2012 26th International Conference on Advanced Information Networking and Applications Workshops

Clustering Between Data Mules for Better Message Delievery

Xoluqobo Mkhwanazi
Computer Science Department
University of Cape Town
Cape Town, South Africa
e-mail: xmkhwanazi@cs.uct.ac.za

Hanh Le
Computer Science Department
University of Cape Town
Cape Town, South Africa
e-mail: hanh@cs.uct.ac.za

Edwin Blake
Computer Science Department
University of Cape Town
Cape Town, South Africa
e-mail: edwin@cs.uct.ac.za

***Abstract*— Traditionally data cannot be delivered between isolated networks. However by using a data mule which is a combination between an electronic device and a mobile entity we can connect isolated networks. However, the main problem is that a long delay is experience and most of the time a large amount of undeliverable. In this paper we propose an algorithm that increases message delivery by clustering data mules in order to find a more reliable path between the sender and the receiver. Instead of data remaining on the same mule when travelling from one network to another, our algorithm allows mules to transfer data to nearby mules arriving to the destination network sooner. Using movement traces for the data mules for rural-like areas we compared two algorithms for a different number of nodes in the network and different communication ranges for each node. The preliminary results show that our proposed algorithm increases the network performance.**

Keyword-algorithms, latency, mule communication, data mules, isolated networks

I. INTRODUCTION

Simple ideas make a world of a difference. One of these ideas is the data MULE which stands for Mobile Ubiquitous LAN Extensions [12]. It is a combination of any data carrier and mobile entity. This means a data mule could be a person with usb stick or a laptop powered by a bus. The data mule has the purpose of picking up data one access point, buffering it and dropping it off at another point [12].

The data mule works in three-layer architecture [12] shown in Figure 1. The first tier consisted of nodes e.g. sensor nodes, laptops and cell phones at the bottom of the figure, the second tier consisted of the data mules and finally the last tier consisted base stations. The first and third tier was either sender or receiver, and the second tier was the space which only the data mule can travel.

Data mules can be used in various areas e.g. in wireless sensor networks (WSNs) and isolated networks. The isolated networks are defined to be networks that are out of range to each other and this may be due to terrain or lack of infrastructure. We found that by using mules to connect isolated networks or sensor nodes to base stations there was a long latency experienced. The main reason for this latency is because the user of the mule has no control on the direction or speed of the mule [12,16].

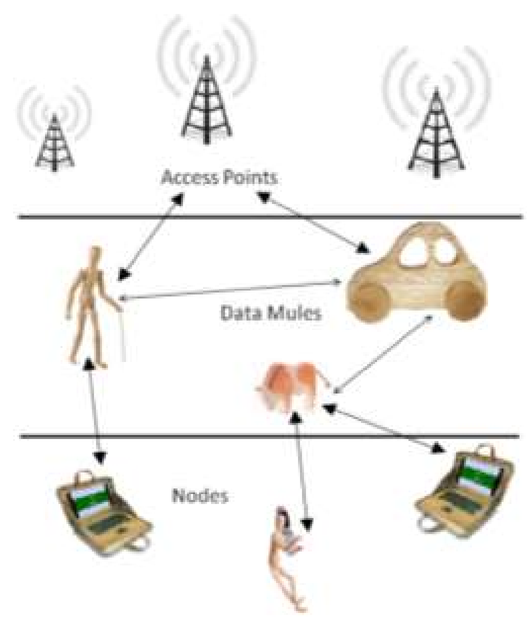

Figure 1: The MULEs three tier architecture: first tier – nodes, second tier – data mules and third tier- access points.

For our research we aim to reduce latency experience when using data mules to connect isolated networks in rural areas. In order to do this, one can relay data between multiple data mules. We aim to develop an algorithm that efficiently handles the way multiple data mules communicate with each other. This paper proposes clustering multiple data mules when in range of each other for better communication between the data mules. As a result more data may be successfully sent from the first tier and second tier in a shorter period of time. In this paper, we will briefly discuss related work, present a new algorithm (data mule inter-communication algorithm) that allows for exchange of messages between data mules, discuss our experiment and results, finally conclusion and further work.

