

## Priority Based Data Dissemination in MANET

<sup>1</sup>Florian Pregizer and <sup>2</sup>Alexander Sauer

*Institute of Information Technology; Ulm University*

<sup>1</sup>florian.pregizer@uni-ulm.de; <sup>2</sup>alexander.sauer@uni-ulm.de

### ABSTRACT

Opportunistic networks can provide network services in an infrastructureless ad-hoc manner. This is especially useful if either no infrastructure is available or an existing infrastructure is not to be used. This can be due to high cost, or security considerations. Much effort went into the development of protocols and other theoretical background over the last decade. However use cases and applications have been neglected. In this paper we present a protocol for priority based data dissemination as well as two applications built on top of the protocol. These were implemented as Android applications on top a framework for ad-hoc communication. To show the functionality and to give some performance metrics, real world evaluations were conducted.

**Keywords:** 802.11, data dissemination, delay-tolerant networking, dtn, dtn-applications, real-world.

### 1 Introduction

Despite the continuing digitalization, increasing interconnectivity and a growing availability of mobile network services, there exist plenty of scenarios where no infrastructure exists or where it is overloaded. During outdoor activities like hiking or mountaineering there often is no, or only bad data connectivity. In general rural areas with low population densities are economically not attractive for providers. But also in urban environments an infrastructure might be unavailable, for example underground or indoors.

After catastrophic events or during power outages a previously existing infrastructure might no longer be available. In this case communication has to be provided by some other means. There are also other factors when one does not want to use an existing infrastructure. Using roaming data services during inter country travel can become quite expensive. Furthermore security considerations can lead to a preference of direct communication between devices and the avoidance of potentially insecure infrastructure. Highly unstable and overloaded data services can be encountered during trade shows or festivals. In these instances channel bandwidth and medium access can no longer accommodate the demand of hundreds of users.

In these aforementioned scenarios smaller decentralized networks like MANETs <sup>1</sup> can help to mitigate some of the problems. Opportunistic networks transmit Messages without prior knowledge of the underlying topology, which in a mobile scenario is constantly changing. Transmitting data in such a network is possible even if there is no end-to-end connectivity between two nodes [1]. Such

---

<sup>1</sup>Mobile Adhoc Networks (MANETs) are comprised of nodes that communicate in an ad-hoc manner. As such they don't rely on preexisting infrastructure.

networks are also called delay tolerant networks, since the delay in a partitioned network cannot be predicted a priori.

Due to the constant changes that are innate to these kind of networks, the contact time between two devices cannot be predicted and might be very short. This gave rise to the idea of a prioritized packet transfer in which "more important" data is exchanged first.

In 2012 Jörg Ott concluded [2] that there is a mismatch between the foundations of delay tolerant networking (such as protocols, models and systems) and real world applications that make use of these foundations. In this paper we present two applications for priority based data exchange in opportunistic networks and a generic protocol on which the applications are based. We aim to provide a solution to reliably distribute data throughout a network while prioritizing relevant data. While other approaches often assume unlimited storage capacity we address the limited memory problem in our protocol design.

The first section gives an overview of related work. Then our own approach is outlined including the protocol design and the packet layout. Afterwards the two applications are presented. The first being a short message service and the second implements a service for exchanging location based on OpenStreetMap data. For each application the evaluation procedure is described and the results presented. The paper finishes with a conclusion and an outlook.

## 2 Related Work

Over the last years many papers have been published that introduce protocols and try to improve the data dissemination rate with various means. Others deal with the correct modeling of mobility or human behavior. In this section we present approaches to opportunistic networking that are application driven and make use of some kind of prioritization.

A prioritization of data has been researched by Batabyal and Bhaumik in [3], who proposed a "Fair queue message scheduling policy". This policy prioritizes based upon the overall waiting time, which is the summarized time a packet had to wait at all previous network links before being transferred over the next hop. To minimize the variance of the waiting time at every hop the message with highest overall waiting time is preferred.

