ISSN:2320-0790

An International Journal of Advanced Computer Technology

Protocol for Message Forwarding in Cellular and Mobile Opportunistic Wireless Network (Oppnets)

Lokendra Singh Songare
M.E. Scholar,
MITM, Indore,
Lokendra.songare@gmail.com

Mr.Mohsin Sheikh.
Computer Science Department,
MITM, Indore,
Er.mohsin.paper@gmail.com

Abstract: We describes hybrid approach for routing in opportunistic networks. In such networks there is no guarantee that a fully connected path between source and destination exists at any time, rendering traditional routing protocols unable to deliver messages between hosts. Thus, there is a need for a way to route through such networks. We propose hybrid approach which combines Prioritized Epidemic Routing and Probabilistic Routing. This approach prioritizes bundles based on costs to destination, source, and expiry time. Costs are derived from per-link "average availability" information that is disseminated in an epidemic manner.

IndexTerms: Opportunistic networks, Epidemic Routing, Probabilistic Routing.

I. INTRODUCTION

One area that have received much attention recently and that remedies many of the situations where no infrastructure is available is that of ad hoc networking [5]. In an ad hoc network, all nodes participate in the routing and forwarding of packets, so if two nodes can not communicate directly, intermediate nodes aid in forwarding the packet between them. One of the most basic requirements for "traditional" networking, which also holds for ad hoc networking, is that there must exist a fully connected path between communication endpoints for communication to be possible. There are however a number of scenarios where this is not the case (thus rendering the use of ad hoc networking protocols impossible), but where it still is desirable to allow communication between nodes.

Recent times have seen the emergence of a new kind of mobile multi hop wireless network known as Disruption Tolerant Networks (DTN), or Intermittently Connected Networks (ICN), or opportunistic networks. The key distinguishing feature of a DTN1 from a Mobile Ad Hoc Network (MANET) is that there may never be a contemporaneous end-to-end path, but the union of network snapshots over time may present an end-to-end path. Conventional MANET routing protocols typically drop packets in such situations and therefore are insufficient. Applications of DTNs include military communications [6], inter-planetary networks [7] and networks in under-developed areas [8].

We present a novel hybrid approach for routing in opportunistic network. We propose the use of probabilistic routing [9], using an assumption of non-random mobility of nodes to improve the delivery rate of messages while keeping buffer usage and communication overhead at a low level and Prioritized Epidemic Routing [1] where we impose a partial ordering on the messages called bundles for transmission and deletion. The priority function, which is slightly different for transmission and deletion, is based upon four inputs - the current cost to destination, current cost from source, expiry time and generation time. Inter-node costs are computed using a novel metric called average availability. Each link's average availability is epidemically disseminated to all nodes. As a result of this priority scheme, hybrid approach maintains a gradient of replication density that roughly decreases with increasing distance from the destination. Epidemic routing is unbeatable from the point of view of successful delivery as long as the load does not stress the resources (bandwidth, storage). Furthermore, unlike most existing works in the literature, Epidemic does not rely on extrapolating previous contact information. This approach uses the simplicity and power of Epidemic while fixing it in the one place that it is weak – high loads – producing a simple, yet robust and efficient it uses the probabilistic routing.

II. RELATED WORK

Vahdat and Becker present a routing protocol for intermittently connected networks called Epidemic Routing [10]. This protocol relies on the theory of epidemic algorithms [11] by doing pair-wise information of messages between nodes as they get contact with each other to eventually deliver messages to their destination. Hosts buffer messages even if it there is currently no path to the destination available. An index of these messages called a summary vector is kept by the nodes, and when two nodes meet they exchange summary vectors. Afterthis exchange, each node can determine if the other node has some message that was previously unseen to this node. In that case, the node requests the messages from the other node. This means that as long as buffer space is available, messages will spread like an epidemic of some disease through the network as nodes meet and "infect" each other.

Each message must contain a globally unique message ID to determine if it has been previously seen. Besides the obvious fields of source and destination addresses, messages also contain a hop count field. This field is similar to the TTL field in IP packets and determines the maximum number of hops a message can be sent, and can be used to limit the resource utilization of the protocol. Messages with a hop count of one will only be delivered to their final destination.

