Efficient Publish/Subscribe-based Multicast for Opportunistic Networking with Self-Organized Resource Utilization

Janico Greifenberg
Dampsoft GmbH

jgre@jgre.org

Dirk Kutscher
Technologiezentrum Informatik
Universität Bremen
dku@tzi.org

***Abstract*— This paper presents a publish/subscribe-based multicast distribution infrastructure for DTN-based opportunistic networking environments. The distribution approach is designed to combine an effective distribution of content to interested nodes in the presence of resource constraints, mobility and unstable connectivity. By considering local resource constraints such as limited storage space and limited available bandwidth at opportunistic contacts as well as knowledge about interest for content in the network environment, nodes make local decisions about resource utilization and DTN bundle prioritization. Without further coordination, this approach uses the overall available network resources more effectively compared to other approaches such as epidemic forwarding. We have implemented this approach and have performed a series of measurements in mobile opportunistic networking scenarios under different configurations.**

I. Introduction

Content distribution is a popular application in mobile communication scenarios. We can distinguish three different delivery models: 1) real-time streaming, 2) on-demand-fetch, and 3) push distribution. With respect to persistent communication requirements, real-time-streaming is typically most critical as the perceived service quality is directly related to the ability to receive and process an isochronous stream of data units, such as video packets of a real-time video stream — which normally requires a permanent end-to-end paths between senders and receivers.

Push-based distribution does not necessarily require interactive end-to-end communication, as it is based on asynchronous distribution of messages or larger application data units to a single or a group of receivers. These relaxed requirements allow for resource-friendly, cost-efficient distribution mechanisms.

In fact, asynchronous, scalable distribution of information units over unidirectional links is used today for cost-efficient large-scale content distribution. For example, some 3G and digital broadcast networks provide commercial services for mass push content distribution, e.g., the EZ Channels service by the Japanese operator KDDI[1].

[1] http://www.kddi.com/english/corporate/news_release/2006/0828a/index.html

In [7], we have shown how push-based content distribution can be provided relying on store-carry-and-forwarding mechanisms and Delay-Tolerant Networking (DTN) [4] concepts, which can be used in scenarios where direct end-to-end communication cannot be achieved due to incomplete or prohibitively expensive infrastructure, high mobility, over-utilization etc. Based on the notion of end-to-end DTN communication in content distribution scenarios, we have also shown how DTN-based distribution can complement infrastructure-based distribution, e.g., by enabling mobile users to share information bundles that they have received over the infrastructure.

DTN-based store-and-forward communication is well suited to multicast and data sharing services, since bundle storage and forwarding to multiple contacts is an essential feature of DTN bundle routers. The goals for content distribution are: achieving a high total delivery rate for content bundles to interested receivers, achieving a fairly distributed delivery rate for content bundles from different sources, achieving minimum delivery delays, and optimizing resource utilization.

DTN routing/replication approaches try to adapt forwarding and replication in a way that these goals can be met. Earlier work [6] has shown that knowledge about a network's condition can help to enhance performance with respect to these goals, mainly because basic replication approaches such as flooding would incur a sub-optimal resource utilization. In DTN networks, scarce resources are 1) communication resources (e.g., overall spectrum resources, bandwidth at a contact between two nodes) and 2) local storage and processing resources at individual bundle routers. For DTN routing, approaches such as MaxProp [2] have been developed, that apply a bundle prioritization scheme based on path likelihoods to peers according to historic data.

In this paper we are presenting an approach for publish/subscribe-based content distribution in DTN networks. The intended use case is non-real-time distribution of larger, self-contained information bundles, such as web pages, audio/video podcast files etc. In publish/subscribe communications, nodes are decoupled in time, space and synchronization [3], which corresponds to the disconnected nature of nodes in a

DTN network. Unlike traditional centralized publish/subscribe approaches, we implement subscription management and distribution as a distributed function in the network.

We are exploiting knowledge about explicitly formulated interest in content (subscriptions) to provide a more efficient and resource-friendly replication system compared to flooding and other basic approaches. Nodes in the network individually apply prioritization to control bundle processing based on their local resource constraints. We show how this approach can cope with disruption, network-topology changes and how the self-organized local prioritization can lead to a better overall performance with respect to the above-mentioned goals.

The rest of this paper is structured as follows: section II compares our approach to existing work in the DTN routing domain, section III presents our protocol, and section IV provides evaluation results from measurements and simulations. The main results are summarized in section V.