Ott et al. present a floating content concept in [4] where they couple content to an anchor zone. This coupling is combined with a time to live which limits the availability of the content. The anchor zone has a replication range within which the holder of a content item replicates it actively and an availability range which describes the area in which the content items aren't replicated but are still present. Outside these two ranges the content is deleted. To minimize spamming of the network, content items with large anchor zones are deleted when the content holders run out of storage space. Further prioritization is for items with small storage amount which are close to their anchor.

With the so called "Drop policies" Soares et al. propose in [5] three priority classes. Prioritization takes place when too many incoming messages cause a buffer overflow and when the messages are sent. As this could lead to a monopolization of the network resources by the messages of the highest priority class the messages with the longer remaining TTL are preferably transmitted. A longer remaining TTL stands for a higher probability that the message reaches its destination.

While Batabyal and Bhaumik prioritize regarding the overall waiting time, Ott et al. choose the anchor zone, the distance to it and the storage amount as criteria. Soares et al. Order the messages by only three distinct priority classes and their TTL. These three proposals have the disadvantages

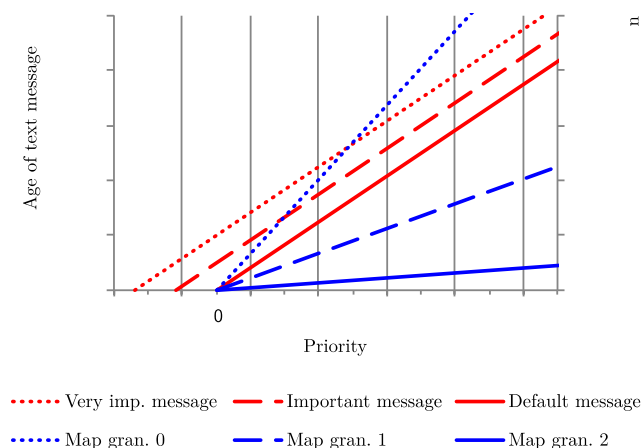
that they aren't content sensitive, they don't prioritize when sending the message or they don't support a continuous prioritization. Therefore this paper proposes a priority based data dissemination with content sensitive prioritization that is used when sending and storing messages.

### 3 Data Dissemination Strategy

The main goal of a data dissemination strategy is to provide a data set to every device in the network - or at least in a certain "hop-distance" or geographical radius. In a mobile environment the goal is limited by certain constraints: Devices only have a finite amount of storage. Power consumption is an issue due to a limited supply of battery power and due to wireless communication channel bandwidth and allocation are problematic. Our approach aims to mainly address the storage problem. Nevertheless the prioritization of data also reduces the number of unnecessary packets and should therefore also reduce energy consumption and channel load. Wireless communications are inherently broadcast in nature. This circumstance is of benefit for data dissemination. Our protocol addresses every device in communication range and as such reduces redundant resends to further reduce the energy and bandwidth footprint. As a further requirement we wanted to implement a protocol that is self pruning, meaning that that "out of date packets" are no longer distributed in the network. This is mainly achieved by a dynamic prioritization function that is described in further detail in the following section.

#### 3.1 Prioritization

As an improvement on the protocols presented in the related work section we introduce a content sensitive prioritization function. This function is not restricted to discrete values but can be implemented as a continuous function. As the priority is determined by the corresponding application, the function needs to be implemented on a per application basis. As an argument it is given an application packet and is expected to return a value denoting a priority value. Per design we chose smaller values to represent higher importance. As this function is dynamic it may produce different values per call. This makes it possible to implement "auto pruning" based on time or location of the device.