The resource usage of this scheme is regulated by the hop count set in the messages, and the available buffer space at the nodes. If these are sufficiently large, the message will eventually propagate throughout the entire network if the possibility exists. Vahdat and Becker do however show that by choosing an appropriate maximum hop count, delivery rates can still be kept high while the resource utilization is lower in the scenarios used in their evaluation [10].

A communication model that is similar to Epidemic Routing is presented by Beaufour et al. [12], focusing on data dissemination in sensor networks. The Pollen network proposed by Glance et al. [13] is also similar to Epidemic Routing.

Chen and Murphy propose a protocol called Disconnected Transitive Communication (DTC) [14]. It utilizes an application-tunable utility function to locate the node in the cluster of currently connected nodes that it is best to forward the message to based on the needs of the application. In every step, a node searches the cluster of currently connected nodes for a node that is "closer" to the destination, where the closeness is given by a utility function that can be tuned by the application to give appropriate results.

Shen et al. propose Interrogation-Based Relay Routing, a routing protocol for routing in ad hoc space networks with Scientific Earth Observing (SEO) satellites [15], characterized by frequently changing topologies, and sparse and intermittent connectivity. The satellites interrogate each other to learn more about network topology and nodal capacity to make intelligent routing decisions.

Work by Li and Rus [16] deal with a similar problem of communication in disconnected networks. They propose a solution where nodes actively change their trajectories to create connected paths to accommodate the data transmission. While this might work in military applications and in some robotic sensor networks, in most scenarios it is not likely that nodes will move just to accommodate communication of other nodes (if it is even possible to communicate the need for it).

Grossglauser and Tse looks at the utility of using the mobility of nodes to deliver messages to their destination from a slightly different point of view. One major problem with ad hoc networks is that due to interference of concurrent transmissions between nodes they scale badly. Grossglauser and Tse show the by only doing local communications between neighbors and instead relying on the movement of nodes to bring a message to its destination, this problem can be mitigated [17].

III. PROPOSED SYSTEM

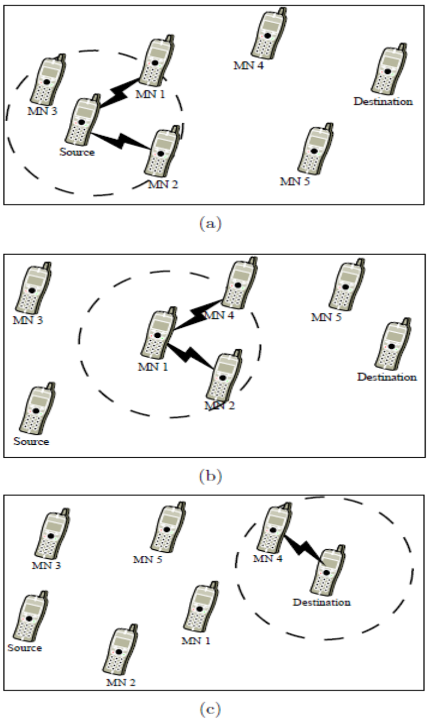
The proposed routing protocol is a novel approach for routing in opportunistic network. This protocol works in two phases, in first phase the selection of neighboring node among the all nodes in range, which promises the delivery of message to destination is selected; here we select two such nodes for more chances of forwarding the messages in right direction. In second phase the exchange of messages takes place in such a way that the node only exchanges those messages that it wouldn't have, so unnecessary message exchange is avoided. For first phase we use Probabilistic routing technique for selection of nodes and for message exchange we are using Epidemic routing. In this approach we combine both the Probabilistic Routing and Epidemic Routing. Our approach works in two steps; in the first step we use Probabilistic Routing for selecting the only two neighbors with highest probability value among all. Second step will follow Epidemic Routing to determine which messages stored remotely have not been seen by the local host. In turn, each host then requests copies of messages that it has not yet seen. ProEp protocol uses advantages from Probabilistic Routing and Epidemic Routing, so it selects only two nodes which have greater probabilistic value to avoid the unnecessary broadcasting. This ultimately saves the resource utilization and cost.

In this approach we combine both the Epidemic Routing and Probabilistic Routing. A node forwards the message to the two neighbors which are having maximum delivery predictability. Delivery predictability, $P_{(a,\,b)}$ ε[0, 1], at every node 'a' for each known destination 'b' is ability of 'a' to deliver message to destination 'b'.

