

Stability and Hop-Count based Approach for Route Computation in MANET

Sridhar K N and Mun Choon Chan

School of Computing, National University of Singapore, Singapore 117543

Email: {sridhark,chanmc}@comp.nus.edu.sg

Abstract—In this paper, we present an approach for MANET routing based on stability and hop-count, where the stability metric considered is the residual lifetime of a link. We view stability based routing not as a separate routing protocol but as an enhancement to a hop-count based routing protocol (e.g. DSR or AODV), so that the expected residual lifetime as well as hop count of a route are taken into account.

First, we investigate how residual link lifetime is affected by parameters such as speed and mobility pattern using simulation. The result shows that residual link lifetime is a function of current link age, mobility speed and mobility pattern and does not vary monotonically with age. Therefore, intuitive idea such as *older links are more stable*, which is used in existing stability-based routing algorithms like Associativity Based Routing (ABR), does not hold across a large spectrum of mobility speeds and models. Instead, the reverse can be true.

We propose a stability and hop-count based routing algorithm (SHARC) using DSR as the basic routing protocol. The stability of a path is calculated using a simple histogram based estimator. Simulation results show that SHARC performs better than a hop-count only algorithm (DSR) and a stability only algorithm for both throughout of long-lived TCP and response time of short data transfer. SHARC also performs close to an algorithm with perfect knowledge of link residual lifetime in many cases.

I. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a wireless network consisting of mobile nodes, which can communicate with each other without any infrastructure support. In these networks, nodes typically cooperate with each other, by forwarding packets for nodes which are not in the communication range of the source node.

Typically, routing protocols are classified according to the route discovery philosophy, into either reactive or proactive. Reactive protocols are on-demand. Route-discovery mechanisms are initiated only when a packet is available for transmission, and no route is available. On the other hand, proactive protocols are table-driven. Routes are pre-computed and stored in a table, so that route will be available whenever a packet is available for transmission.

In our work, we see the classification from a different perspective. We classify routing protocols on the basis of metrics considered by them. We broadly classify routing protocols into hop-count based, QoS (e.g. bandwidth) based, and stability (or availability) based. Each of these classes of routing protocols can be either proactive or reactive.

Majority of the routing protocols proposed till date are based on the hop-count metric. Hop-count based algorithms typically try to optimize the length of the route. The second category of our classification is routing protocol based on QoS metric such as bandwidth. There have been many proposed solutions for QoS routing in MANETs.

The last category based on link stability is unique to wireless network. Link stability refers to the ability of a link to survive for a certain duration. The higher the link stability, the longer is the link duration. The stability of a link depends on how long two nodes, which form that link, remain as neighbors. Two nodes are neighbors when they remain within each other's communication range, or the signal strength is above certain threshold. Mobility causes link breakage and leads to route recovery. Transport layer performance degrades as a result of packet loss and trigger congestion control mechanism. A more *stable* link should therefore be preferred. However, routing algorithms that are based only on link stability have either been shown to exhibit little improvement over hop-count based algorithm [3] or the improvement comes when link lifetime can be accurately predicted [7]. A crucial issue with stability based routing algorithm is that much longer routes can be obtained compare to hop-count based routing.

In this paper, we make two contributions. First, we investigate the impact of mobility on link stability over a wide spectrum of mobility speeds and mobility patterns. We show from extensive simulation results that link lifetime is a function of current link age, mobility speed and mobility pattern and does not vary monotonically with age. Therefore, the intuitive idea that *older links are more stable* [2] does not work well across a large spectrum of mobility speeds and models. Second, we propose an algorithm called Stability and Hop-count based Algorithm for Route Computing *SHARC* that considers both the hop-count and stability metrics. SHARC uses DSR (which is hop-count based) as the basic routing protocol and uses a simple estimator for computing residual link lifetime. While simple, the estimator provides lifetime estimates that is *sufficient for route selection purpose*. We are not aware of any other ad hoc routing algorithm that combines hop-count and stability metrics. Simulation results show that SHARC performs better than both DSR and a stability based algorithm. SHARC also performs close to an algorithm that has perfect knowledge of link residual lifetime.