II. Related Work

Routing in opportunistic networking scenarios has been studied from different perspectives. Since information units, e.g., DTN bundles, typically have to be stored on nodes for later communication opportunities, the concept of *replication* instead of or in addition to *forwarding* is often considered useful. Routers forward multiple copies of bundles for increasing the delivery probability or for improving the performance with respect to other metrics.

Epidemic routing is a simple approach in DTN networks that works without knowledge about topology and future contact schedules to maximize the distribution of bundles by replicating all bundles to all contacts. In [12] the authors describe an approach where opportunistically communicating nodes try to exchange in-transit bundles in a pairwise fashion with the goal that eventually the bundles will arrive at the intended receivers. At a contact, each node performs local decisions, e.g., based on resource constraints and current resource utilization , whether to accept a bundle or not. Spray and Wait [11] also routes by replicating bundles to opportunistic contacts but it is more conservative about resource utilization.

These approaches have the advantage that they can be applied without any knowledge about the network and that it does not require any kind of global coordination. However, [6] has shown that DTN routing protocols can perform better the more knowledge about the network is available, e.g., knowledge about future contacts, resources (storage and communication) etc. With respect to epidemic routing, knowledge about the network could be used to avoid unnecessary replications thus achieving a more efficient of network resources. Different approaches that try to leverage such knowledge have been developed:

PROPHET [8] is a probabilistic routing protocol that uses a history of DTN node contacts in order to calculate delivery probabilities for individual bundles. When nodes meet, they exchange delivery probability information that is the basis for local decisions for a node which bundles to request from the contact node.

MaxProp [2] is an approach that tries to leverage knowledge about previous contacts in order to prioritize packets with respect to delivery and deletion scheduling. MaxProp ranks packets based on a cost assigned to each destination, which is an estimated delivery likelihood. This is obtained from nodes exchanging their contact history information in the network. In addition, complementary mechanisms such as delivery acknowledgments are applied, which allow nodes to free storage space for packets that have already been delivered.

RAPID [1] is a so-called intentional routing protocol that can optimize a specific routing metric such as worst-case delivery delay by treating DTN routing as a resource allocation problem. In this approach, routing metrics are translated to per-bundle utilities that determine how a specific bundle should be replicated in the system. In order to do that, RAPID tracks network resources through an in-band control channel for approximating a local view of the global network state.

While our approach also leverages information about the network to perform local resource utilization decisions, we focus less on the topology and more on reported interest of receivers (and receiver groups). Nodes send subscriptions to communicate their interest in certain content, and the content is then directed towards those subscribers.

Compared to RAPID, our approach also relies on per-bundle utility calculation for local replication decisions, however we do not require estimates for future contacts. Instead, we assign relative priorities to bundles, which is based on subscription information as an indication for interest in parts of the network.

Compared to our earlier work for channel-based multicast distribution in DTNs [7], this paper's contributions lie in the notion of sharing detailed knowledge about subscription state and in using this knowledge for local decisions on resource utilization and forwarding.

III. The DTN Pub/Sub Protocol

The DTN Pub/Sub Protocol (DPSP) is a probabilistic multicast routing protocol for opportunistic networks. DPSP routers do not try to maintain a view of the network topology and select an optimal path. Instead, the routers replicate bundles to their neighbors in order to get the bundle delivered by multiple hops of store-carry-and-forward. Because bandwidth and storage are scarce resources, not all bundles can be copied to all neighbors. The resources have to be used efficiently to achieve the best possible user experience. For the selected use cases, two factors contribute to the user experience: increasing reliability, i.e., delivering as many subscribed bundles as possible, and reducing delay; these are the goals for DPSP routing. Since information bundles are exptected to be rather self-contained, constant delivery latency (low jitter) is not considered a critical requirement for the protocol.

Content is identified using a channel-based subscription system: interested users subscribe to channels and senders publish content by sending it as DTN bundles to the channel. Channels are identified by URIs which can be used as destination endpoint identifiers in the DTN bundle protocol. We assume that interested subscribers have ways to learn about services and channel URIs. This information could be distributed out-of-band — or over a dedicated, well-known DPSP channel, i.e., similar to the Session Announcement Protocol (SAP [5]) for announcing IP multicast sessions. Content is transmitted in self-contained application data units, so that the receivers can process it when they receive the content even if the sender is not reachable at that time. Thus, senders and receivers are completely decoupled in terms of time and network topology.

