OPPORTUNISTIC MEDICAL DATA DELIVERY IN CHALLENGED ENVIRONMENTS

Nikolay Pavlovich Laptev, Shengsang Chai and Zhen Huang
{nlaptev, schai, zhen}@cs.ucla
Computer Science - University of California, Los Angeles

ABSTRACT

Delay Tolerant Networks have a great potential to be used in areas where there is poor connectivity. However the lack of adequate simulation results makes the choice of which DTN protocol to use unclear. In this paper we focused on designing and simulating a delay tolerant medical information delivery network that is representative of the scenarios where DTNs are to be used. We have simulated a variety of DTN protocols as well as created our own movement model. In conjunction we have also created our own general purpose DTN protocol compares favorably against other DTN protocols. Through extensive simulations, we show that MedProp, given sufficient density outperforms prioritized Epidemic and Prophet as well as the standard MaxProp protocols when it comes to overall message delivery (both high priority and low priority).

1 INTRODUCTION

The delay (or disruption) tolerant network is a network architecture where data is stored (possibly for long durations) and forwarded by intermediaries for eventual delivery due to the absence of an immediate end-to-end path at any given point in time. Current applications include content delivery to users of a public transportation system [1], battlefield communications [2], and wildlife information tracking [3]. Given the variety of situations in which a DTN is deployed, the design of the network must suit its purpose. We can imagine that in a multimedia content delivery system, delay may be a less important criterion than it is in a battlefield communications system. Clearly the routing protocol chosen for a particular scenario can affect those performance factors and is one of the most critical components of the network. Knowing this, we expected to find a comprehensive study of various DTN protocols used under realistic scenarios, but as we will later outline, we found most studies had one of the following characteristics:

a) The study assumes, without explanation, that an effective DTN protocol is being used in the system proposed.
b) The study designed or used a routing mechanism that is highly specific to the application
c) The study used a DTN where it is not necessary or the scenario used is unrealistic

Our goal is then to study the effects of implementing different general-purpose DTN routing protocols and settings in an environment where DTNs are likely to be deployed. We focus on creating a scenario that closely mimics possible real world applications. In particular, our motivation is to enhance DTN-based systems that provide medical services to rural areas. This scenario is a challenge for traditional methods of connectivity. For instance, dialup is both slow and prone to failure. Wireless mesh infrastructures, while more resilient, are too expensive. But CustoMed [6] shows that a robust medical information delivery network can be deployed by simply using PANs running on inexpensive BlueTooth or WiFi devices. In our study we also design improvements on existing general-purpose DTN protocols that is useful not only in the medical scenario but also in any application where the delivery rate of critical data is a major concern.

The rest of this paper is organized as follows: in 1.1 we present previous work, 1.2 presents several system components, section 2 examines a newly designed DTN protocol, 3 discusses the implementation of our movement model, 4 describes our simulation, finally 5 presents our results, and section 6 concludes the paper.

2 PREVIOUS WORK

While it is clearly infeasible to survey all works analyzing DTN protocols and related applications in this paper, we present samples of studies that manifest the aforementioned characteristics.

Absence of protocol evaluations - Often researchers are more interested in the hardware, social and legal issues, and high level applications of a DTN-based system. For example, [4] introduces a framework, CAM, that leverages social relationships of communities to deliver information in a rural environment. The authors discuss storage requirements, usage incentives, but assume that the appropriate DTN protocol will be chosen by network architects themselves. We observe that the majority of DTN related work falls in this category. Our study contributes evaluation results that actual implementers of these systems may use to guide their decision.

Highly specific routing protocols - Some studies compare general DTN protocols to ones designed especially for certain scenarios and demonstrate the performance improvement of that protocol under those tightly controlled conditions. In developing SWIM [5], researchers proposed an epidemic dissemination routing protocol where each node in the peer to peer network drops undelivered packets after a certain time. This time-to-live value is entirely dependent on the specific environment in which SWIM is used, and its parameters are derived through calculations and simulations. Suppose one must design a system built around a DTN. Even if the system is similar to the one described, say, another wildlife information tracking system, the network architects must decide whether epidemic routing is fit for this system and whether the method of calculating the time-to-live value is applicable given possibly different mobility models and terrains. Finally if the new scenario is a close enough match, the time-to-live must be exhaustively recomputed for optimal performance. In most cases, however, the system being designed may not be similar enough to those described in narrowly targeted studies, and the authors' contributions, such as the optimization of epidemic dissemination, may not be very useful in future work. For example, we see in ZebraNet [3], a deployed system with a similar purpose, that TTL was not a major design concern since the storage device could hold 110 days of data. Instead delivery ratio was highlighted as the more important goal. This problem is especially evident in model based approaches such as [7]. Our study analyzes general purpose routing protocols to abstract as much scenario specific information as possible without degrading the realism of the study. We do not claim the superiority of any one protocol under all circumstances.

Unrealistic scenarios - The need to perform comparison studies for different general-purpose DTN protocols have largely been neglected, but we are aware of two existing studies. In [8], the authors created a test scenario where Internet connectivity is provided to a remote village. The location choice and the purpose of the DTN are reasonable: organizations have already attempted to establish an asynchronously communication link between villages and large cities in South Africa. The use of satellites as well as motorbikes also reflects the data transportation tools available in rural environments. However, most studies stress that a realistic mobility model must be used for any simulation to be useful in practice (for example, in [9], using the wrong traffic model in VANETs created overly optimistic performance results). For the scenario presented, there was no indication of what mobility model was used for the villagers and therefore the representation of the actual data delivery system is not entirely convincing. The second scenario presented in the study shared similarities with [10] since both studies used a city-wide DTN. While DTNs can certainly be useful in cities, high tech metropolises such as San Francisco (used in [8]) and Helsinki (used in [10]) can be feasibly connected via traditional wireless networks. Google has indeed offered free WiFi service to San Francisco in 2005. Based on this observation, we feel that evaluating DTNs in large cities may not be useful in the long run. Instead, governments and corporations may take decades to connect developing world villages, making DTN protocol analysis important to ad hoc networks deployed in those environments. As we will discuss, our study uses a rural village scenario with improvements which makes it arguably more realistic than the one presented in [8].

