithms *Epidemic*, *Prophet*, the Data-Driven combination of both *Epidemic-Prophet-Prophet* and the second combination *Epidemic-Spray&Wait-Prophet*. Epidemic Routing reaches high delivery ratios for time-critical and non-critical data. However, starting at $100MByte$ of produced data per hour the delivery ratios decrease constantly due to overload situations in the scenario. A closer look at time-critical *High* class data for the Epidemic Routing Algorithm proves the success of handling data by priority classes (cf. Fig. 2 b). Delivery ratios are slightly better compared to all data in Fig. 2 a). Prophet performs considerable worse due to required learning times for meaningful probability values. Furthermore, Prophet is not able to handle the highly mobile technician. Prophets delivery ratios for *High* class data are $10-15\%$ worse compared to the epidemic approach. *Epidemic-Spray&Wait-Prophet* performs best for *High* class data. However, due to the extensive use of packet copies for classes *Middle* and *Low* the delivery ratio for all data is low (cf. Fig. 2 a). The combination of *Epidemic-Prophet-Prophet* provides very

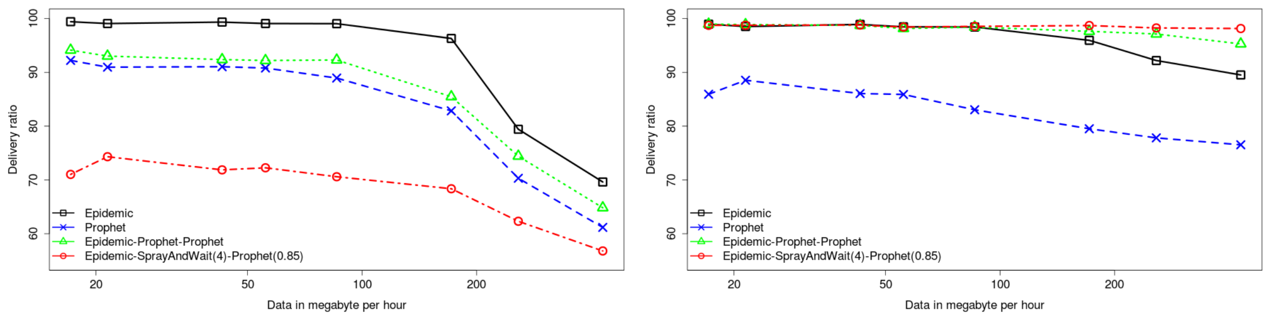


Fig. 2. Delivery ratios for (a) all priority classes and for (b) high priority class

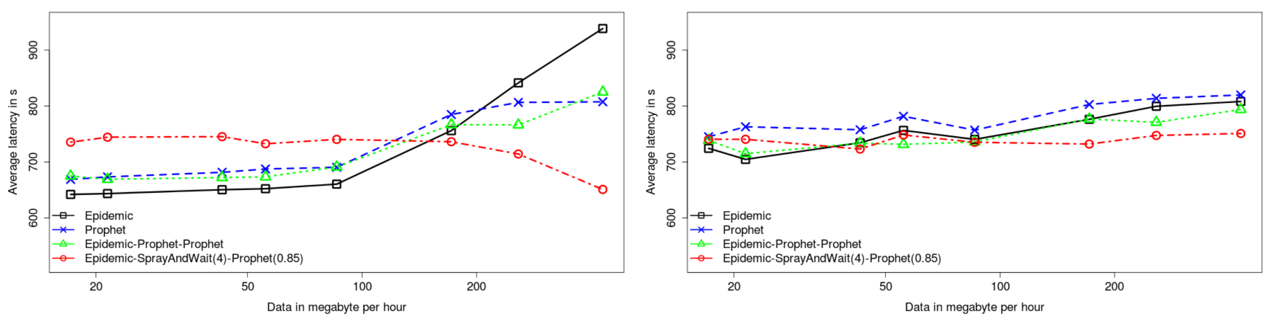


Fig. 3. Latencies for (a) all priority classes and for (b) high priority class

high and stable delivery ratios for time-critical data, especially at high network loads. In this configuration the epidemic approach is only applied to class *High* data which leads to later network overload situations.

The latencies for all and time-critical data is illustrated by Fig. 3. Epidemic achieves the lowest latency values until the network overloads and latencies for *Low* and *Middle* class data start to grow extensively. Prophet has slightly higher latency values, again due to learning of probabilities. However, it is worth to point out that Prophet delivers less data packets to the destination (cf. Fig. 2). The latency values do not consider undelivered packets. The combination of *Epidemic-Spray&Wait-Prophet* performs good for *High* class data due to the usage of Epidemic Routing. In contrast, Fig. 3 a) shows much higher latencies for *Epidemic-Spray&Wait-Prophet*. The reason for this is based in the restricted number of packet copies for priorities *Middle* and *Low*, which results in limited distribution of the data packets. If all four copies are used, packets have to wait until the current node encounters the destination. *Epidemic-Prophet-Prophet* performs best and provides solid latencies for all data classes.