Because of this decoupling, a source can publish new data at any time. A new channel does not need to be allocated in advance, it is created when something is published with a new channel URI. When an application publishes content, it passes the payload and the channel URI to its local bundle router. The router creates a new bundle and inserts it into the local storage.

When a node decides to receive data from a certain channel, it issues a subscription. The subscription is independent from the sender and from the sending time at the sender. A node can even subscribe to a channel, before the first content is sent. Subscriptions are passed to all neighbors who build a list of active subscriptions which they forward to their neighbors. Thus, a subscription eventually propagates through the network. In addition to the channel URI, the entries of the subscription lists contain a subscription's creation time, its lifetime, the number of hops from the original subscriber to the current node, and a unique identifier for the subscription so that duplicates can be detected.

An overview of our network scenario is depicted in figure 1. We distinguish the following network entities: 1) sources that send bundles to channels, 2) sinks that subscribe to channels and 3) other nodes that are not interested in specific content bundles but store, carry and forward bundles and subscriptions for others. The number of entities of each type is not limited, there can be multiple sinks, sources and other nodes. Furthermore, a source can send bundles to different channels, but only a single source serves each channel.

The core DPSP operation is the exchange of subscriptions and bundles when two neighbors meet and then establish an opportunistic contact. The sequence of messages is illustrated in figure 2.

When two nodes meet, they first exchange their subscription lists (1). Then each node builds a queue of bundles from the local storage to forward to the neighbor (2). There is one queue for each neighbor, even when a node has multiple contacts at the same time. The bundles in the queue are passed to filter functions that remove bundles whose probability to be delivered is not improved when they are replicated to the current neighbor (3). Then the nodes sort the bundles in their queues by their priorities (4). The bundle's priority is calculated based on the utility of replicating it for increasing its delivery probability. After that, the routers start sending the bundles from the queues (5) until the queues are empty or the contact breaks down (6).

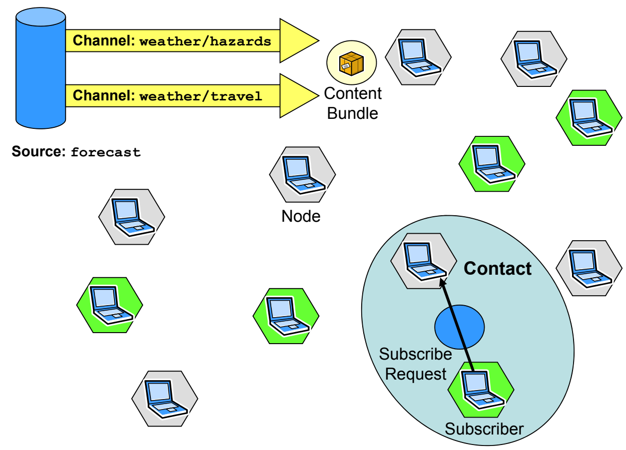

Fig. 1. DTN publish/subscribe scenario overview

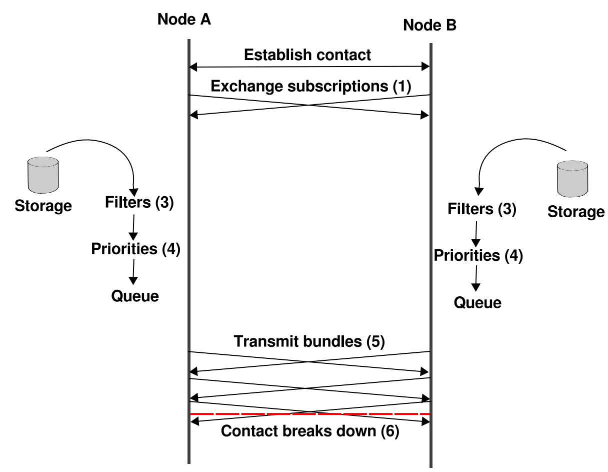

Fig. 2. Contact sequence

Routers do not only receive bundles for channels they subscribe to, but other bundles are distributed pro-actively as well in order to increase the probability of satisfying a subscription and to decrease the average time it takes to deliver bundles to interested sinks.

A. *Selecting Bundles for Forwarding*

DPSP is intended to optimize for *reliability* (delivery probability) and *short delays*. In order to use the scarce contact capacity efficiently to reach these goals, a bundle router needs to select those bundles that should be replicated to its neighbor (steps 3 and 4 of the protocol sequence).