II. RELATED WORK

In this section, we shall discuss works related to mules. By using mules to connect isolated networks, a delay tolerated network (DTN) is formed. DTN's are disconnected networks, where transmission of information across the network is subject to long latency sometimes measured in

978-0-7695-4652-0/12 $26.00 © 2012 IEEE
DOI 10.1109/WAINA.2012.199

days or hours [9,11]. A good example of a DTN that uses data mule for connectivity is the DAKNet project in India.

A. DAKNet

In the DAKNet project [9], the mules are used to connect surrounding isolated village networks. The mule is a bus equipped with a laptop charged by the bus, and travels on its designated route from the city to the village. Each village has a Wi-Fi-enabled kiosk and when the bus is in range it picks up the data (e.g. requested internet pages) and buffers it. When the bus arrives in the area that has access to the internet i.e. a HUB, the pages on the bus, are uploaded and will be collected later when available, and the data will be delivered to the destination villages.

In a report about empowering India, it was reported that in the DAKNet project most messages were delivered within 6 hours [1]. This is clearly an indication of latency that is experienced.

B. VANets

In order for data to hop from one mule to another, the mules have to be connected to communicate. When the mules connect to each they form an ad hoc network [10] furthermore if we assume the mules are partially vehicular they form a vehicular ad hoc network (VANets) [6,7]. VANets are specialized mobile ad hoc networks (MANets). A MANet is a type of ad hoc network that changes locations and re-configures itself. The nodes in the network are mobile and use wireless connections to connect to various networks. VANets are distributed, self-organizing communication networks built up from travelling vehicles and a subset of MANets, mobile ad hoc networks. This means an algorithm designed for MANet should be able to work for a VANet with a few changes to accommodate the network differences.

C. Optimal Relay Path Algorithm

An algorithm that works well in a MANet is the optimal relay path algorithm (ORP) [10]. In brief, the ORP algorithm uses the strong assumption that in an ad hoc network of mobile computers (mules) the trajectory of each host is approximately known. ORP calculates the shortest path from one mule to another. It uses the current direction, speed as well as the path of the mule in order to calculate the time that the sender mule and receiver mule are in range. It also calculates whether or not sending via the intermediate mule/s between the sender and receiver will be faster.

Some of the disadvantages with the algorithm are that every mule in the network needs to know the current position and direction of all of the mules in order to calculate the optimal path. As a result an error is incurred during the calculation of when to send a message and this means that at the time that a sender mule sends a message to the intermediate mule, or the receiver mule, the mule may be out of range.

III. DATA MULE INTER-COMMUNICATION ALGORITHM

We propose a new algorithm called data mule inter-communication (DMI) which is an integration of clustered mules and ORP. It handles communication between multiple mules that are clustered and makes use of the ORP algorithm for mules that are not clustered. The aim is to improve the network performance.

One of the key factors that affect the design of the algorithm for DMI is the movements of the mule. The mule movements have a predictable pattern and the reasoning is that the mule would be a vehicle e.g. a bus or taxi equipped with a wireless device to transfer data. The vehicle has schedule route it travels on however, the speed of the vehicle changes unpredictably.

Keeping this in mind, we shall discuss the main components that form our algorithm namely the cluster creation and maintenance and the communication protocols used.

A. Cluster Creation and Mainteance

Clusters are created when the mules are within each other's range. In order for a mule to join an existing cluster, it must be within range of a single cluster member.

Take for example mule A and mule B are within range to each other and forms a cluster. If mule C is within range of either mule A or mule B, it also belongs to that cluster. In each cluster there is a head mule, the head mule knows the communication route to each mule from itself. To find a communication route the head mule sends a query to its neighbors asking if they are neighbors with the destination node. Once the route been discovered, a message with the communication route is sent back to the head node.

The head mule is the most connected mule, namely the mule that has the shortest path to all the other mules in the cluster and all the cluster members are connected. This allows for better and accurate exchange of data.

Since the mules are constantly moving the clusters are generally temporary and cluster members change often. To avoid broken paths, a mule will send a message if a neighbor moves away.

B. Communication Protocols

Two communication protocols are used to form the DMI algorithm. One to handle communication within the clusters (intra-cluster) and another to handle communication with mules outside of a cluster (inter-cluster).

For inter-cluster communication protocol we use the ORP algorithm. This means is applicable when the sender and receiver mules are not in the same cluster. The sender mule uses the ORP algorithm to find the optimal path and the optimal time to send the message to the intermediate/receiver mule.