This paper is organized as follows. In Section II, we present related work on stability based routing. The simulation results on residue link lifetime is presented in Section III. In Section IV the descriptions of SHARC and its evaluation are presented. Finally, we conclude in Section V.

II. RELATED WORK

As many popular MANET routing algorithms are hop-count based, we will present related work on stability based routing protocols in this section.

Path stability depends on the availability of all the links constituting the path. A link is available when the radio quality of the link satisfies the minimal requirement for a successful transmission. Stability based protocols use stability or variations of stability as the routing metric. The implicit goal of most stability based routing protocols is to find and select the long(est)-lived routes. The difference lies in how the stability of a link is estimated and how these link estimates can be combined to form an end-to-end estimates.

Associativity Based Routing (ABR) [2] is probably the first protocol in the class of stability based protocols for MANETs. In ABR, a new metric called *associativity* is defined to determine link stability. In simple terms, ABR is based on the idea that nodes which are neighbors for a threshold period are more likely to remain as neighbors for longer time, or less likely to move away. ABR assumes that after the threshold period, nodes move with similar speeds and directions and tend to stay together. One of the problem with ABR is the choice of the threshold value. This value may vary depending on the mobility patterns.

Signal Stability based Adaptive (SSA) [3] is a routing protocol, which finds route based on signal strength, and location stability. In SSA, a mobile node measures the signal strength received from other nodes, and this information is used to estimate the link stability between them. The location stability mechanism is considered only as a supplement to signal-strength measurements. Simulation results in [3] shows that the performance of SSA with location stability mechanism is not much better than a simple shortest path algorithm.

The protocol RBAR[5] is an extension to SSA which assigns a threshold to the level of signal-strength and based on this threshold choose the routes. This protocol suffers from the disadvantage of having to choose the optimal threshold values.

The work by Gerharz et. al. [4] studies the issue of link duration in mobile wireless ad hoc network and is closest to our work. They found that link durations vary with age and proposed techniques to measure residual lifetime online. They then proceed to propose two metrics for selecting a stable link: highest average residual lifetime and highest 75% quantile. From their analysis they found that initially a link's average residual lifetime decreases with increasing current age, and after a threshold, where threshold corresponds to the modal value of link duration distribution, the residual lifetime increases with current age. However, in our simulations we found that this may not be true in some cases.

Cheng et.al., [7] also study the distribution of link lifetimes in ad hoc network. However, they focus mainly on the factors which influence the link lifetime. They consider the number of mobile nodes, node minimum speed and moving probability as dominating factors that influence link lifetime. From our experiments, we observe that average node density does not differ much across different mobility models. In [7], the authors also mention the possibility of considering route length along with route lifetimes though no algorithm was proposed.

In the PATHS analysis [6], the authors study the link durations and path durations. Detailed analysis was carried out by considering the effects of number of hops, maximum

velocity and transmission range. In this study, the authors found that maximum velocity and number of hops have a inverse relation with path duration, whereas transmission range has a direct relation with path duration. The authors also mentioned that for higher mobile speeds, path durations can be approximated with exponential distributions. The simulation results on link and path distributions are used to develop a analytical model for path duration.

Yih-Chun Hu et. al., [8] explore the cache strategies in DSR and propose some mobility metrics. They found that link-cache strategies are better than path-cache strategies. As one of the link-cache strategy, they propose technique to combine stability value of a link, which is dependent on the usage of the link, and hop-count. They found that this technique though perform better, but not better than a static scheme of 5 seconds expiration. We will show that our scheme performed better than the best link-cache schemes.

Our work is unique in the following way. While related work study the impact of mobility on link duration, they do not extensively evaluate the relationship over a wide spectrum of mobility speed and patterns. In addition, we proposed a routing algorithm that combines element of hop-count and stability while related work does not.

III. MEASURING LINK DURATION AND RESIDUAL LIFETIME

In this section, we describe our study of link duration and residual lifetime.

A. Mobility Patterns

Node mobility is one of the most important characteristics of MANET. There have been various mobility models or patterns proposed for MANETs. These patterns try to capture most of the common mobility patterns, but few patterns capture realistic movements of nodes in MANETs. There have also been work that study various mobility models, and the performance of routing protocols across different mobility models. In our work, we use the following mobility patterns: random waypoint, Reference Point Group Mobility (RPGM), freeway mobility and Manhattan mobility models[1]. These 4 models capture a broad spectrum of behaviors that help to illustrate the impact of mobility pattern on routing.