2.1 SYSTEM COMPONENETS

We must also mention some important characteristics of the medical system which we simulate. We answer the following questions:

a) *What is the size of the data that is being disseminated?*
Clearly taking buffer size into consideration we would want to disseminate smaller files first which are popular. We only rarely disseminate large files, and the frequency with which we disseminate these files is based on their popularity. In our simulator we adjust this frequency through a configuration parameter.

b) *How much bandwidth is available?*
Based on the CustoMed project, our scenario will begin with a bandwidth capacity that of Bluetooth (which we will simulate in our scenario). We assume that a satellite will have high bandwidth relative to that of Bluetooth.

c) *What are the different classes of messages?*
In our simulation we classify messages based on priority level. We feel prioritization is essential to most DTNs. For example, we envision control messages, emergency notices to be a part of most DTN based systems.

d) *Predictable villagers vs unpredictable villagers?*
In our simulation we use both predictable and unpredictable villagers by modifying the movement model which those villagers use.

e) *What metrics are to be used for the analysis of DTNs?*
The metrics with which we measure the performance of DTN include: latency, message delivery ratio, hop-count. We feel that these are core metrics that should be applied to DTNs in general, not just to this scenario.

We use these ideas in the following detailed protocol and scenario design.

2.2 DTN PROTOCOL IMPLEMENTATION

2.2.1 Deterministic vs Stochastic, Single Copy vs Multi Copy

Many DTN protocols have been proposed recently and here we consider them for the implementation of our scenario. All existing DTN protocols fall into two categories: deterministic and stochastic. MED (Minimal Expected Delay) is one of the deterministic methods [8]. It estimates the delay on each link via the prediction of the future contacts and then uses the Dijkstra's shortest path algorithm to decide the best route from source to destination. The delay on each link is pre-computed and thus fixed. MEED [18] is a revised version of MED, which estimates the delay of each link based on past contact information. Both methods are practical only when the nodes move infrequently. Another protocol in this category is the data-ferry [19]. There are one or more nodes which serve as data-mules which travel between nodes and base stations. However, this protocol needs a network topology oracle and the movement pattern of the nodes in order to work. Furthermore, the reception range of the data-mule itself is limited. The only way to cover more nodes is to add more data-mules, which is ineffective and increases the overall cost.

The other category is stochastic routing in which there are two major methods: single-copy and multi-copy. For single copy, only one copy of the message exists in the entire DTN. The main problem in the single copy method is what the next hop to the destination is. Many utility functions were proposed to help make decision. These utility functions are based on the previous contact information such as time of contact and contact frequency. One utility function is to use the last encounter time where the message will be sent to the node which has the latest encounter time with the destination. However carefully one designs the utility function, the delay is large due to the slow propagation of the single message in the DTN network.

In multi-copy, there is more than one copy of message in the network. Message replication is used to increase the probability of delivery. The naive way is epidemic routing where the node sends the messages it has to every node it meets. This method is better than any other methods with regards to the delivery ratio; however, it is optimal only when buffer size is large. The performance degrades drastically if this condition is not met. In order to reduce duplication in the network while retaining the routing effectiveness, many methods have been proposed. 2-hop relay limits the number of hops the messages travel. The source (hop=2) sends the message to every node it meet. These nodes (hop=1) can only send the messages to the destination. Using the same idea, 3-hop and k-hop methods were proposed. A variation of the 2-hop relay is spray-and-wait [23], which limits the number of transmissions of a message per node to some number L. There are always L copies of the message in the network.

There are many researchers working on DTNs in rural environments. Interestingly, [20] introduced a protocol called DTLSR which specifically addresses the requirements of this setting. It is a variation of MEED, which is a link-state-based deterministic routing protocol. The link cost is computed based on previous contact information. Whenever the link state changes over a certain window size, the source node will flood the change to every other node in the network. The way it works is that nodes simply exchange their link-state-change vectors when they meet. Eventually the link-state change information will be disseminated over the whole network. The only difference between DTLSR and MEED is that they divided the entire network into several administrative areas to reduce the overall number of times that messages are exchanged. As stated before, deterministic routing protocols are useful only when the nodes move infrequently.

In the medical scenario, the messages about a person's condition can be prioritized into three categories: high, medium and low. The messages with high priority are generated under emergency conditions and must be sent as soon as possible given any opportunity. A critical observation is that high priority messages are sent relatively infrequently compared to those with lower priorities. Obviously, the deterministic routing protocol is not appropriate, since the nodes in our scenario move frequently. The single-copy method is not suitable as well since it is less effective than the multi-copy method in the rate of packet dissemination. The messages with high priority cannot be sent at first available opportunity.

High priority messages must be disseminated to the whole network as soon as possible, meaning that a large portion of the nodes would hold these messages in their buffer. Any node with the message can send it if they meet the destination node. Naive approaches, such as flooding and spray-and-wait do not consider the message priority and just disseminate every message given the chance to do so. They perform poorly on several metrics even though they achieve high delivery ratios as we will demonstrate. Furthermore, the performance degradation given a limited buffer size is also a concern for most networks. With our medical scenario (and in fact most scenarios in general), more realistic protocols should be considered.

2.2.2 MedProp: A Multi-Copy, Utility Based Stochastic Protocol

In order to meet the requirements of delivering medical information in a rural environment with limited resources, we adopted a utility-function-based multi-copy stochastic routing protocol called MedProp. With the utility function, we can send the messages in order of their priorities. With the multi-copy characteristic, we can increase the rate of dissemination. It is called multi-copy because every node sends all the messages in its buffer to all its neighbors. However, it is different from flooding because each <message, connection> pair is sent in an order that is computed by a utility function. Meanwhile, with the intrinsic flooding characteristic and limited buffer size, we must also design the policy for deciding which messages are to be dropped from the buffer when it is full.

Borrowing a similar idea from MaxProp [21], the design of the utility function is as follows: Let P_i^j denote the probability of node i meeting with node j in a DTN with s nodes. Initially, $P_i^1 = P_i^2 = P_i^3 \ldots = P_i^{s-1} = 1/(s-1)$. When any node i meets with another node j, set $P_i^{j'} = P_i^j + 1$, and renormalize the probability. With this method, each node maintains a local table with the probability of meeting with every other node in the network. Every time two nodes meet, they exchange their local tables with each other. In this way, each node has a global table storing the meeting probability of any two nodes in the network. We denote the cost from node i to node j $cost(i,j) = 1 - \sum_{x=i}^{j} P_x^{x+1}$. With this global table, we can compute the minimum cost from node i to node j using Dijkstra's algorithm.