For intra-cluster communication we derived a protocol from a combination of several existing ad hoc wireless networks [2,3,15]. From our cluster creation, the most connected mule is the chosen head mule. When the sender mule is not the head mule and wants to send a message, it first checks if the receiver mule is a neighbor. If the receiver mule is not a neighbor, it sends the message to the head mule. It is important to note that when sending a message to the head node if the receiver mule is encountered on the way, the message is delivered there. If the head mule is the sender mule, it sends the message directly to the destination mule.

IV. EXPERIMENTAL TEST

Our proposed algorithm (DMI) was tested against the established algorithm (ORP). We consider the relative performance of the algorithms. We report factors: the communication range (referred to as node range), re-clustering (when using the clustering algorithm) and the percentage of successful messages.

A. Mobility Trace Collection

The tests are designed to evaluate the performance of the DMI algorithm in a rural-like environment. The rural area that we found that we could easily obtain node movements was in Texas, USA ("roustabout camp"). From various surveys [6,7,8] we chose the VANet mobility software that simulated vehicle movements realistically. It can also cope with stopping and starting of vehicular movements. We used open-source software VANet MobiSim that is based on CANU [6,13], to obtain mobility movements that were realistic as possible.

Using the VANet MobiSim, we collected mobility traces for 25 nodes; we also specified the size of the network to be 100km by 100km with Roustabout Camp as the centre of the map and the average moving speed of each node to be between 50km/h to 100km/h. These isolated networks present the stops for a taxi or bus. It is good that the mobility traces for the nodes include such parameters of random stops, as this further depicts the realistic movements of a bus or taxi. Furthermore, node paths are used from the traces collected. For each node, there is a known path. Simulation runs for 10 000s.

B. Test Set Up

We run the test for each algorithm with a different number of nodes in the network. For each network, each node has different communication ranges. The number of nodes varies from 5, 10, 15, 20 and 25 nodes. The ranges are 10m, 25m, 50m, 75m and 100m. For the clustering algorithm we also test the effect of changing how often clustering must occur, we looked at re-clustering every 20s, 50s and 100s. For each test, we used the same three message data sets. In the message data set, we had addresses for 100 messages for each of the different number of nodes in the network.

Another important factor was that nodes maintain the same trace movements. All this will ensure that the comparison is fair reflection of the ORP algorithm vs. the clustering algorithm. It will also show to what extent does the clustering have effect of the ORP algorithm, and possible what changes we have to make. We graph the ORP vs. the clustering algorithm over different ranges.

C. Results

The preliminary results are averaged from the three data sets. Our first sets of graphs we analyze the clustering for the DMI algorithm. We will first analyze the average number of clusters in the network versus the node range for all the node sets, and the average number of nodes in the cluster versus the number of node range for the different number of nodes in the network. This allows us to see how the number of clusters in the network and the average cluster size affects the performance of the algorithm.

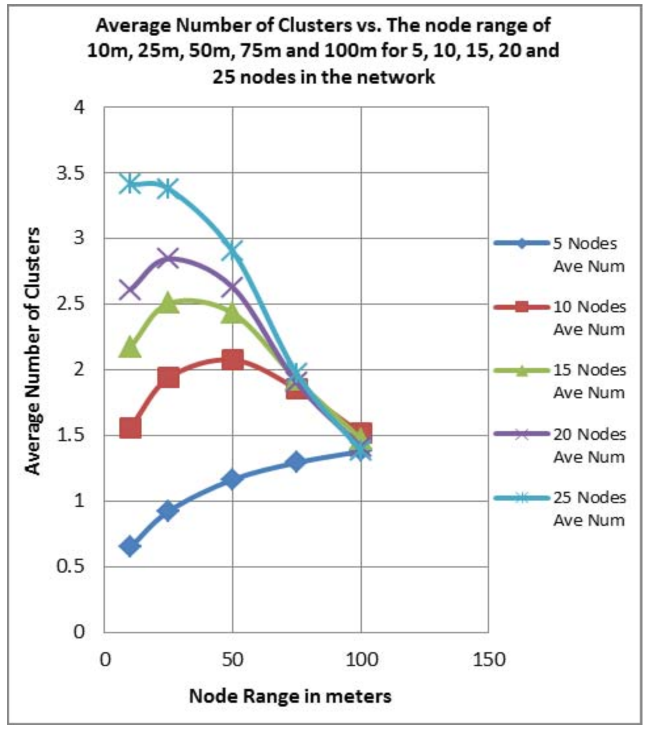

Figure 2: The average number of clusters in the network versus the different node communication ranges (10m, 25m, 50m, 75m and 100m) for the different number of nodes (5, 10, 15, 20 and 25 nodes)