B. Simulation Environment

We use different mobility pattern generators for different mobility patterns. We use the "setdest" tool, which comes along with the distribution of ns-2 [14] to generate random waypoint mobility pattern. Mobility generator obtained from the Toilers group [11] is used to generate the group-mobility (RPGM) mobility patterns. For freeway and Manhattan mobility patterns, the generators from the Nile group at USC [12] and the bonnmotion [13], respectively, are used. For the group mobility models, we consider three cases. In the first case, termed as RPGM1, we have a single group with 50 nodes. In the second case, termed as RPGM2, we consider 5 groups with 10 nodes each. Finally, in RPGM3, there are 10 groups with 5 nodes each.

In all the simulations, 50 nodes are simulated in an area of 1Km by 1Km. The transmission range of the node is set to 250m. Maximum speed of the node is allowed to be varied

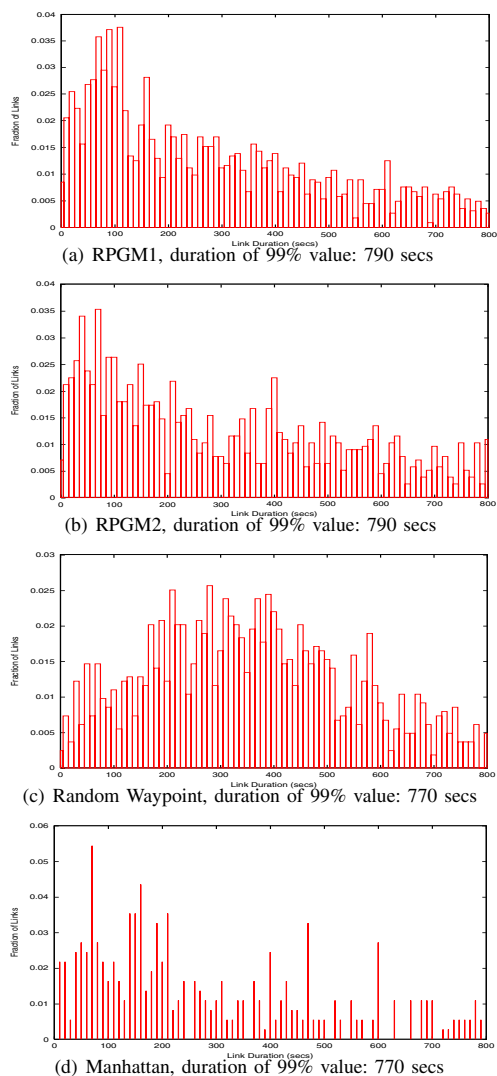


Fig. 1. PDF of Link Duration, 1 m/sec

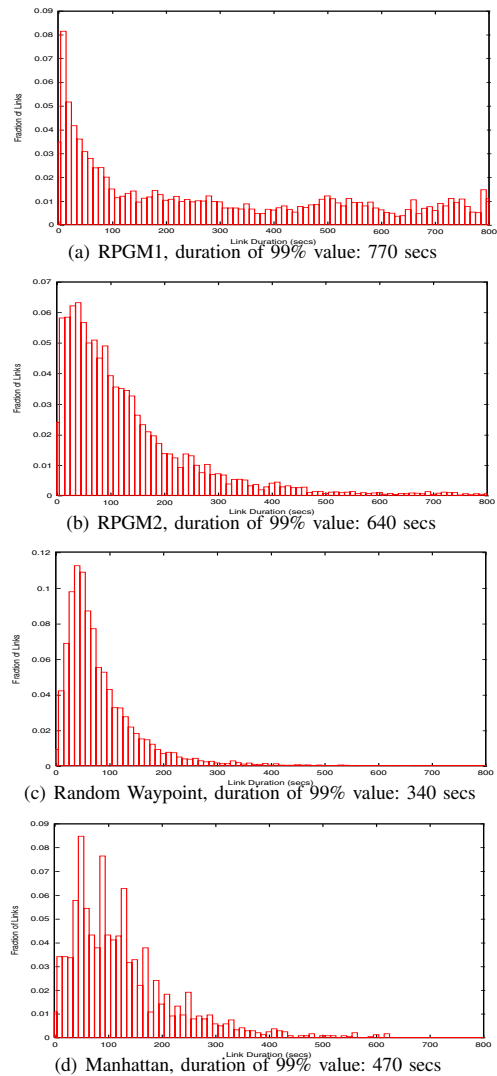


Fig. 2. PDF of Link Duration, 10 m/sec

from 1 m/sec to 50 m/sec. For the study of link durations and residual lifetimes, only the speeds of 1 m/sec, 10 m/sec, and 30 m/sec are considered.

This simulation environment is used in all simulations in subsequent sections. All simulations are carried out for 900 secs. and we take the average of 5 to 7 runs unless stated otherwise.

C. Link duration

Link duration is calculated as the duration of continuous connection time between a node and its neighbor. In order to remove any edge effect, a link duration is considered only when the link is broken before the end of the simulation. We look at the probability density function (PDF) of these durations using bin size of 10 seconds. For maximum speed, we consider the speeds 1, 10 and 30 m/sec, although only 1 and 10 m/sec plots are shown due to space limitations.