Unfortunately, reliability and minimum delay are sometimes contradictory goals so that a tradeoff is necessary. The forwarding criteria of DPSP allow for different priorities, so that a network can be configured to emphasize either reliability or short delays.

For step 3 of the protocol sequence, we define a set of filters that remove bundles from the router queue, so that they are not even forwarded to the neighbor when the contact is long enough and has enough capacity to transmit all bundles in the queue. There are three filters which can be used in any combination: **Known Subscription Filter**, **Hop Count Filter**, and **Duplicate Filter**. In section IV we evaluate the impact of applying these filters on the performance of the protocol in different combinations.

The **Duplicate Filter** removes bundles that the current neighbor has already received. A node's subscription message provides information about which bundles the node has already seen before.

The **Known Subscription Filter** removes bundles for which neither the current node nor the current neighbor have seen a subscription. This filter avoids forwarding bundles nobody is interested in, but it also impedes the pro-active distribution of bundles.

The **Hop Count Filter** removes bundles if the neighbor's corresponding subscription provides a higher hop count than the current node's subscription. The intention of this filter is to prefer shorter delivery paths. The disadvantage of this filter is that it assumes a stable and symmetric path.

For ordering bundles based on assigned priorities (step 4 of the protocol sequence), we define a set of heuristics to determine the priority of bundles. A heuristic is used to compare two bundles and determine their relative priority with respect to forwarding it to the current neighbor. The priority heuristics are named **Short Delay**, **Long Delay**, **Subscription Hop Count**, and **Popularity**. Their effectiveness is evaluated in section IV. Priority heuristics can be combined with other priorities and with any combination of filters.

Short Delay compares bundles by their age (creation time) and prefers newer bundles, aiming at minimizing the delivery delay. The disadvantage here is that those subscribers who can only be reached by a long path from the sender are less likely to receive a bundle before it expires.

The **Long Delay** heuristic compares bundles by their age and prefers older bundles, so that all bundles are more likely to be delivered before they expire, even when their path is long. The disadvantage is, that the average delivery delay is likely to increase.

The **Subscription Hop Count** heuristic compares bundles by the hop count of the subscription to the bundle's channel. This hop count is a metric for the distance between the current neighbor and an original subscriber. The intention of this heuristic is to prefer forwarding bundles that are already close to their destination. This assumes that the path of the subscription and the path of the bundles are roughly similar in length. It does not mean the bundle needs to pass the same nodes as the subscription, but that the current node and the subscriber still have about the same number of hops between them. This is the case, for example, when nodes move around in groups.

The **Popularity** heuristic compares bundles by the number of nodes subscribed to the bundle's channel, i.e., the priority is based on channel's popularity. When bundles for popular channels are assigned higher priorities, the total number of subscribers receiving bundles is likely to increase. On the other hand, the subscribers of unpopular channels will have a worse user experience, as many of their bundles are dropped along the way.

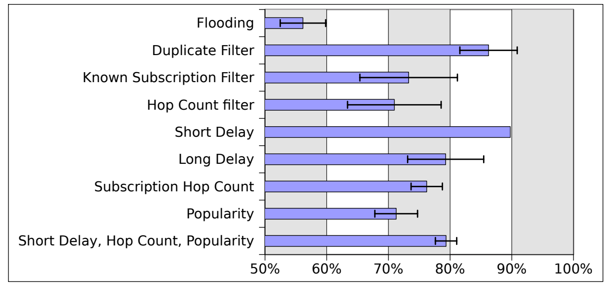

Fig. 3. Delivery rate

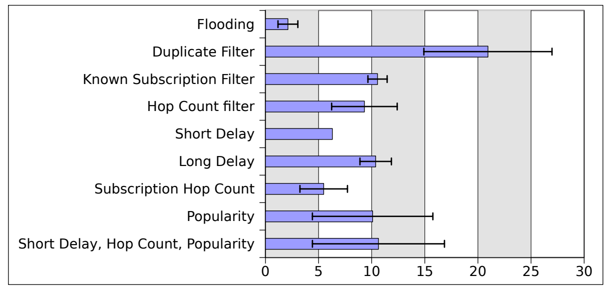

Fig. 4. Delivery delay (in seconds)

B. *Storage Management*

Another scarce resource in opportunistic networks is the storage space on the nodes that carry the bundles, so it needs to be managed effectively, too. In essence, received bundles are competing for storage space, and bundle routers need to determine which bundles to delete, when storage capacity is exceeded.