**Figure 1: Graph of the priority function for text- and map-data. The lower the priority value the more important the content**

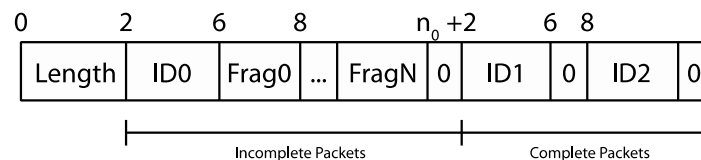
As an example for such a content sensitive function, figure 1 shows the priority value of data packets generated by the applications presented in section IV. The blue data points depict the values for the map application. The red data points are for the text application. The values for the axis and different patterns are explained in the corresponding application sections.

### 3.2 Protocol Design

The protocol design is separated in two parts. The first part is the meta data communication. The meta data trigger the exchange of the actual payload, the data packets, between the single devices. This payload exchange forms the second part of the protocol.

#### 3.2.1 Meta Data

The meta data describe the data that are present on a device with IDs, which are individual hash values for every data element. A meta data packet is shown in figure 2. It includes all IDs of the data sets present on a device.



**Figure 2: Packet layout of the meta data packets. After a length field the available data sets are listed. Each data set is identified by its ID and a 0-terminated list of missing fragments.**

Because a data set can exceed the maximum packet length which is defined by the maximum transfer unit (MTU), the data sets can be fragmented. If a data set is fragmented and some fragments are missing, the continuous numbers of the missing fragments are listed after the data set's ID. The meta data communication is triggered by three events. The first event is the appearance of a new device in the adhoc network. All devices send periodically beacons with their MAC address. With these beacons every device creates a list of its active neighbours, which enables the easy detection of new devices that are unknown to the list.

Another event is the periodical meta data communication, which is important, because the priorities can change continuously. Therefore a new meta data communication could evoke another modified reaction to the current situation. The periodical meta data communication also increases the stability of the system.

The third event that can cause meta data communication is when a device detects that fragments of a data set are missing. After a defined waiting time interval the device with incomplete data sets sends a meta data packet to evoke the sending of the missing fragments.

When a device receives a meta data packet it first the stores the information about ids and fragments in a data structure (senderMap). It then checks if the packets are missing fragments. These are sent first if the matching data set is present to complete data sets before beginning a new transfer. Afterwards the device checks its own data sets against the ones described in the meta data packet, beginning with the one with the highest priority. If a packet is not stated in the received meta data, it is sent immediately.

To avoid continuous sending and overloading of the network, only a defined amount of data sets are sent in a row. The communication is then continued after a new meta data communication.

The following algorithm describes the meta data triggered sending of data packets in pseudo code.

```

Input: New meta-data packet
senderMap ← sender.ids and sender.fragments
while incomplete datasets do
    SEND MISSING FRAGMENTS
end while
i ← 0
while priorityQueue ≠ ∅ ∧ i < 2 do
    data ← priorityQueue.first
    priorityQueue.removeFirst
    if data.id ∉ senderMap then
        i ← i + 1
        SEND DATA
    end if
end while

```

### 3.2.2 Data Packets

The data packets carry the actual payload, which can be fragmented, as stated in section III-B1. The packet layout of a data packet is shown in figure 3. After a mode field that describes whether the data has to be stored in volatile or non volatile storage follows the individual id of this data set. The fields' number of fragments and continuous fragment number inform about the fragmentation and when a data set is complete. The last field is the payload length followed by the payload itself.

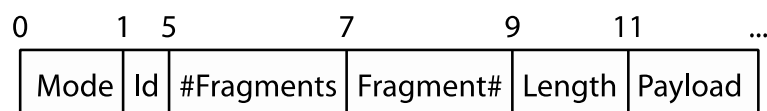


Figure 2: Packet layout of the data packets.

The payload includes the information that is relevant for the content sensitive calculation of the priority like time stamps or geographical data.

When a data set has been sent by another device, triggered by the meta data communication, the receiving device handles every packet as follows. At first there is a check if the data set already exists. When it does and is incomplete, missing fragments are added to it. Otherwise the arriving packet is discarded. This method of fragmentation allows for packets to be reliably reassembled as long as there is no collision of ids. The integrity of the individual packets is entrusted to the underlying transport layer.