Figures 1 and 2 show the link duration plots for speeds 1 m/sec and 10 m/sec respectively. The 99 percentile values are specified under each plot. These values are useful because

they help us to focus on the relevant portion of the plots. Due to space limitation, only plots for RPGM1, RPGM2, random waypoint and Manhattan will be shown.

We can see from Figures 1 and 2 that in almost all the cases, group mobility patterns have longer tails than other mobility patterns. Therefore, there are higher fraction of links with longer durations, as compared to other patterns. This behavior is quite natural considering the properties of RPGM mobility pattern.

As the speed increases, the duration at which the 99 percentile value occurs decreases. At a speed of 10 m/sec, link durations above 500 secs are rare for random waypoint, freeway and Manhattan models. Whereas for group models, even at the speed of 30 m/sec, the 99 percentile values are above 500 secs. So at higher speeds, durations above 500 secs is not important except for the group mobility patterns.

The modal values of link durations is an interesting aspect in these plots. The modal values tend to decrease with increase in speeds. Gerharz et. al. [4] show in their link duration study that the histogram's peak (modal value)

occurs roughly at the transit time of two mobile nodes crossing each other's transmission range. However, from our results we did not find this pattern in all the cases. From RPGM1 at low speed (1m/sec) the peak occurs at 100secs, whereas for RPGM2 the modal value occurs at 70secs. According to [4], it should have occurred at roughly 250secs.