When two nodes meet, they exchange summary vectors which in this case also contain the delivery predictability information stored at the nodes. This information is used to update the internal delivery predictability vector, and then the information in the summary vector is used to decide which messages to request from the other node as described below.

Each host maintains a buffer consisting of messages that it has originated as well as messages that it is buffering on behalf of other hosts. A hash table indexes this list of messages, keyed by a unique identifier associated with each message. Each host stores a bit vector called the summary vector that indicates which entries in their local hash tables are set. To avoid redundant connections, each host maintains a cache of previously communicated hosts. When two hosts come into communication range of one another, they exchange their summary vectors to determine which messages stored remotely have not been seen by the local host. In turn, each host then requests copies of messages that it has not yet seen.

For example, while sending the message the source node searches the nodes in his range, then by exchanging delivery predictability information he finds MN1 and MN2 have higher delivery predictability than other nodes therefore source node forwards message to nodes MN1 and MN2 as shown in Fig. 1.a. The nodes who receive the message from source node they again follow the same procedure as source node but as shown in Fig.1.b MN2 is receiver of source as well as node MN1. MN1 and MN2 only exchange its summary vector. And by exchange they know that they don't have new messages to exchange so they stop communication. In Fig. 1.c the node MN4 follows same procedure and message reaches to the destination.



(a)

(b)

(c)

Figure 1: Hybrid Routing

Mathematical Modeling

Let $P_{(a,\ b)}$ ε[0, 1] represents delivery predictability value at every node a for each known destination b. This indicates how likely it is that this node will be able to deliver a message to that destination.

Delivery predictability calculation

Delivery predictability calculated in three parts:

- To update the metric whenever a node is encountered

$$P_{(a,\ b)} = P_{(a,\ b)old} + (1 - P_{(a,\ b)old}) \times P_{init} \quad (1)$$

Where,

– P_{init} ε [0, 1] = an initialization constant (0.75),
– $P_{(a,\ b)old}$ = Previous probability of node,
– $P_{(a,\ b)}$ = New probability of node.

- Ageing of delivery predictability values:

$$P_{(a,\ b)} = P_{(a,\ b)\ old} \times \gamma^{k} \quad (2)$$

Where, γ ε [0, 1] = the ageing constant,
k=Number of time units that have elapsed since the last time the metric was aged.

- Transitive probability calculation:

$$P_{(a,\ c)} = P_{(a,\ c)old} + (1 - P_{(a,\ c)old}) \times P_{(a,\ b)} \times P_{(b,\ c)} \times \beta \quad (3)$$

Where,
– $P_{(a,\ c)old}$ = Previous probability,
– $P_{(a,\ b)}$= Probability of node A to node B.
– $P_{(a,\ c)}$ = Probability of node A to node C.
– β ε [0, 1] = A scaling constant that decides how large impact the transitivity should have on the delivery predictability

By using these three formulas we get delivery predictability value of each node.

Index representation of buffered messages:

Summary vector= Host ID + locally generated message ID.
SVA = Summary vector of node A.
SVB = Summary vector of node B.
Calculation of missing messages:
Missing messages= SV ASV B
And In last step, A transmits the requested messages to B

IV. EXPRIMENTS AND RESULTS

Simulation Setup

To study and evaluate the performance of the proposed protocol, we have developed the wireless network simulator framework. The simulator contains a model of the wireless nodes. Furthermore, the simulator has the limited no of nodes. Nodes are moving within the bounded area randomly with a constant speed. To aid in the evaluation of the protocol, we have develop a simple simulator. The simulator focuses on the operation of the routing protocols, and does not simulate the details of the underlying layers. When doing an evaluation of a protocol or system, it is very important that the models used in the evaluation are realistic. Since we base our protocol on making predictions depending on the movements of nodes, it is vital that the mobility models we use are realistic. One mobility model that has been commonly used in evaluations of ad hoc routing protocols is the random way-point mobility model. In this model, nodes randomly choose a destination and a speed and move there. Upon arrival at the destination, the nodes pause for a while and then choose a new destination.

In this evolution of the given protocol, we have focused on comparing the performance with regard to the following metrics. First of all, we are interested in the *message delivery delay*, i.e. to find out how long time it takes a message to be delivered. Even though applications using this kind of communication should be relatively delay-tolerant, it is still of interest to consider the change the *hope count* and *queue size* values.This indicates how the system resource utilization is affected by the different settings, which is crucial so that valuable resources such as bandwidth and energy are not wasted.