Complete data sets allow the calculation of their priority which is compared to the priority of the already present data sets on the device. These are kept in order by the priority queue. When shortage of storage occurs, the priority queue evicts the element with the lowest priority and adds the new element. If the new element has a lower priority than all other elements, it is discarded.

The following pseudo code explains the handling of new packets.

```
Input: dataPacket
id ← dataPacket.id
fragmentList ← dataPacket.fragments
if ∃ Complete Dataset with id then
  DISCARD PACKET
  EXIT
end if
if ∃ Incomplete Dataset with id then
  if fragmentList ⊆ dataset then
    DISCARD PACKET
    EXIT
  end if
end if
dataset ← dataset ∪ fragmentList
if dataset is not complete then
  EXIT
end if
if priorityQueue.size < MAXSIZE then
  priorityQueue ← priorityQueue + dataset
  EXIT
end if
priority ← PRIORITY(dataset)
lowestPriority ← priorityQueue.lowestPriority
if priority ≤ lowestPriority then
  EXIT
end if
priorityQueue ← priorityQueue – lowestPriority
priorityQueue ← priorityQueue + dataset
```

## 4 Applications

### 4.1 Ad-hoc Framework

The ad hoc network was implemented in an ad-hoc framework developed at Ulm University called Adhoc3 which has been proposed in [6] and demonstrated at [7]. The Android software transmits and receives data packets over a wireless

802.11 ad hoc network. Using UDP broadcast packets the data is forwarded in a multi-hop fashion on the application layer. In addition to the packets needed for the actual data transmission, there is an additional type of packet sent. With these so called "hello-packets" the clients exchange information about their current context (e.g. list of neighbours, position, etc.) and network statistics.

Within this framework the protocol for priority based data dissemination was implemented. To test and evaluate the protocol, two applications using the protocol were added to the Adhoc3 application. These will be presented in the following sections.

### 4.2 Text Message Service

The text message application provides an easy exchange of messages between devices in Manet. An advantage to common messenger apps is the usage without infrastructure and internet connection. Because of the possibly very short contact time between the devices, the application uses a prioritization of the messages.

The user interface of the text message service resembles the common appearance of messaging apps. As seen in figure 4 the bigger part of the space is for sent and received messages. The user can

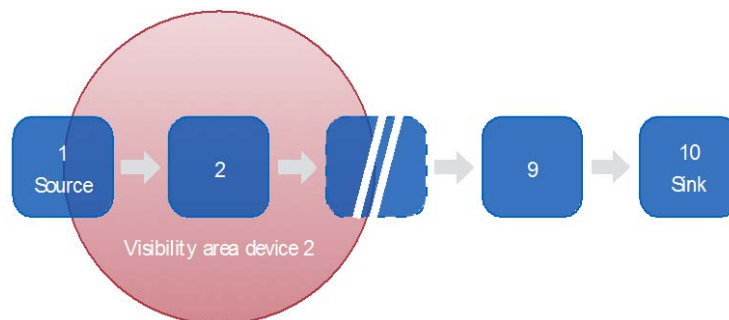
enter and send text, choose a specific user priority for this message and adjust the settings like the user name or the storage size available for the messages.



**Figure 3: User interface for the text message service.**

#### 4.2.1 Linear propagation:

a) Test Setup: For the evaluation ten Android Smart-phones with the installed alternative operating system CyanogenMod have been used. This OS provides the access of extended WiFi features that are used for the ad-hoc mode.