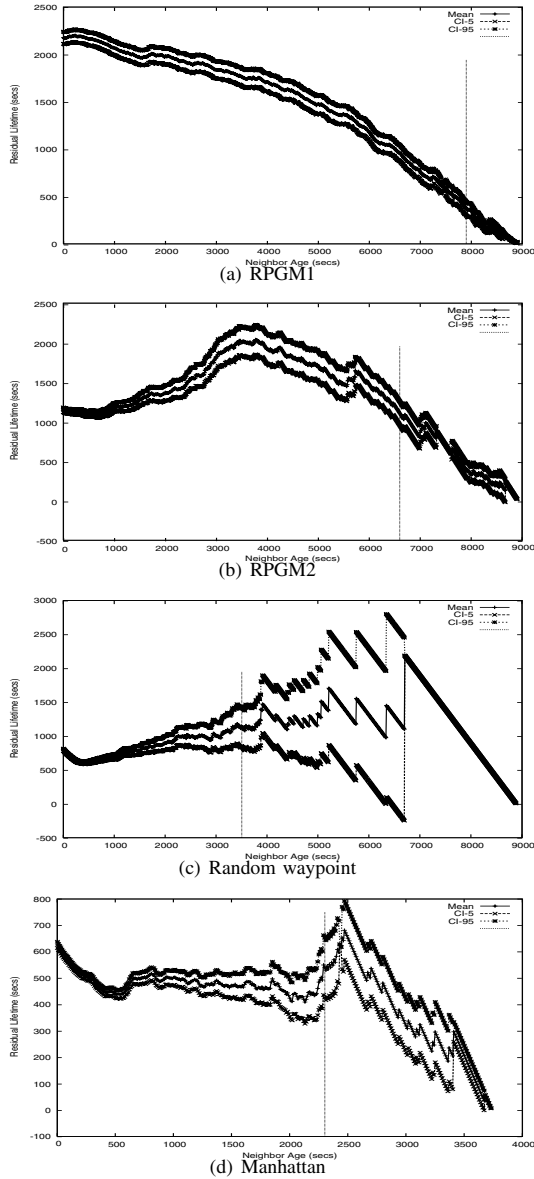


Fig. 3. Residual Lifetime, Speed 1m/sec

D. Residual Lifetime

The collected link duration values from the previous measurements are used to calculate the residual link lifetime. The residual lifetime value is computed as follows. Let l_i be the number of links with link duration i secs and R_a be the average residual link lifetime when the current link age is a .

$$R_a = \left(\frac{\sum_{i>a} (l_i * i)}{\sum_{i>a} l_i} \right) - a; \quad (1)$$

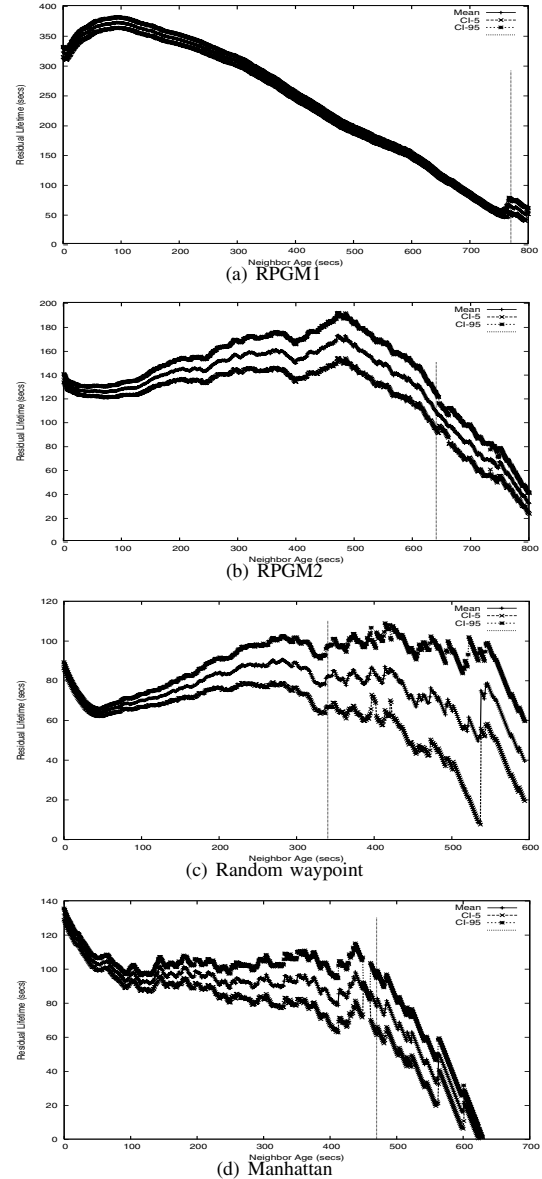


Fig. 4. Residual Lifetime, Speed: 10m/sec

In other words, the residual lifetime for a link of age a is the average lifetime of all links with durations above the age a , minus age a .

Figures 3 and 4 show the residual lifetime plots for speeds of 1 m/sec and 10 m/sec respectively. We also show the 5% and 95% confidence interval values for each link age in the plots. Using results from the preceding section, we can also obtain link durations corresponding to the 99 percentile values. The obtained values are indicated in the figures 3 and 4 as a vertical line. For example, in fig 4(a), the vertical line is at 770 secs, which corresponds to figure 2(a). In order to ensure that the simulation time is sufficiently long, the simulation duration for the 1 m/sec and 10 m/sec cases are 9000 secs and 900 secs respectively.

From Figure 3, we see that for low speed (1 m/sec), the residual lifetime decreases with increase in age for the

case of RPGM1. For the RPGM2 and random waypoint, the residual lifetime initially decreases, and then increases after some time. Finally, for the Manhattan model, the residual lifetime decreases and remains constant after some threshold. In all cases, for neighbors with sufficiently long lifetime, the residual lifetime decreases again. However, this final decrease occurs for less than 1% of the links and is not considered important. At higher speeds (10 m/sec and 30m/sec), the mobility models exhibit similar patterns.