For example, the node i has three messages $< m_1, m_2, m_3 >$ in its buffer and the two connections $< c_1, c_2 >$. There are 6 pairs to be sent, $< m_1, c_1 >$, $< m_1, c_2 >$, $< m_2, c_1 >$, $< m_2, c_2 >$, $< m_3, c_1 >$, $< m_3, c_2 >$ respectively. The pair $< m_i, c_j >$ denotes sending message i to node j. Medical messages, as mentioned, have different priorities, and the order is specified in the following way:

1. The messages with high priority are always sent first. They are sent to every currently connected node.
2. Each message keeps the number of hops it has already traveled. In our case, we set the hops to 5. The messages which travelled less than 5 hops are sent right after the messages with high priority. The newly-generated messages are given a head start.
3. For each of the rest of the pairs $< m_i, c_j >$, we compute the minimum cost from node c_j to the destination of message m_i, $\min\_cost(c_j, Dest_i)$. They are sorted and sent in descending order.

This way, the messages with high priority, which are generated relatively infrequently, are disseminated quickly to nearly every possible node in the whole network and thus reach the destination as soon as possible. Meanwhile, we find the best routes for the other messages. They are sent along these routes at the first opportunity so the latency is minimized. Instead of sending the messages along the best route, we also send them to other connected nodes in order to increase the probability of delivery.

We have not yet addressed the buffer and treatment of messages with medium and low priorities. In practice, the buffer size is limited. Due to the intrinsic flooding characteristic of our approach, we need to design the message-dropping policy for the buffer. A naïve approach is FIFO (first in, first out), which is easy to implement. However, it is not practical in our scenario because the messages with high priority might be dropped from the buffer and never have the chance to be sent. Therefore, the buffer-drop policy must consider priority. Doing so, we also have to consider the messages with medium and low priority. Intuitively, the messages with low priority should be dropped first. However, we found out during the simulation that these messages with low priority will rarely have a chance to be sent if there is a fairly large number of messages in the network. The reason is that whenever a message should be dropped, the messages with low priority may always be dropped. In order to avoid this situation, we specify instead that messages with medium and low priorities are dropped with 30% and 70% probability respectively. The organization of the MedProp buffer is as follows:

- The messages with high priority are always on the top.
- The messages which travelled less than 5 hops are right below.
- For the other messages, we divide them into two categories: messages with medium priority and messages with low priority. For messages in each category, we compute the minimum cost from the current node to the destination of message m_i, which is $\min\_cost(current\_node, Dest_i)$. We put them in descending order.
- The probability of being dropped for the messages with medium and low priorities: 30% and 70%.
- Messages are dropped starting from the bottom of the buffer.

Visually, we represent the buffer architecture in Figure 1.

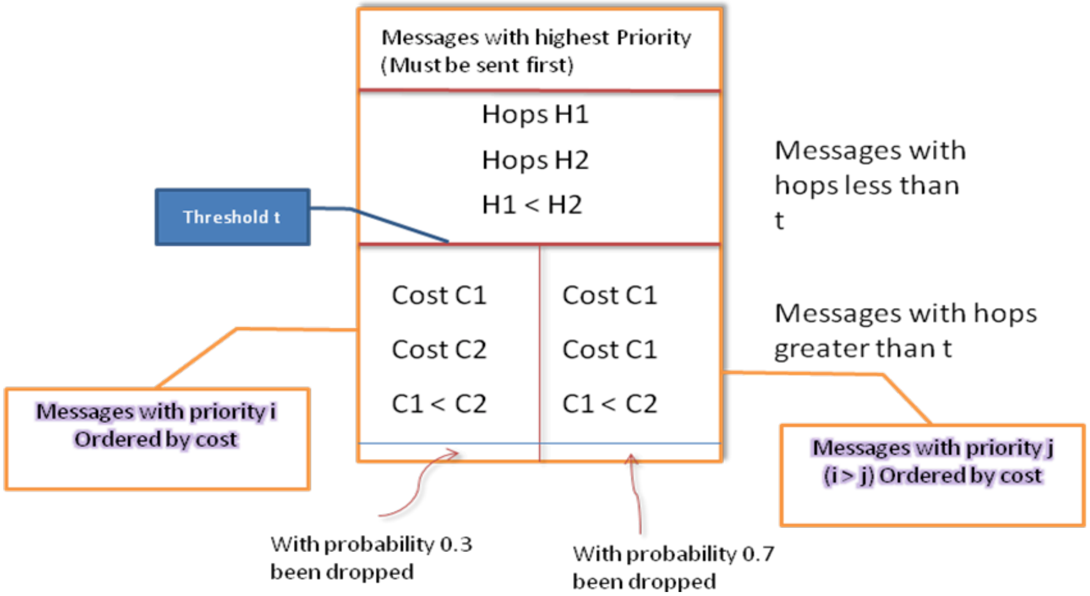

Figure 1: The MedProp buffer architecture

2.3 SCENARIO CREATION

As previously mentioned, we find the rural connectivity scenario both interesting and worthy of investigation. Not only does the environment provide a stress test for DTN protocols, a protocol that performs well in this scenario should also do well in other scenarios where DTNs are deployed since they should share similarities such as low node density, lack of infrastructure, and the need for prioritized data delivery. Delivering medical information from remote village kiosks to hospitals in large cities has garnered some attention from the scientific community as seen in [4] and [11]. Providing medical services is seen as a step in the development of regions, and the social motives of rural development through technology are described in [12]. The lack of communications infrastructure, the low population density and the poor economic conditions in general make DTNs a natural fit for providing connectivity to these areas. In fact, the system described in [4] proposes DTNs as its underlying network architecture. Current studies in medical information delivery, however, have so far ignored how the protocol selection and settings can affect system performance.