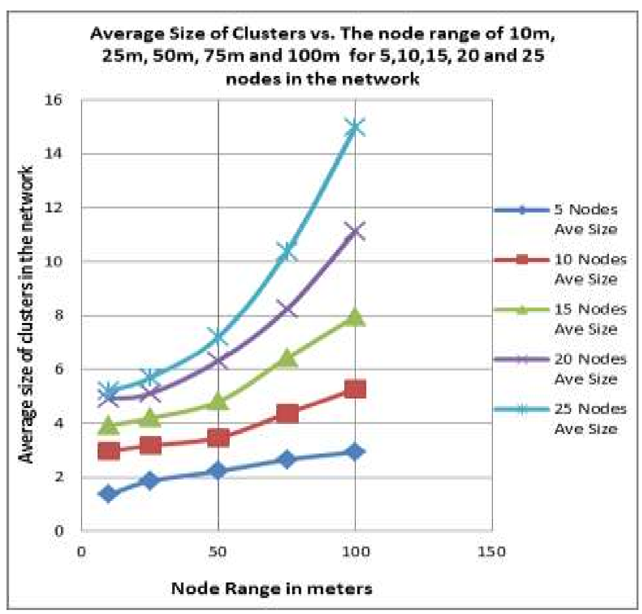

Figure 3: The average size of clusters in the network versus the different node ranges (10m, 25m, 50m, 75m and 100m) for the different number of nodes (5, 10, 15, 20 and 25 nodes)

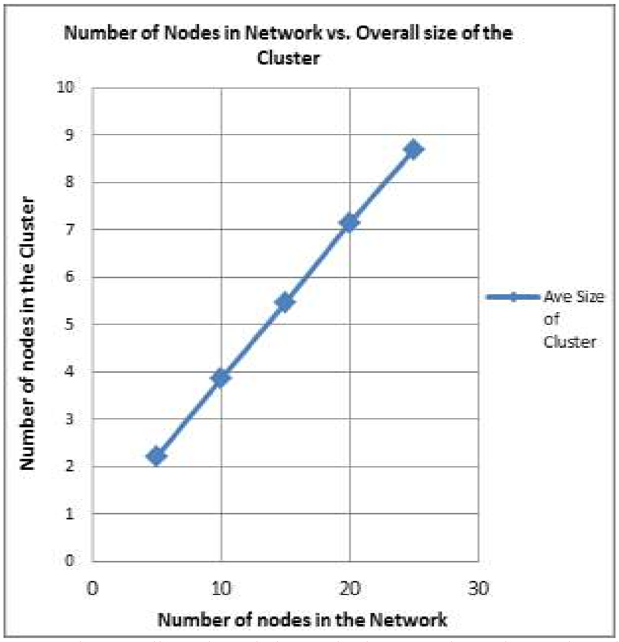

Figure 4: The overall number of clusters in the network vs. the number of nodes in the network regardless of node range.

As we can see from Figure 2 that as the node range increases the average number of nodes, begin to converge towards roughly 1.5 clusters. This means that on average as the node ranges the number of clusters formed is between 1 and 2 clusters. This is clearly supported by the increase size of clusters shown in Figure 3. As the node range increases and the number of nodes increases the average size of the cluster, measured as the number of nodes, increases.

From Figure 4 we note that there is a clear linear relationship between the number of nodes in the network and the number of nodes in the ne. This means as the number of nodes in the network increases, so does the size of the clusters. This affects the performance of clustering.

The second set of graphs (displayed on the next page) looks at the performance of the DMI algorithm dependent on different times we re-clustered the network. In this test we investigate the effect of frequent re-clustering has on the network performance.

In Figure 5 we note that how often we re-cluster is dependent of the number of nodes in the network and the node range. For instance for 10m (Wi-Fi range) and 100m range, the best re-clustering is interchangeable as the number of nodes increase.

At 10m range we note that the when we cluster every 100ms the performance overall is better. However we also note that depending on the number of nodes in the network the different clustering at times are better. For instance at 20m and 25m it is clear that re-clustering at 50ms, will yield better results whereas clustering at 20ms, does not perform better.