Earlier work [4] has also noted a similar behavior where there is an initial decrease and then an increase. However, they do not consider the later decrease in link lifetime. At higher speeds, apart from RPGM1, all model confer with patterns of previous work: initial decrease and then further increase.

Based on the results obtained, we conclude that the heuristic of existing stability-based routing algorithms, for example Associativity Based Routing (ABR), of assuming that *older links are more stable* does not hold across a large spectrum of mobility speeds and models. In fact, in RPGM1 and Manhattan, the reverse is true. *Newer links are more stable*. In cases where the heuristic is correct, it is difficult to obtain a good estimation of the threshold when residual lifetime starts increasing as the threshold depends on many factors including speed and mobility pattern. Even when this threshold is available, the likelihood of finding links with long residual lifetime may also be low, making the heuristic less useful for enhancing route selection.

In the next section, we will propose the use of a simple residual lifetime predictor and a routing algorithm that can benefit from the estimation.

IV. ROUTING MECHANISM

In this section, we will describe our routing algorithm based on residual lifetime and hop-count.

Routing based on finding the minimum hop route has been used for a long time. In wireless network, the use of minimum hop route has several advantages, including simplicity, less interference, and lower consumption of network resources (bandwidth). However, since minimum hop routing does not take into account link duration, shorter route may not be the best route. On the other hand, routing algorithms based on stability, like ABR, SSA and long lifetime routing (LLR), have also proven to be advantages in some cases. However, it has been observed that stability based routing could select longer routes, resulting in poor performance caused by excessive node interference and wastage of network bandwidth. While the hop-count and stability metrics may seem contradictory at times, it is possible to combine them in order to take advantage of the strengths of both.

We term our algorithm *SHARC* for *Stability and Hop-count based Algorithm for Route Computation*. In this paper, we will consider an implementation of SHARC using the DSR routing protocol as the base routing protocol. The additions and modifications carried out on DSR are explained in the following paragraphs. We would like to emphasize that our approach can be applied to any hop-count based routing and DSR is chosen simply as one possibility.

Link Stability Estimator: In order to distribute stability information, the route-request packet of DSR is changed

to carry residual lifetime information. Every node stores the link duration values of its neighbors. By collecting this information and aggregating them into bins of 10s, each node maintains an estimate of the residual lifetime distribution using the samples collected so far and equation (1). During the initial period when the number of link duration samples collected is low, it is likely that a newer link will be chosen. From Figures 3 and 4, the choice may be a good estimate given the limited information, since residual lifetime tends to decrease initially across all mobility patterns and speeds.

Every intermediate node on receiving the request packet includes the residual lifetime value in the route request message. The path structure is changed by associating every path with an additional stability value. This stability value of the path is the sum of all the residual lifetime divided by the length of the path. The cache structure is also enhanced to maintain the stability metric along with the addresses of intermediate nodes. The route selection mechanism is incorporated in all the nodes so as to be compatible with DSR routing mechanism.

Typically, the stability value of a particular link is calculated based on the most recent (short-term) history, starting from the most recent link establishment. This is true in the majority of the stability based techniques (ABR, SSA, DSR caching schemes). In our link estimator, we consider not just the most recent link behavior, but also the connectivity history of all neighboring nodes. This helps in having a better understanding of the environment in which the node operates, making the estimates more accurate. However, it should be noted that an implicit assumption made is that the environment is homogeneous and nodes retain the same mobility patterns. Implications of heterogeneous mobility patterns are part of the future study.

The amount of memory needed for link stability estimation depends on the length of the link lifetime history kept and can be easily bounded. The accuracy and effectiveness of the estimator will be investigated using simulation.