In formulating our scenario we chose Bujumbura, the capital of Burundi, Africa as the anchor city. Being the economic center of Burundi, some medical and communications infrastructure is expected to exist. Our central kiosk is located there and is the sink for all messages. Two villages, Gatumba and Bubanza are within a reasonable distance such that medical workers can routinely travel there via automobile. This is an important consideration since villagers are not expected to travel to cities frequently, and consequently medical vehicles are the primary mode by which information will be delivered to hospitals from the villages. We also note that due to the political and economic situation of Burundi, we do not anticipate telecommunications development efforts to reach rural areas of the country for some time, making DTN based networks the only viable option for the foreseeable future. Additionally, AIDS is widespread in the country, creating the need for rural medical services. OpenStreetMaps [13] provided sufficient transportation information for the region such that we could model the area using OpenJUMP [14], an open source GIS program. We mention that we purposely avoided highly populated countries because DTN connectivity is most challenging where nodes are sparsely distributed. Performing our simulations using areas with high population density, such as in Bangladesh, would place less performance stress on most DTN protocols given sufficient storage (we confirm this in our simulation results).

Next we consider the node types and their mobility. Kiosks are stationary and act as information repositories. Villagers are slow moving nodes and are the sources of data. Cars are faster moving nodes and transport data only. We also employ a satellite node in our simulations as [17] has shown that developing nations are willing to deploy satellites for special civilian purposes. Satellites are special nodes in that their transmission capacity increases and decreases over a set period during the course of the simulation.

With regards to node mobility, cars and some villagers use the Geographically Restricted set of mobility models [15]. The movement of these nodes is constrained by the roads placed on map, that is, they cannot move where there are no roads. The basis for this type of mobility model is clear for cars. For the constrained villagers, we can assume that buildings prevent them from moving freely on the map and they frequently move to popular areas, such as markets, that are connected by roads, therefore limiting them to following certain paths. Other villagers are not limited by roads as they maybe farmers or hunters who roam the area. Within the Geographically Restricted and Non-restricted models, the randomness of the movement can vary. Cars, the medical vehicles in particular, can follow a completely pre-determined path using the Map Route movement model. For humans, Random Walk is sometimes used in simulations, but the idea of a node suddenly changing its direction for no purpose is unrealistic. Random Waypoint is more common in modeling MANETs, and it is a candidate model for our simulation since it gives villagers the appearance of having a purpose in their movement. However as noted in [16], Random Waypoint does not capture group movement seen in human behavior. This led us to implement a Geographically Restricted and Non-restricted versions of the RendezVous model [2] with some improvements to its realism. The standard RendezVous model allows groups of nodes to select waypoints on the map, simulating situations when a group has a common interest at certain points in time. An example of this type of behavior is when some friends had previously agreed to meet at a predetermined location and time. In our implementation, we additionally modeled the fact that sometimes members of the groups can act individually in their own interest, disregarding the actions of the other members of the group. For example, a person visits the local market by himself. Also the standard RendezVous model allow individuals to disassociate themselves from groups with high probability, but normally humans form closely knit cliques and tend to stay in them. Hence we keep an individual's group affiliation during the simulation. Since our RendezVous implementation captures both the individual and clique behavior of real humans, we primary use this movement model in our simulations.

3 **SIMULATION**

Our scenario includes the anchor city and two villages with some number of villagers moving within them. There is also a kiosk with equal length roads from both villages to the anchor city, which contains the data sink. We also have two medical vehicles which follows a route between the villages. Villagers generate messages, and vehicles, when near the village, download these messages and deliver them back to the central kiosk. The vehicles in our scenario use BlueTooth to connect to the medical devices carried by villagers and WiFi to communicate with kiosks and other vehicles. A satellite link provides variable connectivity over the entire simulation area.

We extended ONE [22], an open source DTN simulator written in Java, with our customized node types, the mobility model that we have described, the MedProp protocol (and other protocols used in the comparison), and the custom maps. The default scenario is shown running in Figure 2. We have used various settings to simulate our scenario. Table 1 presents of some of the base settings we have used:

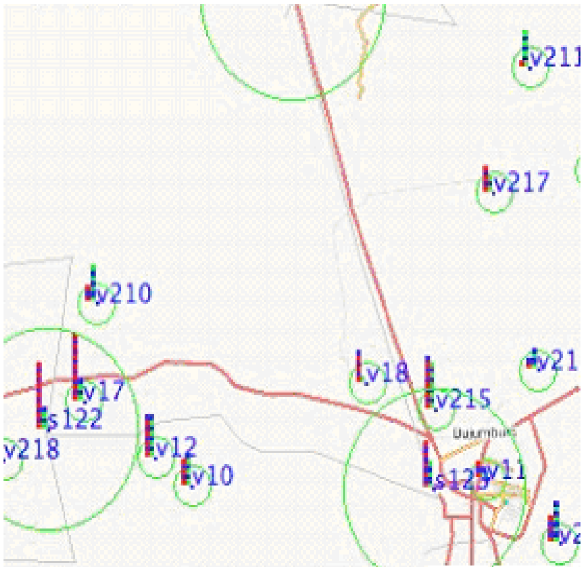

Figure 2: Image of default simulation in ONE.

Name of attribute	*Default Value*	*Variable?*
Buffer Size	40MB	No – Constant
Number of hosts	5 nodes in each village	Yes
Transmit Range	20m (BT), 100m (WiFi)	No – Constant
Transmit Speed	250K/s	No – Constant
Movement Model	Random Waypoint	Yes
Wait time (before transferring)	0,120ms	No – Constant
Speed	0.5-1m/s	Yes

Router	Customized RendezVous	No – Constant
Active times (when nodes can send msg)	Length of scenario	No – Constant
Message TTL (Duration during which nodes can create messages)	60 minutes	No – Constant

Table 1: Settings used

We start with the basic case without using any message prioritization and using a standard Epidemic routing DTN protocol built in ONE. The movement model which we use in this basic case is Random Waypoint. With each run, we vary the node density (as measured by nodes per village), node speed, and the protocol used. Note that while we compared MedProp to several protocols (see Figure 9), we chose to display the Epidemic comparison as the differences are most evident here. Also we gave the nodes a medium capacity buffer; otherwise naïve multi-copy methods have no way of being competitive with MedProp.