We ran simulations for each scenario several times, varying the queue size at the nodes (the number of messages can buffer and the hop count value set in the messages. Following values for parameters are kept fixed in our simulation.

Parameter	Values
P_{init}	0.75
γ	0.25
B	0.98

Figure 2: Parameter Setting

The setup of experiment includes 24 nodes on approximately 100m X 100m. I am taking nodes with varying hop count & queue size. Nodes ranges from 24, 22,20,18,16 with the hop count value 3 & 5 and queue size with 5 and 10 number of message storing capacities.

Results

The performance could be measured using the following parameters:

- Number of hop count given for the message.
- Queue or buffer size of the node.
- Travelling time (delay) of the message from source to the destination.
- Number of nodes available on the field.
- Speed of node

Result analysis using Travel time (delays)

Initially we are taking different nodes separately and observe the effect with change in hop count and queue size value with all four combinations. In each case we plotted a graph with reference to delays (average travel time), as seen in fig 3 and 4.

As seen in Fig. 3 if we decrease the number of nodes the direct effect on the delays. When hop count is 3, the average travel time is lesser than hop count value 5. It is because if we decrease the hop count value, there will be less number of intermittent nodes. If message will reach to its hop count value message would be dropped. So when hop count value was less and we are tried to send message with such minimum value and if destination was not found within that hop count value, ultimately message was dropped. So it is better to have minimum value for hop count which ultimately goes through lesser number of intermittent nodes and requires less time to travel. But if we choose low value for hop count this will leads to message drops when hop count value will reach. And if we increase the value for hop count affects greater delays.

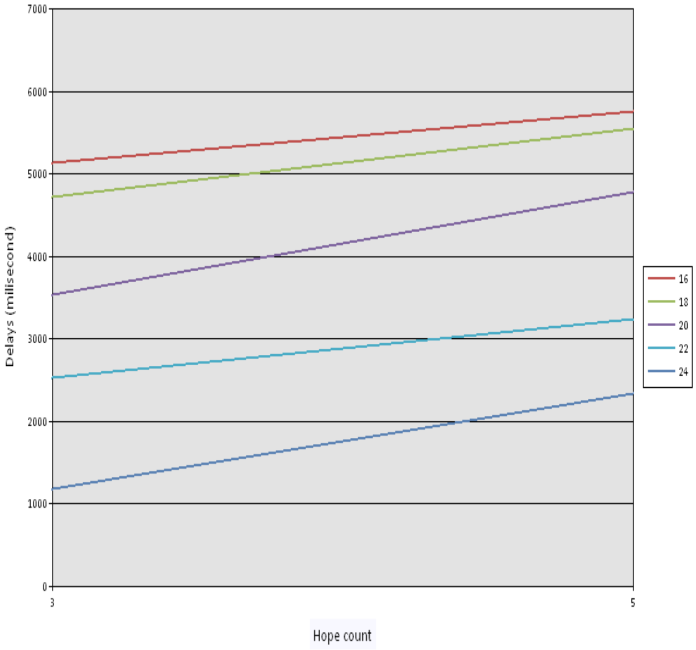


Figure 3: Hope count vs. delay with different nodes

After that we are changing the values of queue size and observe the changes which shown in Fig. 4. Now in this case if we increase the size of queue, the delays would be increases. As we know, queue means the buffer which holds the message generated by self and received while moving around the network for routing purpose. So when we increase the buffer capacity so less messages would drop. But this will affect the message exchange capabilities, when more messages are in queue we can't drop more messages for new ones.