A naive implementation would simply drop any incoming bundles when the storage limit is reached. This has the disadvantage that bundles are removed that would otherwise be delivered soon, while older bundles without a chance to reach their destination, fill the storage space until they expire.

Our implementation uses the same heuristics for storage management as for assigning priorities in the forwarding queues. The **Subscription Cop Count** heuristic uses the locally available subscription information only, because no current neighbor's data is available. The bundles with the lowest priorities are deleted first. This approach assures that the bundles that are preferred for forwarding are available from the store.

IV. Experimental Evaluation

We have implemented the DTN Pub/Sub Protocol for RDTN[2], our Ruby-based DTN bundle router, and used it to conduct two series of bundle transmission tests, one with the

[2]http://dev.tzi.org/retrospectiva/projects/rdtn/wiki/rdtn

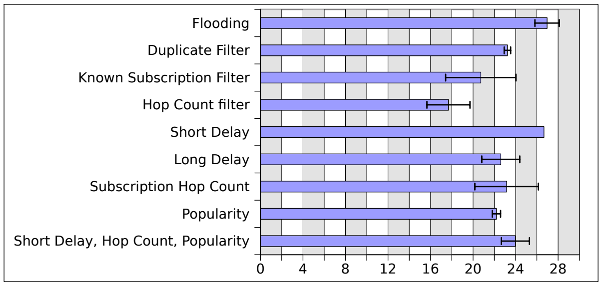

Fig. 5. Average number of replicas

Kasuari emulation framework [9][3] and one with a custom simulator that we have integrated into RDTN.

Kasuari allows us to emulate the behavior of opportunistic network links between virtual Linux nodes on a physical host computer. Kasuari uses Xen[4] virtual machines, which are connected via virtual network interfaces. The link behavior is emulated in real-time, controlled by mobility simulations using an adopted version of the ns2 network simulator[5]. Running experiments on virtual machines allows us to use our actual implementation instead of one that is only written for a simulator.

Kasuari provides a realistic runtime environment for proof-of-concept measurements with the actual implementation, however, the virtual machine-based approach requires significantly more hardware resources than traditional means of simulation, which limits the maximum number of nodes in the test network. In order to test our routing scheme in larger scenarios, we have created a discrete event simulator that works directly with the RDTN code.

For assessing DPSP's and our heuristics' performance, we compare different configurations to each other, taking simple epidemic forwarding (flooding) as a reference. We have conducted nine experimental runs with different configurations: one test case with epidemic forwarding, one test case with each priority heuristic, a run with a combination of short delay, subscription hop count and popularity, and one test case for each filter. All tests of filters are run with the combination of all the priorities, so that we have a baseline to measure the impact of the filter.

The Kasuari-based experiments use random waypoint scenarios which model the unpredictability of opportunistic networks. The simulated networks consist of 30 nodes moving at a speed of $20m/s$ on a $1500m \times 1500m$ area, each with a communication radius of $250m$. Three nodes are designated as senders, each publishing bundles for a separate channel. Five different nodes are configured as subscribers, each subscribing two or three channels. The channels provide between one and four subscribers. The other nodes implement DPSP, but do not send or subscribe any bundles. All nodes have a storage capacity of 50KB.

[3]http://www.kasuari.org

[4]http://www.cl.cam.ac.uk/research/srg/netos/xen/

[5]http://www.isi.edu/nsnam/ns/

For the test cases with epidemic forwarding, we use the same configuration of senders and subscribers as in the DPSP cases, although subscriptions are not distributed. The subscribers are only needed to determine, when a bundle has been delivered.

Each simulation run takes one hour. The senders generate bundles with a payload size of 1KB, at a rate of four bundles per minute, but stop after 58 minutes to give the last bundles a chance to propagate through the network.

Our measurements have shown that with this configuration, many contacts are long enough to transmit the entire queue, so that only the storage capacity is a scarce resource. However, as we use the same methods to manage the storage that we use to organize the transmission queues, we are confident that experiments with other mobility configurations will support the findings presented here.