3.1 EXPERIMENTAL RESULTS

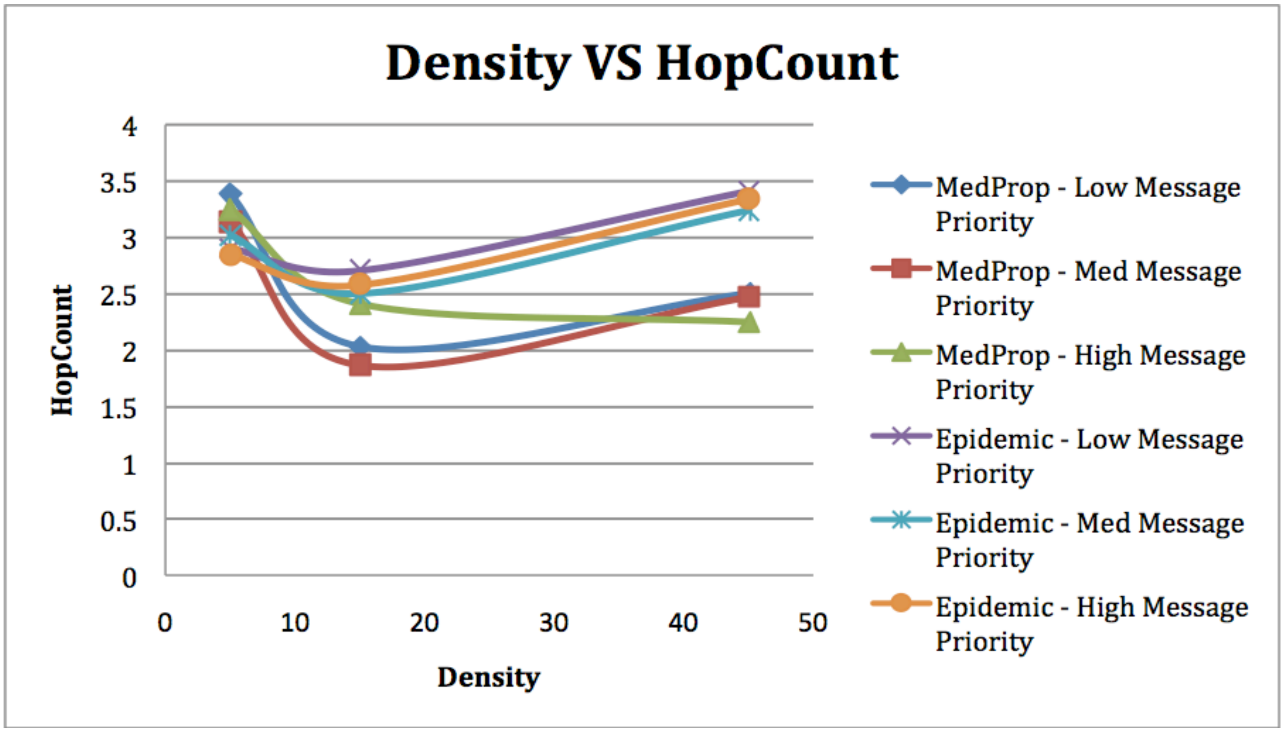

Figure 3: Density vs Hop Count

In Figure 3, we can see that initially the hop count decreases with increasing density but later increases because the limited buffer size forces messages to be dropped. For the same reason, it is also clear that MedProp decreases the hop count relative to Epidemic which leads to a significantly less network overhead. High priority messages routed using MedProp have significantly better hop count performance than any other type of message using either Epidemic or MedProp. This is due to the fact that high priority messages in MedProp sit on top of the buffer, which allows these types of messages to be forwarded aggressively at the expense of other types of messages. However, both medium and low priority messages are routed more efficiently using MedProp than using Epidemic.

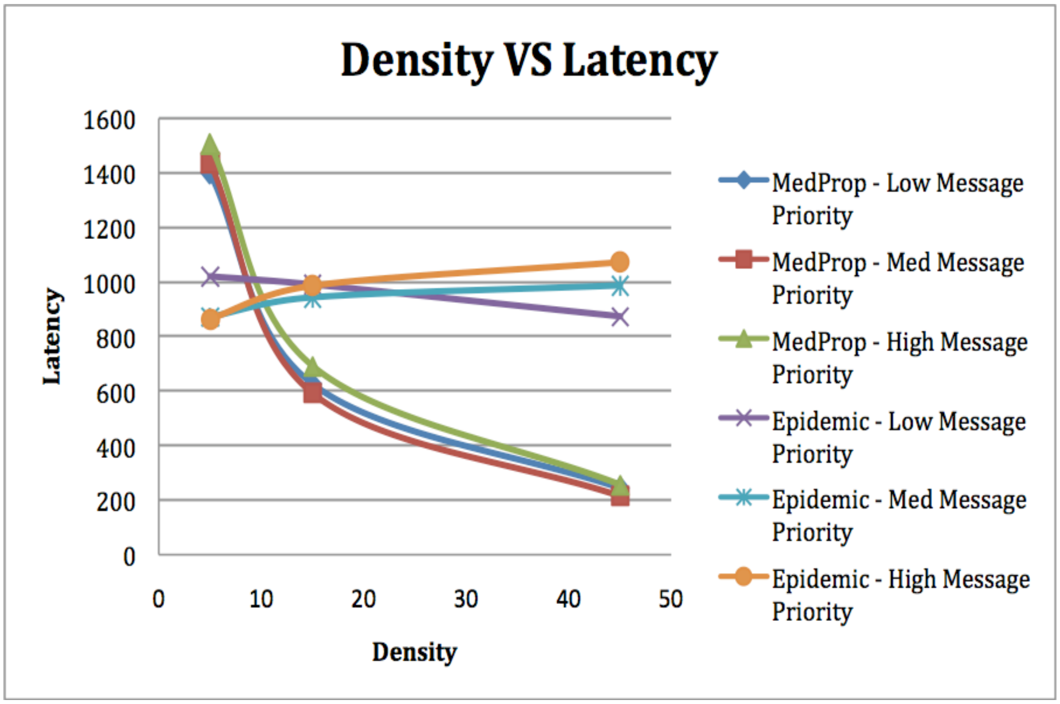

Figure 4: Density vs Latency

Figure 4 shows that with increasing density MedProp actually reduces latency whereas Epidemic's latency stays relatively constant. Also there is a large difference in Epidemic's latency between the messages of different priorities while the difference is negligible in MedProp. Finally, we notice that MedProp actually performs poorly when the density is very low but improves rapidly as density increases. The effects of density on MedProp's latency performance are due to the fact that it does not use flooding, as opposed to Epidemic. Thus the reliability of the local routing tables is poor when density is low, causing greater latency to be experienced. However the benefit of MedProp becomes evident fairly quickly.