At 100m range we note that for 5, 10 and 20 nodes performs best when re-clustering occurs less often whereas for 15 and 25 nodes it performs as possible. We also note that at 100m node range the performance was overall better.

When we look at the case for 50m node range we notice that the when we cluster every 100ms the performance is better. We also note that cluster every 100ms, is better for all cases, with a marginal difference. This means at 50m, it is better to maintain the clusters, and do clustering less often. The best option is to cluster at every 100ms only.

From Figure 6 (on the next page), we compared the percentage of successful messages of DMI and. ORP, for the varying node ranges. For our DMI algorithm, we re-clustered every 100ms. We can see that at a 10m and 100m the clustering algorithm clearly starts to perform better than the optimal relay path algorithm as the number of nodes in the network increase. At 50m, the difference in performance is insignificant. We also note that DMI performs best in our extreme cases, namely 10m and 100m.

D. *Discussion*

We notice that as the number of nodes in the network increases so does the size of the cluster and decreases the number of clusters. From Figure 4 we notice that the average number of nodes begin to converge towards roughly 1.5 clusters. This means that on average as the node ranges the number of clusters formed is between 1 and 2 clusters at a range of 100m. This means a larger percentage of the node belong to the same cluster. For example from Figure 5 for 25 nodes, on average, the cluster size is 15nodes and this means about 60% of nodes to the same cluster. This reduces the effectiveness of having clusters and as result affect the performance of the DMI algorithm. This explains the slight decrease performance of the DMI algorithm at 100m.

We noted at a 10m node range the cluster size was restricted to a small amount of nodes in the network i.e., in Figure 4 at 25nodes on average the cluster size is about 5 nodes, and this means only 25% of the nodes belong to one cluster. DMI performed better than ORP at 10m node range.

We also noticed that frequency of re-clustering was dependent on the number of nodes in the network and the range. The results are mostly affected by the mobility traces that are collected. In Figure 5 it is clear that for 5 nodes the frequency of re-clustering is not significant and this due to the low number of nodes in the network and low frequency of the nodes meeting. However as the number of nodes increases (between 10 – 20 nodes) it begins to be significant, because there are more nodes moving around and the nodes are more likely to meet. At 25 nodes in the network we notice how often we cluster has a low significance, because it is crowded in the network, and the nodes have a high probability to meet other nodes.

This leads us to conclude limiting the cluster size in the network will lead to better performance of the DMI algorithm. However, from the preliminary results the DMI algorithm showed a small improvement to the ORP algorithm.