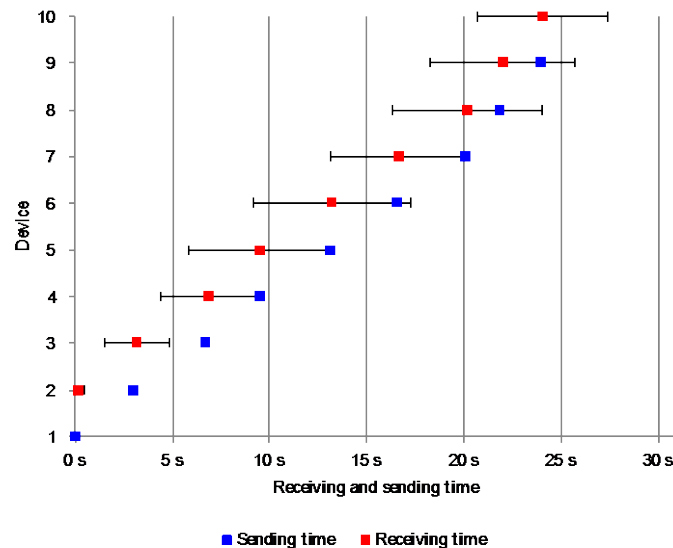
**Figure 4: Test setup of the linear propagation. Every device has maximum two neighbours.**

In this scenario the ten smartphones were arranged in a row to test the propagation of the text messages from the first device via the intermediate ones to the last device. The arrangement was in a way, that every device can only see its direct two neighbours like shown in figure 5.

At test start all devices were prepared to receive messages. Because there wasn't any data present only meta data packets were exchanged. In a ten seconds interval then 21 text messages were initiated on the first source device and sent to all devices in range, which was according to the test setup only the next neighbour for the source device and maximum two devices for the intermediate smartphones. Starting at the source the messages propagated autonomously from device to device until they were present on all ten smartphones.

b) Results: The averaged sending and receiving timestamps of the text messages are illustrated in figure 6. The sent messages propagated in on average 24.14 seconds from device 1 to device 10. The relatively regular propagation of the messages along the linear chain of devices can be seen.





**Figure 5 Hop to hop propagation of text messages. The results were averaged over 21 runs. Error bars show the standard deviation.**

Deviations from this are the result of the variable time a smartphone had to wait until it received a meta data packet that triggered the text message exchange. This lead to the growth of the standard deviation with further propagation because the uncertainty sums up at every device.

### 4.3 Map Application

The map application covers another useful area for the priority based data dissemination. It allows the exchange of map segments between devices, while beginning with the segments that have little information density and cover a large area. In the long run highly detailed segments which contain only information about a small geographic area are added. The second parameter, despite the information density of the files, is the distance of the map segment center to the current location. Segments that are close to the current device location are transferred before remote segments.

The two criteria are merged in one continuous priority function that determines which segment has the highest priority at the moment of transmission. In the map application OpenStreetMap data is used. To achieve small file sizes and continuous zoom and to avoid redundant data, vector map data is used within the application. Therefore the OpenStreetMap XML data had to be prepared.

In order to offer segments with different information density, the main map is separated into three granularities. While granularity 0 includes only the main roads, highways, railways, ferries and rivers, granularity 1 contains smaller roads and greenland. For granularity 2 remain all other categories which are not included in the first two granularities. After the filtering every main granularity map is separated in small segments, that can be displayed individually by the rendering engine. These segments are the ones exchanged by the application between the devices.