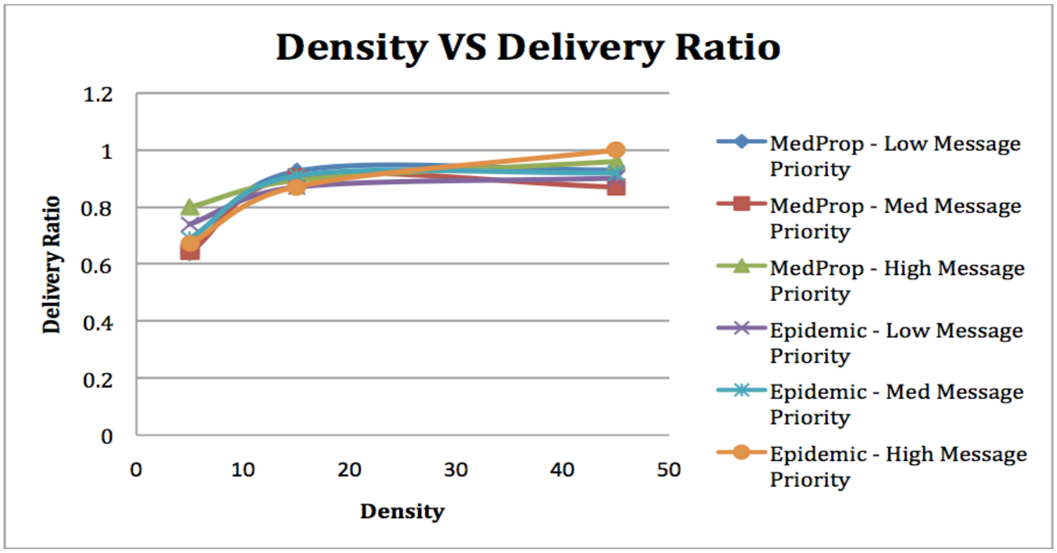

Figure 5: Density vs Delivery ratio

Figure 5 shows that with low density, delivery ratio is best with MedProp but at high density, Epidemic is best. Here the relatively large buffer size we use allowed Epidemic to perform nearly optimally in terms of delivery ratio. The lower priority messages are frequently dropped but will eventually reach the destination, and higher density helps with this type of propagation. If the buffer size was smaller, MedProp will outperform Epidemic at high density due to it having a more efficient message delivery strategy. Furthermore, we consider the low density simulation to be more reflective of actual DTN scenarios.

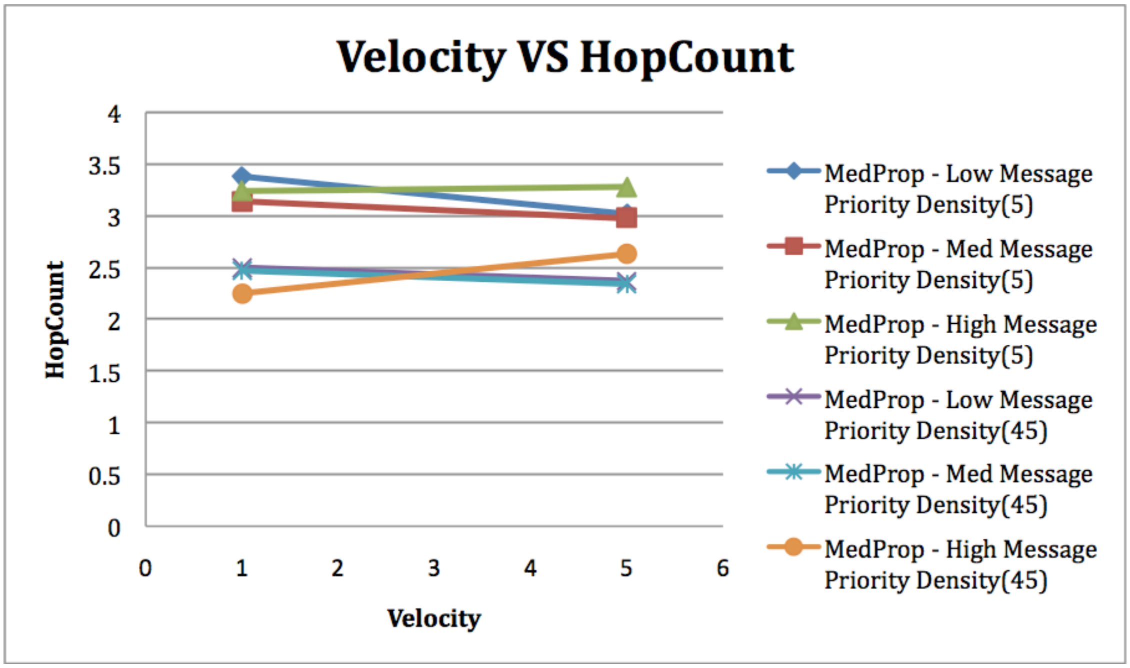

Figure 6: Velocity VS Hop Count

In Figure 6 we see that MedProp performs better in terms of hop count for any given velocity tested. Also in both MedProp and Epidemic, hop count for the high priority messages increases and medium/low priority messages' hop counts decrease for increasing velocity due to the higher contact rate. This is because the flooding of high priority messages in a high contact rate environment uses up the most constrained resource (in this case, time is most constrained). This suggests that it's best to decrease the size of high priority messages because these messages tend to flood the network. In realistic scenarios it is feasible as emergency or control messages can be encoded in less than 1KB. Intuitively with high node density we would have a lesser hop count, however, as explained above, the flooding of high priority messages forces other messages to simply be dropped when more contacts are made with increasing velocity.