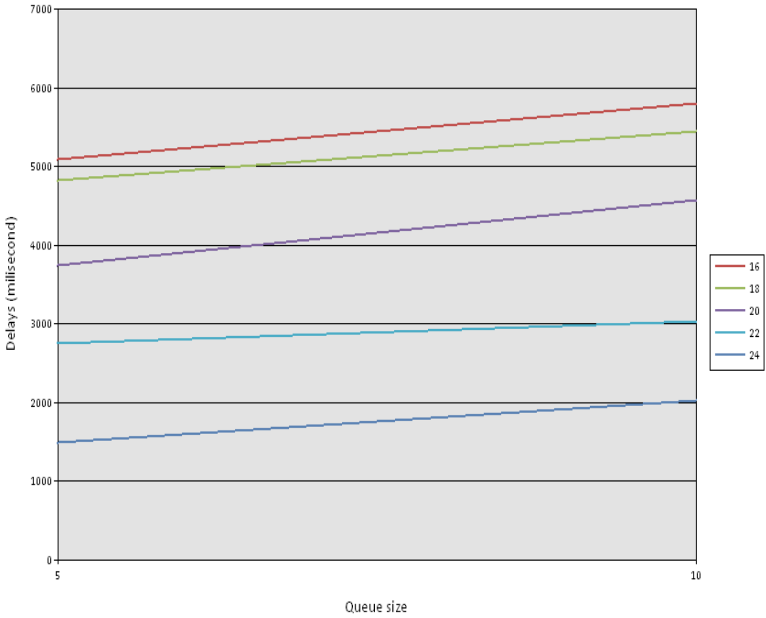


Figure 4: Queue size vs. delay with different nodes

Fig. 3 and 5 have only one difference that, in Fig. 3we are taking nodes separately but in Fig. 5we consider average values for all nodes in observation. Fig. 5 clearly shows that increase in the value of hop count would results in higher delays. It is because as the intermittent nodes increases off course delays should increasers parallel.

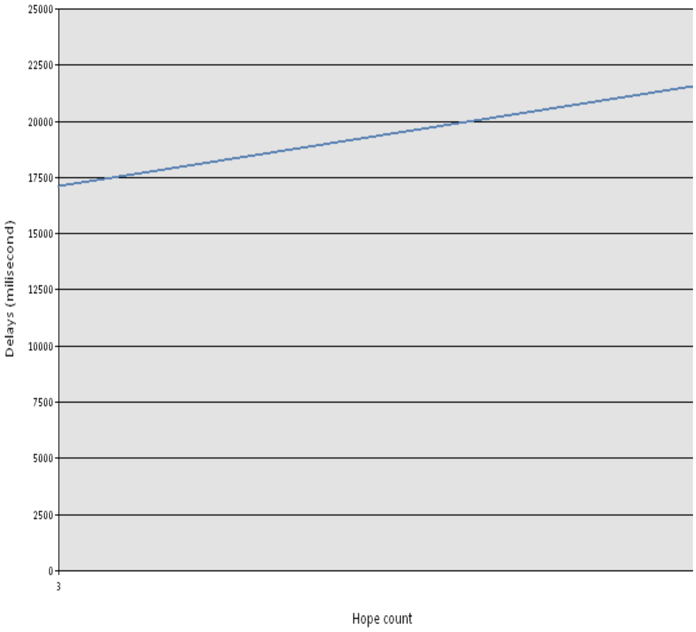


Figure 4.3: Hope count vs. delay

Here also Fig. 4 and 6 have only one difference that, in Fig. 4 we are taking nodes separately but in Fig. 6we considering average values. Fig. 6 clearly shows that increase in the value of queue size would results in fewer delays.

Effect of decreasing the number of nodes would result in similar results as hop count results. Here also decrease in nodes will have increases the delays because of less intermittent nodes for routing the messages.

Looking at the delivery delay graphs Fig. 4.3, it seems like increasing the queue size, also increases the delay for messages. However, the phenomenon seen is probably not mainly that the delay increases for messages that would be delivered even at a smaller queue size (even though large buffers might lead to problems in being able to exchange all messages between two nodes, leading to a higher delay), but the main reason the average delay is higher is coupled to the fact that more messages are delivered. These extra delivered messages are messages that were dropped at smaller queue sizes, but now are able to reside in the queues long enough to be delivered to their destinations. This incurs a longer delay for these messages, increasing the average delay.