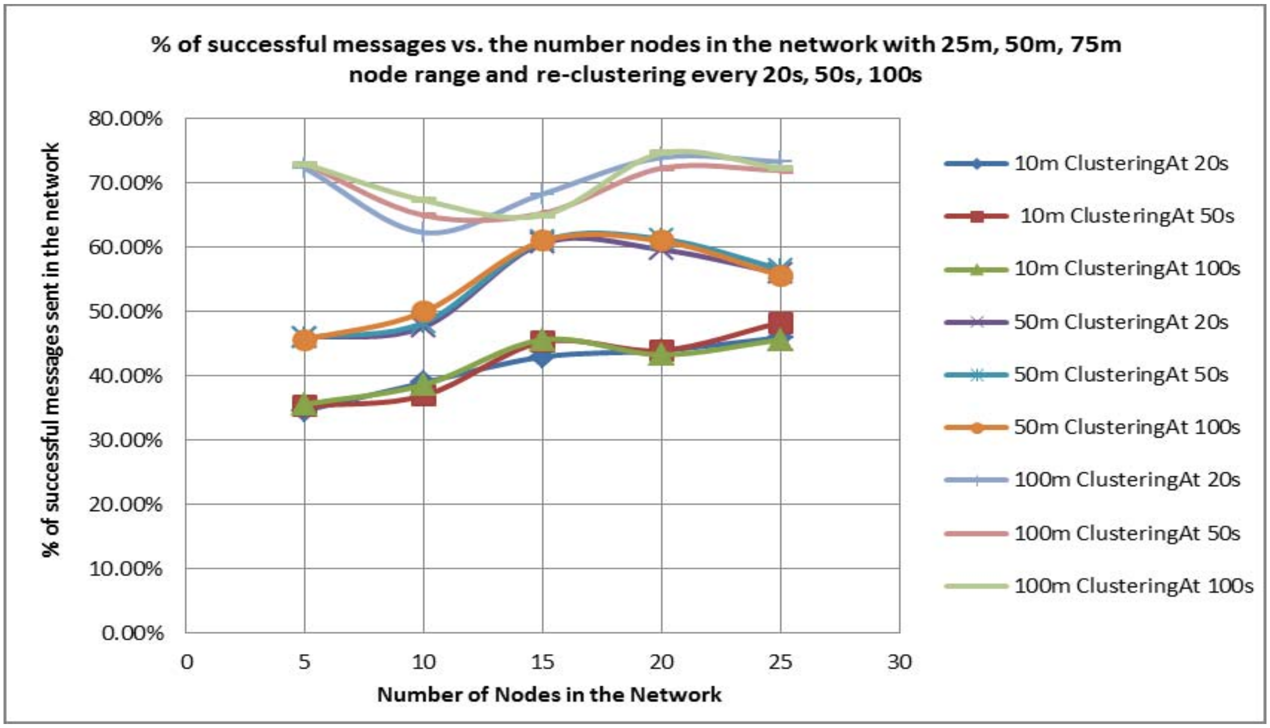

Figure 5: A comparison of % successful messages sent for different re-clustering times (20ms, 50ms, and 100ms) for the node with 10m, 50m and 100m range.

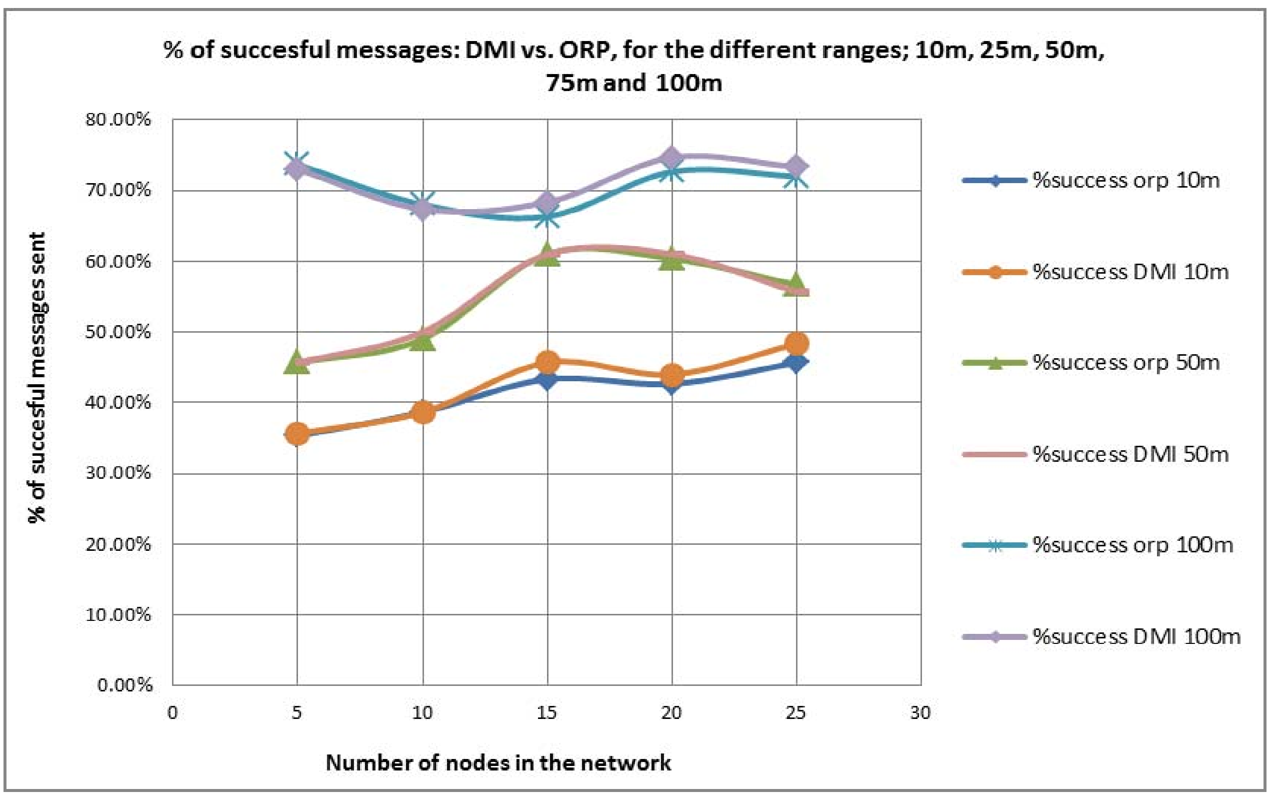

Figure 6: A graph looking at the performance of the DMI algorithm versus the ORP algorithm, over different ranges: 10m, 50m and 100m