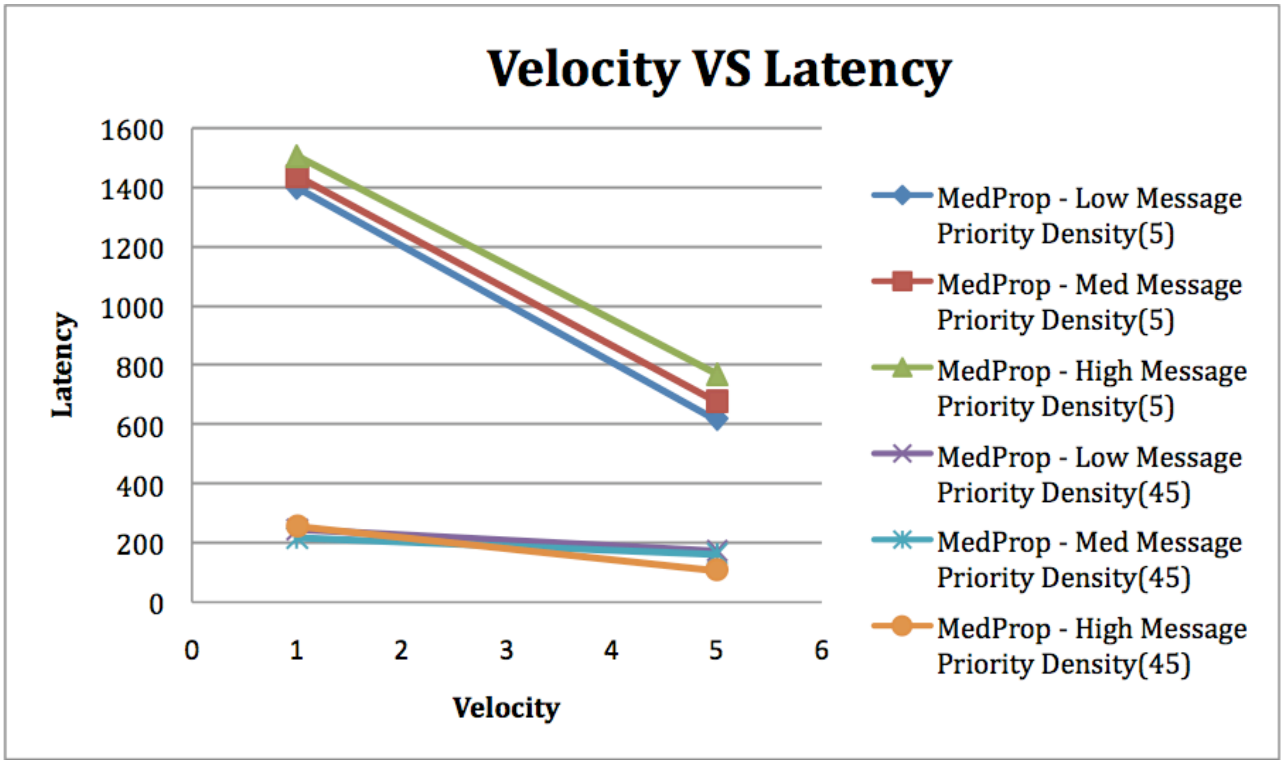

Figure 7: Velocity VS Latency

Figure 7 shows that MedProp's latency performance improves with greater velocity and is especially evident when density is low. This is as one would expect, since high velocity produced lesser latency due to a higher contact frequency between nodes. We note that when density is high, velocity's effect is almost negligible because such a low latency has already been achieved at low velocities and may not be improved much.

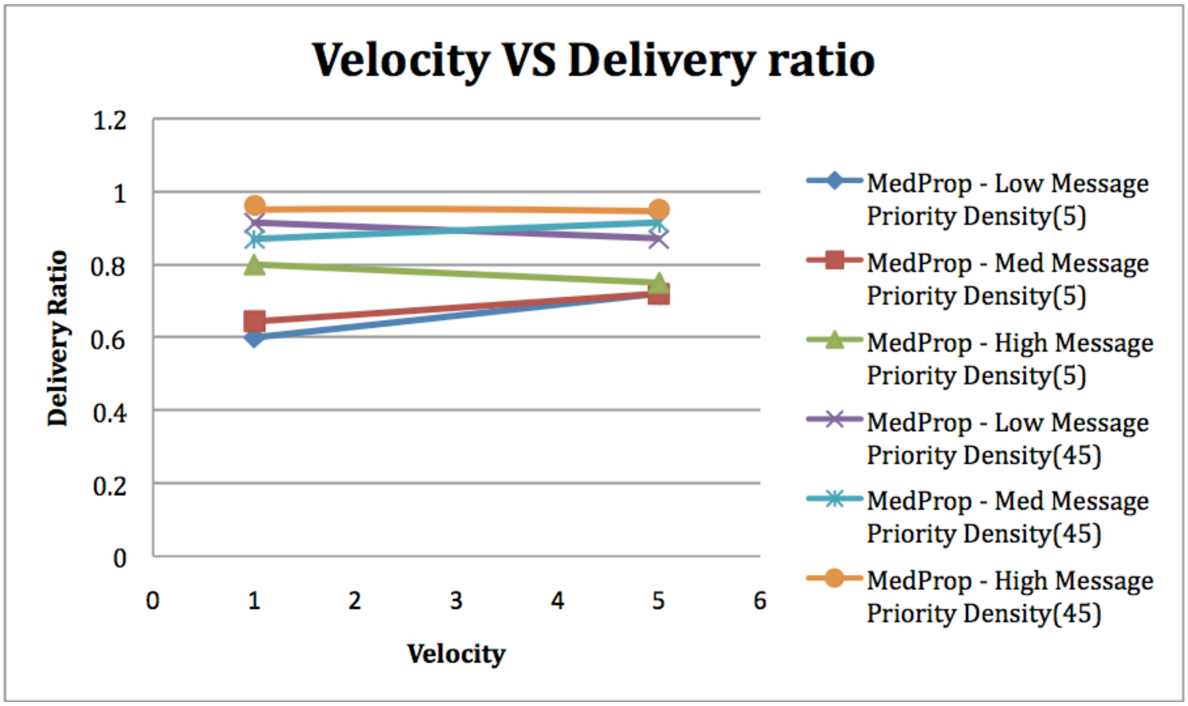

Figure 8: Velocity VS Delivery Ratio

In Figure 8 we can see that at high density, increasing velocity results in no significant difference in the delivery ratio given that our ample buffer size facilitates the eventual delivery of almost all data regardless of priority. With low density however, higher velocity improves the medium and low priority message delivery ratios at the expense of a slight decrease in the high priority message delivery ratio. With higher mobility we expected that the performance of the three priority classes would converge as shown because nodes have more frequent contacts with each other, decreasing the hop count (as seen in Figure 6), therefore allowing eventual delivery of the message since a very small percentage of the messages generated must be dropped from the network. This also illustrates the effectiveness of our multi-copy design as opposed to single-copy.