Fig. 4 illustrates the created packet copies in the agricultural scenario. *Epidemic* Routing clearly creates the most packet copies which leads to low delivery ratios at high network loads. *Prophet* is identified as a more resource saving routing algorithm. The *Epidemic-Prophet-Prophet* configuration results in slightly higher packet copies compared to Prophet, but much

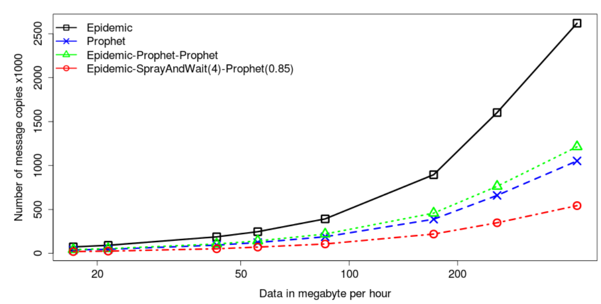


Fig. 4. Total number of data packet copies

less compared to Epidemic Routing. *Epidemic-Spray&Wait-Prophet* saves the most network resources. However, it also fails to provide solid delivery ratios.

The evaluation of the presented Data-Driven Routing concept points out the need for appropriate routing algorithms depending on the requirements of the data to be transfered. Epidemic Routing only works well for low network loads. Prophet Routing is not able to achieve high delivery ratios for important data. A configuration of *Epidemic-Spray&Wait-Prophet* only works well for data of class *High* and presents a non-optimal combination of DTN routing algorithms. However, *Epidemic-Prophet-Prophet* meets the requirements needed for the agricultural DTN scenario. This configuration enables fast delivery of time-critical data while saving network resources when transferring less critical data at the same time. Data-Driven Routing may enable DTN scenarios to optimize delivery and resource usage by data prioritization and DTN routing combination.

## V. Conclusion

Most delay-tolerant scenarios include a broad range of data equipped with different requirements for time-critical delivery. As of today, DTN scenarios select a DTN routing algorithm that fits best. Due to varying data requirements this is not feasible for many scenarios. Consequences are network overload situations and / or high delivery latencies. The concept of Data-Driven Routing for DTNs combines prioritization classes with appropriate DTN routing algorithms to meet scenario and data delivery requirements at the same time. Flexible, scenario-based selection of data prioritization mechanisms and allocation of DTN routing algorithms allows to use every chance for transferring time-critical data through the DTN while saving network resources at the delivery of less or non-critical data. The concept evaluation using an agricultural grain harvest scenario demonstrated the need for and success of a Data-Driven Routing mechanism in DTN scenarios. Compared to the usage of a single DTN routing algorithm, the Data-Driven solution was able to deliver time-critical data as fast as possible while not overloading the networking with less-critical data.

## References


[1] Komobar research project. [Online]. Available: http://www.komobar.de

[2] M. Barreiros and P. Lundqvis, *QoS-Enabled Networks: Tools and Foundations*. John Wiley & Sons, 2011.

[3] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 27–34.

[4] K. Fall, W. Hong, and S. Madden, "Custody transfer for reliable delivery in delay tolerant networks," *IRB-TR-03-030, July*, 2003.

[5] K. Scott and S. Burleigh, "Bundle protocol specification," *IETF RFC 5050*, 2007.

[6] Y. Shao, C. Liu, and J. Wu, *Delay-Tolerant Networks in VANETs*. Taylor & Frances, 2008.

[7] A. Vasilakos, Y. Zhang, and T. Spyropoulos, *Delay Tolerant Networks: Protocols and Applications*. CRC Press Inc., 2011.

[8] A. Vahdat, D. Becker *et al.*, "Epidemic routing for partially connected ad hoc networks," Citeseer, Tech. Rep., 2000.

[9] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 252–259.

[10] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *Service Assurance with Partial and Intermittent Resources*, pp. 239–254, 2004.

[11] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proc. ieee infocom*, vol. 6. Citeseer, 2006, pp. 1–11.

[12] E. de Oliveira and C. de Albuquerque, "Nectar: a dtn routing protocol based on neighborhood contact history," in *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009, pp. 40–46.

[13] C. Blum and D. Merkle, *Swarm Intelligence: Introduction and Applications*, N. C. Series, Ed. Springer, 2008.

[14] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2008, pp. 241–250.

[15] C. P. Mayer, *Hybrid Routing in Delay Tolerant Networks*. KIT Scientific Publishing, 2012.

[16] M. A. Islam, *Routing Issues in Opportunistic Networks: Evolution of Delay Tolerant Networks from Mobile Ad-hoc Networks to Opportunistic Networks*. LAP LAMBERT Academic Publishing, 2012.

[17] L. Wang and S. Olariu, "A two-zone hybrid routing protocol for mobile ad hoc networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 15, no. 12, pp. 1105–1116, 2004.

[18] J. Whitbeck and V. Conan, "Hymad: Hybrid dtn-manet routing for dense and highly dynamic wireless networks," *CoRR*, vol. abs/1006.3426, 2010.

[19] Y. Chen, W. Zhao, M. Ammar, and E. Zegura, "Hybrid routing in clustered dtns with message ferrying," in *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, ser. MobiOpp '07. New York, NY, USA: ACM, 2007, pp. 75–82.

[20] Omnet++ simulation framework. [Online]. Available: http://omnetpp.org

[21] Inet framework. [Online]. Available: http://inet.omnetpp.org