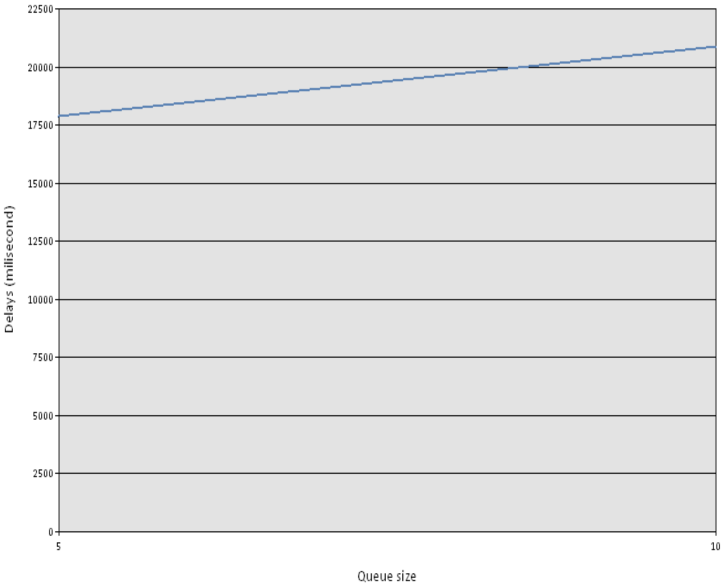


Figure 6: Queue size vs. delay

Result analysis using Speed of node

Considering the speed of nodes is important aspect through that many things are affected. As we change the speed of nodes the hop count value and queue size affected. I assign some random values for speed of the nodes. Finally I calculate the average speed of nodes for the instance in which simulation is running.

Fig 7 and 8 shows the affect of the speed of node during the travelling over the network. At the time only a node pair will transmit the messages, so increase in speed will also increases the queue size and hope count. Increase in speed will effect in more contact of nodes so transferring the messages themselves depending on probabilities will carry more message in message queue. Similarly hope count will also increases as increase in speed of node.

As shown in Fig. 7, it is clear that as I increase the speed of nodes the average hop count value also increases. If node speed increases then the node interaction also increases parallel. As frequency of node interaction increases they would also exchanges messages so frequently. This results increase in hop count value.

As shown in Fig. 8, increase in node speed would give increase average queue size. When node speed increases then the node interaction also increases parallel. Nodes frequently come in range and exchanges messages so frequently. When more message exchanges occur then node also require larger queue size to store those messages.