I. Conclusion And Future Work

The data mule inter-communication (DMI) is an algorithm that handles communication between multiple mules that are clustered and makes use of the ORP algorithm for mules that are not clustered. It is aimed to improve network performance.

We showed that in extreme cases of 10m and 100m there was a small increase in DMI algorithm versus the ORP algorithm. However the DMI algorithm did not perform well when the number of nodes in the network increases. We further noted that as the number of node in the network increase and the range increased, the cluster size increased. This showed us that it was important for the cluster size should be limited. It should be limited to a small percentage of the nodes in the network.

The DMI algorithm can be applied to delay tolerated networks (DTNs) that use multiple data mules for example the DAKNet project. Some of the other areas are Wireless Sensor Networks and VANets.

In the future work we plan to investigate the best size to limit clusters too and test for how long mule should stop at a network for and finally a comparison of the time taken to send a message using DMI vs. ORP to connected isolated networks. The DMI algorithm has potential to greatly improve to also improve the latency experienced when sending data using data mules.

References

[1] R. P. Adler and M. Uppal, "m-Powering India: Mobile Communications for Inclusive Growth," The Aspen Institute and Aspen Institute India, Gurgaon, India, Third Annual Joint Roundtable on Communications Policy ISBN: 0-89843-492-0, 2008.

[2] R Agarwal and M Motwani, "Survey of clustering algorithms for MANET," International Journal on Computer Science and Engineering, vol. 1, no. 2, pp. 98-104, 2009.

[3] Y. Chen, A. Liestman, and J. Liu, "Clustering Algorithms for Ad Hoc Wireless Networks," in DIALM '00 Proceedings of the 4th international workshop on Discrete algorithms and methods for mobile computing and communications, New York, NY, USA, 2000, pp. 54-63.

[4] Eurecom, VanetMobiSim, September 2010.

[5] K Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," in ACM SIGCOMM, 2003.

[6] M. Fiore, J. Harri, F. Filali, and C. Bonnet, "Vehiclar Mobility Simulation for VANets," Simulation Symposium, 2007. ANSS '07. 40th Annual , no. 1080-241X, pp. 301-309, March 2007.

[7] N Lagraa, M Yagoubi, and S Benkouider, "Localization technique in VANets using Clustering (LVC)," International Journal of Computer Science Issues, vol. 7, no. 4, pp. 10-16, July 2010.

[8] F J Martienz, C K Toh, J Cano, C T Calafate, and P Manzoni, "A survey and comparative study of simulators for vehicular ad hoc networks (VANETs)," Wiley Journal on wireless communication and mobile computing, vol. YY, no. Z, 2009.

[9] A Pentland, R Fletcher, and A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations," Computer, vol. 37, no. 1, pp. 78-83, 2007.

[10] D Rus and Q Li, "Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks," in MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, New York, NY, USA, 2000, pp. 44-55.

[11] SearchingNetworking.com, SearchingNetworking.com, May 2011.

[12] R.C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," in Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on, May 2003, pp. 30-41.

[13] C Sommer, I Dietrich, and F Dressler, "Realistic Simulation of Network Protocols in VANET Scenarios," in 2007 Mobile Networking for Vehicular Environments, Anchorage, AK, 2007, pp. 139 - 143.

[14] University of Stuttgart, CANU project home page, September 2010.

[15] R Xu and R Wunsch, "Survey of Clustering Algorithms," IEEE TRANSACTIONS ON NEURAL NETWORKS, vol. 16, no. 3, pp. 645 - 678, Maay 2005.

[16] W Zhao, M Ammar, and E Zegura, "A Message Ferrying Approach for Data Delivery in Sparse MANets," in MobiHoc, Roppongi, Japan, 2004 May 24–26, pp. 187 - 192.