The results (see figures 3, 4, and 5) show that DPSP achieves a better delivery rate than epidemic routing (up to 33% improvement for the configuration using the short delay heuristic). Epidemic forwarding, however, delivers bundles with a lower delay than any of the DPSP configurations, because epidemic forwarding prefers relatively short paths. For longer paths, the probability that a bundle is randomly dropped, increases significantly, which explains the low delivery rate.

We also see that the impact of the different heuristics varies significantly. The general trend we observe here, is that those heuristics that make less assumptions about the topology perform better.

The **Duplicate Filter**, even though a very simple measure, improves the delivery rate in this experiment more than the other filters. However, its performance with respect to the delivery delay is significantly worse than that of any other filter, which we have to analyze more deeply in future measurements.

The **Known Subscription Filter** and the **Hop Count Filter** decrease the delivery ratio of the network compared to the other configurations, but they also reduce the number of replicas for each bundle. The **Hop Count Filter** has a positive effect on reducing the delay, because it leads to shorter path. This effect is only small in this setting, as the symmetry assumption of hop count filtering hold only rarely due to random-waypoint mobility.

The **Short Delay Heuristic** performs very well both in terms of delivery ratio and average delay. Its inverse, **Long Delay**, yields a slightly worse delivery ratio and its average delay is higher than for the **Short Delay** heuristic, as expected.

The **Subscription Hop Count** heuristic is less successful for both goals than **Long Delay** and **Short Delay**, because it makes the assumption that the path length remains relatively constant, which does not hold in our random network.

With the **Popularity** heuristic, we see a lower delivery rate compared to the other priorities, but still an improvement compared to epidemic routing. The reason for this is, that while some channels have more subscribers, there are more subscribers for the less popular channels altogether so that the overall delivery rate drops. Those bundles that are delivered, are delivered faster than under the other heuristics.

The combination of priority heuristics average out the variations of the separate heuristics. While this is suboptimal for the networks evaluated here, we expect that the relative performance of the heuristics varies in different networks (e.g. subscription hop count probably works better in a network where nodes move in clusters), so that the combination delivers the best performance in general.

The RDTN simulations use a mobility scenario generated by the Generic Mobility Simulation Framework (GMSF)[6] from ETH Zürich. In this scenario for inter-car communication, 100 nodes move around a road network in an area of 3000mx3000m using both a car-follow (i.e. nodes adapt their speed to other nodes "driving" in front of them) and a traffic-light model (i.e. occasional stops at traffic-lights) [10] for realistic movements. Except for that, the parameters are the same as for the Kasuari tests described above.

In this mobility scenario, the nodes meet frequently (approx. seven contacts per minute per node), so that there are many opportunities for transmitting bundles. The high frequency of contacts between different nodes creates loops for the distribution of content bundles, as nodes often move in clusters. We have found out that in these scenarios, loops have to be eliminated, as they cause bundles to be duplicated endlessly (within their time-to-live window). Even when the bundle routers detect and discard duplicate bundles when they have been received received, communication resources would be wasted for the initial transfer. This causes epidemic routing to waste so much resources that only 9% of the subscriptions are fulfilled. Although only ten replicas of a bundle are created on average, each bundle is transmitted 256 times.

DPSP performs better in comparison, delivering 44% of the bundles to their subscribers, when all filters are active, but there are still scenarios where loops can occur. With DPSP in the GMSF-generated mobility scenario, we have seen that, on average, a bundle is transmitted 140 times, while creating only 19 replicas. This means, there have been many redundant transmissions, and as a result, communication resources have been wasted more than intended. This is due to the fact the current set of filters cannot perform duplicate suppresion over multiple hops, which can be an issue for dense large-scale scenarios such as the one we have studied here.

V. Conclusions

Publish-subscribe distribution is a useful communication pattern for opportunistic networking scenarios with DTN store-and-forward services. In DTNs, nodes are inherently decoupled, which fits nicely into asynchronous publish-subscribe operation. For multicast distribution, DTN networks with their intrinsic potential for replication strategies can enable comparatively simple yet effective routing mechanisms.

However, in order to provide such a resource-friendly service to users, some requirements with respect to reliability, timeliness have to be met — moreover a corresponding system has to be robust with respect to network-topology changes caused by mobility and disruptions.

Taking **Flooding** as a starting point, we have asked the question: *how can knowledge about the network and the subscribers' interest be leveraged to optimize the distribution service with respect to overall delivery rates and delays?* In other words: how can the limited network resources be utilized most efficiently?