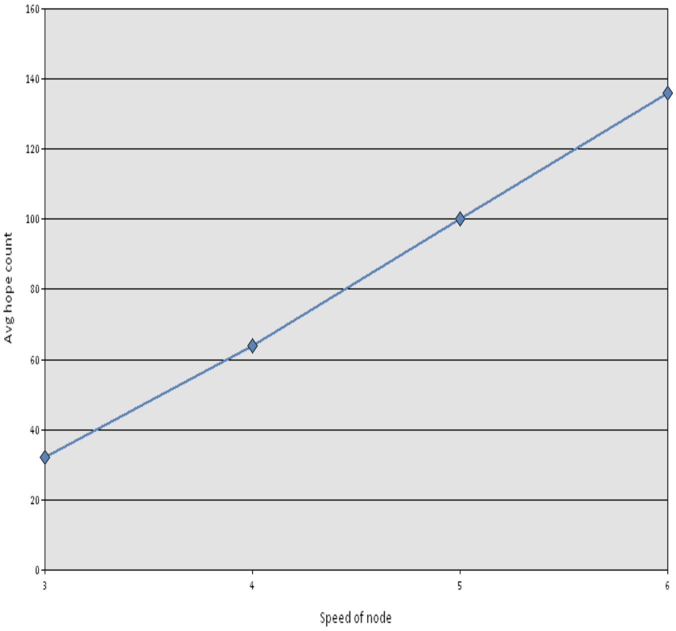


Figure 7: Graph of Speed of node vs. Average hope count

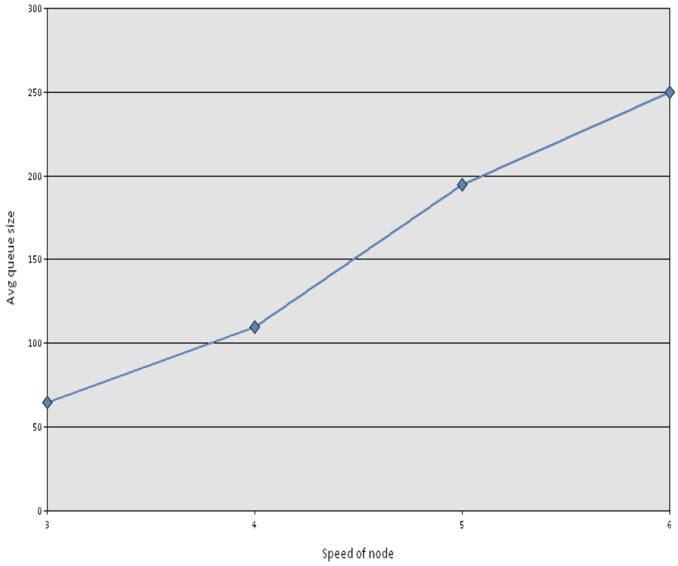


Figure 8: Graph of Speed of node vs. Average queue size

V. REFERENCES

[1] Ram Ramanathan, Richard Hansen, PrithwishBasu, “Prioritized Epidemic Routing for Opportunistic Networks”,MobiOpp’07, June 11, 2007, San Juan, Puerto Rico, USA. Copyright 2007 ACM 978-1-59593-688-2/07/0006.

[2] Anders Lindgren,AvriDoria, OlovSchel´en,” Probabilistic Routing in Intermittently Connected Networks ”.

[3] Chung-Ming Huang, Kun-chanLan and Chang-Zhou Tsai,” A Survey of Opportunistic Networks”, 22nd International Conference on Advanced Information Networking and Applications – Workshops, 978-0-7695-3096-3/08 IEEE DOI 10.1109/WAINA.2008.292.

[4] LeszekLilien, ZilleHuma Kamal, Vijay Bhuse, and Ajay Gupta, “Opportunistic Networks: The Concept And Research Challenges In Privacy And Security”.

[5] Elizabeth M. Royer and Chai-KeongToh, “A review of current routing protocols for ad hoc mobile wireless networks,” IEEE Personal Communications Magazine, vol. 6, no. 2, pp. 46–55, Apr. 1999.

[6] P. Marshall. The disruption tolerant networking program, 2005. http://www.darpa.mil /sto/ solicitations/DTN/briefs.htm.

[7] S. Burleigh et al. Delay-tolerant networking: An approach to interplanetary internet. IEEECommunications Magazine, June, 2003.

[8] A. Pentland, R. Fletcher, and A. Hasson. Daknet: Rethinking connectivity in developing nations. IEEE Computer 37(1), 78-83, Jan 2004.

[9] Anders Lindgren, AvriDoria, and OlovSchel´en, “Poster: Probabilistic routing in intermittently connected networks,” in Proceedings of The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), June 2003.

[10] Amin Vahdat and David Becker, “Epidemic routing for partially connected ad hoc networks,” Technical Report CS-200006, Duke University, April 2000.

[11] Werner Vogels, Robbert van Renesse, and Ken Birman, “The power of epidemics: Robust communication for large-scale distributed systems,” in Proceedings of First Workshop on Hot Topics in Networks (HotNets-I),28-29 October 2002, Princeton, New Jersey, USA, Oct 2002.

[12] Allan Beaufour, Martin Leopold, and Philippe Bonnet, “Smart-tag based data dissemination,” in First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA02), June 2002.

[13] Natalie Glance, Dave Snowdon, and Jean-Luc Meunier, “Pollen: using people as a communication medium,” Computer Networks, vol. 35, no. 4,pp. 429–442, March 2001.

[14] Xiangchuan Chen and Amy L. Murphy, “Enabling disconnected transitive communication in mobile ad hoc networks,” in Proc. of Workshop on Principles of Mobile Computing, colocated with PODC’01, Newport, RI (USA), aug 2001, pp. 21–27.

[15] Chien-Chung Shen, GirishBorkar, SundaramRajagopalan, and ChaipornJaikaeo, “Interrogation-based relay routing for ad hoc satellite networks,” in Proceedings of IEEE Globecom 2002, Taipei, Taiwan, November 17-21,2002.

[16] Qun Li and Daniela Rus, “Communication in disconnected ad-hoc networks using message relay,” Journal of Parallel and Distributed Computing, 2003.

[17] Matthias Grossglauser and David Tse, “Mobility increases the capacity of ad-hoc wireless networks,” IEEE/ACM Transactions on Networking, vol. 10, no. 4, Aug 2002.

[18] P. Jacquet T. Clausen. Optimized link state routing protocol (olsr), October 2003. Internet RFC 3626.

[19] E.W. Dijkstra. A note on two problems in connation with graphs. Numer. Math. 1:269, 271, 1959.