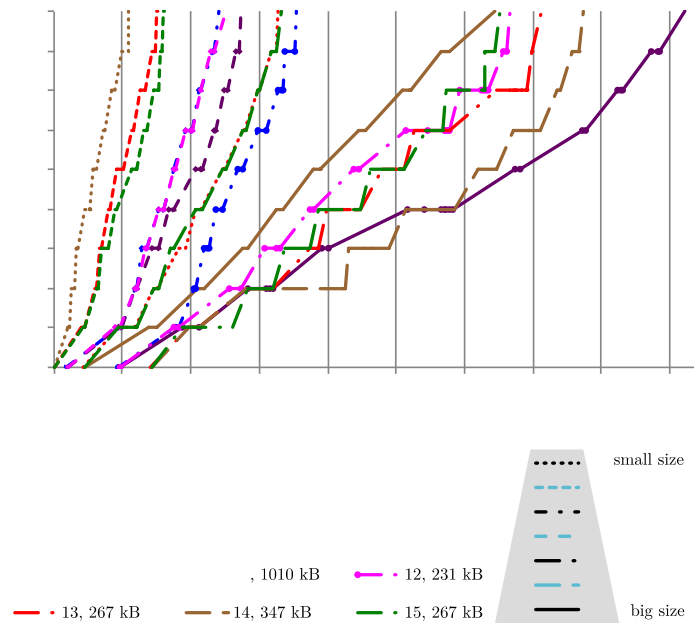
#### 4.3.1 Linear Propagation

a) Test Setup: The test setup of the linear propagation follows the arrangement as described in section IV-B1a. To test the map application an evaluation data set of 15 map segments with all granularities has been created.

b) Results: The on-device logging of the receiving and sending timestamps of the map segments is shown in figure 7. According to the priority based data dissemination the map segments with high



priority were sent first by the source device and arrived regularly in the same order at the last device.



**Figure 6 Hop to hop propagation of map segments. Dotted lines mark smaller segments, whereas solid lines mark bigger map segments. The number in front of the segment size indicates the priority order.**

It can be seen, that the segment size is also a factor. As every segment exceeds the MTU it is fragmented into smaller packets. The more fragments a segment has the higher is the possibility for packet loss when using the unreliable user datagram protocol (UDP). A segment with a low number of fragments can pass a segment with more fragments at any node because only complete segments are transferred. This is due to missing fragments preventing the calculation of the content sensitive priority and therefore other, lower prioritized but complete segments get the chance to be transferred first. This behavior can be seen for example with the first two segments. Although the priority of the first dataset is higher, the smaller segment can be forwarded earlier by every following node.

Furthermore one can see that both big segments can influence the transmission of following smaller segments negatively when the preferred transmission of lost packets survives multiple meta data cycles. The segments 12, 13 and 15 are obviously influenced by the big segment 8, which shows up in the widening of the graph in figure 7 to the right side.

The functionality of the priority based data dissemination is demonstrated by this test. Data segments with a high priority are transmitted first. Only the size of the single data segments and the unreliable UDP transmission could change the order of arrival at the last device.

#### 4.3.2 Route Simulation

a) Test setup: For the second scenario a route was simulated with two devices. Both smartphones were provided with fake GPS data, that simulated the synchronous moving from the start city Ulm to the destination Biberach in Germany. This tested the dynamic change of priority when the devices move, which leads to a change of the location for the priority calculation.

In this scenario one smartphone, the source device, was equipped with the full map data set of the segmented Open- StreetMap data. The second smartphone hadn't stored any map data at test start. It had a storage limit of 15 segments set. The simulation lasted 15 minutes.

On both smartphones the simulation of the GPS signal was started simultaneously. The following start of the map application initialized the metadata transmission. This led to the sending of the highest prioritized map segments by the source smartphone. The second smartphone stored the segments until it reached its storage limit.

The simulated movement of the two devices resulted in a dynamic modification of the segment priorities. Therefore the source device transmitted new map segments that now have the highest priorities. These segments replaced the segments on the second smartphone that were now more remote and lost their high priority.

b) Results: Figure 8a illustrates the situation shortly after the test started. The map segments around the start city Ulm and remote highways have been transmitted to the before empty device with the limited storage capacity. Please note that the map data eastern to Ulm belong to another state and are not part of the map data set.

At the end of the simulation, which is shown in figure 8b, the detailed map segments of the surrounding of Ulm have been replaced by the ones around the destination city Biberach because the latter then had a higher priority. The segments with highways of granularity 0 around Ulm still had a high priority and therefore remained on the device.