In DTN scenarios, centralized coordination services are likely to fail, as coordination channels may not be available due to disconnections. Therefore, we have applied a self-organized coordination scheme, where nodes perform local resource utilization decisions by considering their own resources (bandwidth and storage) only. Based on this approach and the channel-based publish/subscribe pattern, we have analyzed the performance of different filtering and prioritization algorithms in an opportunistic networking scenario with respect to delivery rates and delays, applying both random-waypoint and generated inter-car communication traces as mobility scenarios.

Our measurements with a complete implementation for the RDTN bundle router have shown that each of our proposed algorithms results in an improved performance with respect to delivery rates compared to **Flooding**. On the other hand, we have shown that **Flooding** provides a relative good performance with respect to average delay rates.

In this specific mobility scenario, we have seen that an algorithm such as preferring bundles with a more recent creation time (**Short Delay**), performs better with respect to delivery rates than any other tested algorithm or combination of algorithms. It also performs reasonably well with respect to delays. This is noteworthy since **Short Delay** does not consider any information about the network and subscribers' interest.

We ascribe these results to the following factors: 1) the mobility model does not provide enough stability over time, which increases the likelihood that knowledge about the network becomes inaccurate very fast. 2) When optimizing for a short *average* delivery delay, **Flooding** performs well due its aggressive replication approach, which seems to help to deliver bundles fast to a significant amount of subscribers. However, since some nodes are never reached due to inefficient resource utilization, **Flooding** must be considered sub-optimal with respect to fairness and reliability. As a take-away result, we can state that the applied algorithms should be carefully selected with respect to the network topology and the predominant mobility model.

Our analysis of DPSP performance in dense large-scale scenarios has shown that duplicate suppression must not be limited to hop-by-hop exchanges (as we currently do) but must instead be extended in order to avoid loops over multiple hops, which can lead to unwanted bundle transmission and wasted communication resources that would better be spent for relevant bundle exchanges. Fundamentally, there is a conflict between pro-active bundle distribution (between any two

[6]http://polar9.ethz.ch/gmsf/

nodes) – which is considered useful to utilize contact time as efficient as possible – and the overall resource utilization. Our results are interesting because they indicate that for a better overall performance of the network, under some circumstances pro-active distribution should be reduced even further. One of our eminent tasks for future work will be to formulate DPSP filters that whould implement this approach.

Moreover, we can state that it is also important to consider the communication pattern and the delivery characteristics. When *long-tailed content distribution* is predominant, i.e., there are many channels, each of which is only attracting a small number of subscribers, prioritizing bundles based on their **Popularity** is not optimal for achieving high average delivery rates.

In summary, our approach of applying a configurable set of filters and prioritization schemes for performing local resource utilization decisions in DTN publish/subscribe scenarios has led us to some interesting insights about how network knowledge can be (practically) applied for optimization overall resource utilization in such scenarios. One of the merits of our approach is the unified DTN-based distribution architeture that allows the flexible integration of different filter/routing schemes, which has enabled us to perform a assessment of different schemes that we have documented in this paper.

References

[1] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. Dtn routing as a resource allocation problem. In *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 373–384, New York, NY, USA, 2007. ACM.

[2] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. *Proceedings of IEEE Infocom*, 2006.

[3] P.T. Eugster, P.A. Felber, R. Guerraoui, and A.-M. Kerrmarrec. The many faces of publish/subscribe. *ACM Computing Surveys*, 35(2), June 2003.

[4] Kevin Fall. A Delay-Tolerant Network Architecture for Challenged Internets. Proceedings of ACM SIGCOMM 2003, Computer Communications Review, Vol 33, No 4, August 2003.

[5] Mark Handley, Colin Perkins, and Edmund Whelan. Session Announcement Protocol. RFC 2974, October 2000.

[6] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in Delay Tolerant Networks. Proceedings of the ACM SIGCOMM 2004 Conference, Portland, OR, USA, 2004.

[7] Dirk Kutscher, Janico Greifenberg, and Kevin Loos. Scalable dtn distribution over uni-directional links. ACM SIGCOMM Workshop on Networked Systems for Developing Regions (NSDR), September 2007.

[8] 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.

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

[10] Philipp Sommer. Design and Analysis of Realistic Mobility Model for Wireless Mesh Networks. Master's thesis, ETH Zurich, 2007.

[11] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, New York, NY, USA, 2005. ACM.

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