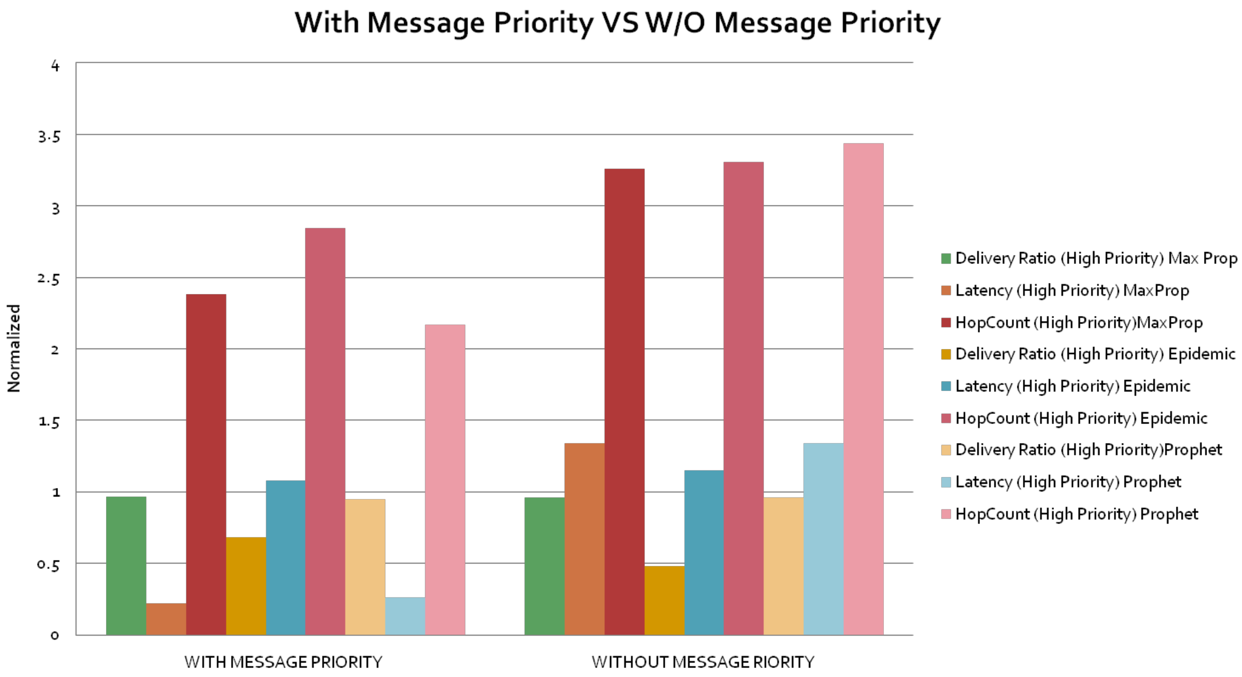

Figure 9: Messge Priority and Delivery Performence

In Figure 9 we can see that on every measure, delivery performance of high priority messages improves with priority implementation. This graph simply summarizes similar results that we obtained from comparisons with other protocols not shown in the above figures.

4 CONCLUSION AND FUTURE WORK

In summary we have created a realistic scenario modeling a medical information delivery in a developing world environment, which is an emerging field where DTNs are currently being deployed. We have identified a realistic network architecture for this environment, from the devices carried by each node to the types of nodes available. We designed a realistic movement model that features both group and individual behavior. We designed a DTN protocol, MedProp, based on MaxProp, by enhancing the previous protocol with high/medium/low priority message support, a unique buffer architecture with an accompanying message drop policy. We found that given sufficient density MedProp outperforms prioritized versions of Epidemic and Prophet as well as the standard MaxProp protocol using latency, hop count and delivery ratio as our performance metrics. We envision some future work may include the simulation of addition scenarios, considering more performance metrics such as amount of overhead, and further improvements in our protocol.

ACKNOWLEDGEMENTS

The writers would like to acknowledge the contributions of Uichin Lee and Daeki Cho to the research of this paper.

REFERENCES

[1] M. Papadopouli, H. Schulzrinne - Seven Degrees of Separation in Mobile Ad Hoc Networks
[2] R. Nichols, A. Hammons Jr. - Performance of DTN-Based Free-Space Optical Networks with Mobility
[3] P. Juang, H. Oki, et al - Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet
[4] B. DeRenzi, Y. Anokwa, et al - Reliable data collection in highly disconnected environments using mobile phones
[5] T. Small, Z. Haas - The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm
[6] CustoMed - http://er.cs.ucla.edu/CustoMed/
[7] C. Becker, G. Schiele. - New Mechanisms for Routing in Ad Hoc Networks Through World Models
[8] S. Jain, K. Fall, et al - Routing in a Delay Tolerant Network
[9] G. Marfa, G. Pau, et al - VANET: On Mobility Scenarios and Urban Infrastructure. A Case Study
[10] J. Kim, S. Han - Routing Protocols in Delay/Disruption Tolerant Network
[11] J. Donner - Innovations in Mobile-Based Public Health Information Systems in the Developing World: An example from Rwanda
[12] A. Pentland, R. Fletcher, et al - DakNet: Rethinking Connectivity in Developing Nations
[13] OpenStreetMaps – http://www.openstreetmaps.org
[14] OpenJUMP – http://www.openjump.org
[15] F. Bai, A. Helmy - A Survey of Mobility Models in Wireless Adhoc Networks
[16] A. Nandan, S. Das, et al - AdTorrent: Digital Billboards for Vehicular Networks
[17] SSTL - News Archive - http://www.sstl.co.uk/index.php?loc=27
[18] E. Jones, L. Li, et al - Practical Routing in Delay-Tolerant Networks
[19] D. Jea, A. Somasundara - Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks
[20] M. Demmer, K. Fall - DTLSR: Delay Tolerant Routing for Developing Regions
[21] J. Burgess, B. Gallagher, et al - MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks
[22] The ONE simulator - http://www.netlab.hut.fi/~jo/dtn/index.html#one
[23] T. Spyropoulos, K. Psounis, et al - Spray and Wait: an Effective Routing Scheme for Intermittently Connected Mobile Networks