The dynamic transmission of map segments is presented in figure 9 where the map segment centers are drawn on a north-up map. The chronological sequence of the simulation is assigned by the color of the markers which is defined by a heat map from beginning to end. The shape of the markers points to the granularity of the segments. One can see that in the vicinity of the starting point Ulm red colors with early chronological assignment dominate, whereas at the end of the route green colors marked with the chronological maximum prevail. A closer look reveals that the remote segments of granularity 0 are transmitted very early in the simulation.

This corresponds to the desired effect to draw a rough picture of geographically big map segments with little information before transmitting highly detailed data. In a smaller scale the segments of granularity 1 are also higher prioritized than segments of granularity 2 and therefore transferred before the latter.

## 5 Conclusion and future work

In this paper we presented an approach to priority based data dissemination for mobile adhoc networks. A generic protocol, that employs a content sensitive prioritization function was proposed. The protocol consists of two parts. One part cyclically broadcasts the available information that a device currently stores. The other part is used to transmit missing data fragments via broadcast to neighboring devices.

Based on the priority function it is ensured, that higher prioritized data is distributed faster throughout the network. On the other hand a dynamic function allows data to age and ultimately be discarded in the network. If the data available to the device exceeds the device storage, the priority function provides the means to discard, or to replace packets in the storage queue.

Two applications were built on top of the protocol. One distributes short text messages throughout the network. The other application exchanges OpenStreetMap data to display a vector based map on the device. The map data was divided into different segments containing different levels of detail with varying section sizes. Utilizing a location based priority function, map segments closer to the device are exchanged first.

The protocol and the two applications were implemented as a smart phone application. Using current of the shelf hardware, the applications were tested in real world scenarios. The results show, that the priority based data dissemination is works well for both applications. On a memory constraint device the protocol in conjunction with the map application performed quite well and was able to keep the important (currently required for displaying a local map) data on the device while discarding unneeded data.

As future work we plan to build further applications based on the priority based data dissemination protocol. Also the existing applications can be improved. Especially for the text application an encryption scheme comes to mind, where public keys are distributed throughout the network and the text messages are encrypted. The protocol itself requires further improvement, as of right now there is no mechanism to prevent neighboring devices from sending non required data over and over again.

From an evaluation standpoint a further investigation into the impact of the different parameters of the protocol should be conducted. Also the limitations regarding device density, contact times or device speed should be investigated. On the other hand the benefits in terms of reduced data transmissions, energy and channel utilization should be subject to further tests.

#### REFERENCES

- [1]. L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 134–141, November 2006.
- [2]. J. Ott, "404 not found: A quest for dtn applications," in *Proceedings of the Third ACM International Workshop on Mobile Opportunistic Networks*, ser. *MobiOpp '12*. New York, NY, USA: ACM, 2012, pp. 3–4. [Online]. Available: <http://doi.acm.org/10.1145/2159576.2159579>
- [3]. S. Batabyal and P. Bhaumik, "A fundamental traffic management strategy for mobile opportunistic network," in *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, Aug 2013, pp. 822–827.
- [4]. J. Ott, E. Hyytia, P. Lassila, T. Vaegs, and J. Kangasharju, "Floating content: Information sharing in urban areas," in *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, March 2011, pp. 136–146.
- [5]. V. Soares, F. Farahmand, and J. Rodrigues, "Scheduling and drop policies for traffic differentiation on vehicular delay-tolerant networks," in *Software, Telecommunications Computer Networks, 2009. SoftCOM 2009. 17th International Conference on*, Sept 2009, pp. 353–357.
- [6]. N. Muslim and F. Pregizer, "Designing a testbed for broadcast opportunistic wireless data dissemination protocols," in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, March 2013, pp. 607–612.
- [7]. F. Pregizer and H. Klee, "Real-time audio transmission in mobile adhoc networks," in *2013 IEEE 5th International Symposium on Wireless Vehicular Communications (WiVeC 2013)*, June 2013.