Route Selection Algorithm: The route selection mechanism assumes that routes are stored in the cache. *min_stability* is the current value of the stability available while searching the cache. It is initialized to -1. Similarly *min_length* is the current value of the hop-count available in the process of searching the cache. It is set to the maximum hop-count possible. *dest* refers to the destination node. Let the function *findRoute* finds the route from the cache matching the destination and the function *shorter* returns the route with a smaller or equal hop-count.

```

repeat
  route = findRoute in the Cache for dest
if route.length <= min_length then
  if route.length = min_length then
    if route.stability > finalRoute.stability then
      finalRoute = route;
      min_length = route.length;
      min_stability = route.stability;
    end if
  else if route.length < min_length then
    finalRoute = route;
    min_length = route.length;
    min_stability = route.stability;

```

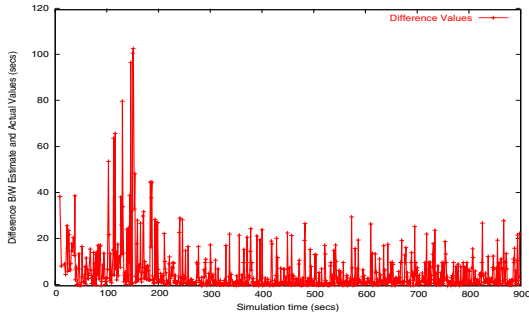


Fig. 5. Error difference of Link residual lifetime Estimator (99% = 40 secs, 95% = 25 secs, 90% = 15 secs)

```

end if
end if
until the end of the Cache

```

Basically, the algorithm tries to find the most stable route among all shortest hop routes. The algorithm can be easily extended to the case where the most stable route among all routes with hop-count not more than N hops longer than the minimum hop route, where $N \geq 0$, are chosen.

The idea behind SHARC can be explained as follow. SHARC attempts to find a route based on two objectives, path length and path stability. From section III, we know that link stability prediction is difficult and inexact. On the other hand, finding a shortest path is *precise*. In addition, there are often more than one shortest path. Hence, a good approach is to use the shortest path algorithm as the initial filter to narrow down the route selections and then use path stability, a less robust indicator, to choose the best route among the available routes.

A. Evaluation

In this section, we describe the performance evaluation of our routing algorithm. We choose to use throughput of long-lived TCP traffic and response time of web-traffic as the performance metrics because we believe they can better capture the effects of link breakages. Many research works use CBR traffic and consider delay and packet delivery ratios. However, since CBR traffic using UDP does not perform any congestion control and error recovery, we believe that the response times of short data transfers like web-traffic better reflect the impacts of link stability.

The simulation settings are similar to Section III, except that we only consider RPGM1, random waypoint and Manhattan mobility models. For throughput measurements, we consider maximum speeds of 1 m/sec and 10 m/sec. The number of TCP sources is varied from 2 to 10. However due to space limitations, plots for only 10 m/sec is shown. For response time measurements, we consider maximum speed of 1m/sec and vary the number of web-clients and web-servers pairs from 2 to 10.

First, we evaluated the residual link lifetime estimator and the result is shown in Figure 5. The mobility pattern is random waypoint, with maximum speed of 10 m/sec. The result shows that while the estimator has significantly larger error in the early phase, the error decreases rapidly after about 200s of simulation. Over the entire 900s of simulation,

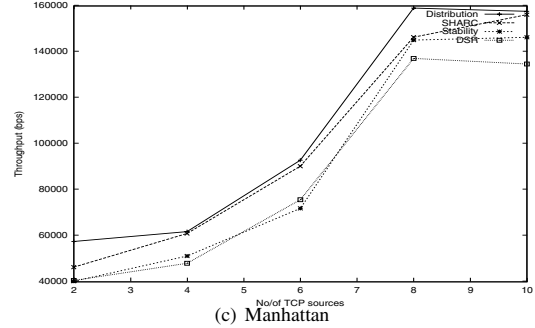
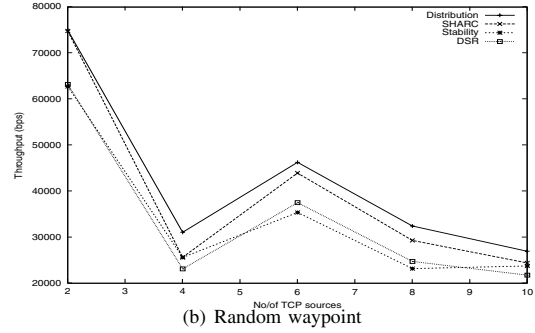
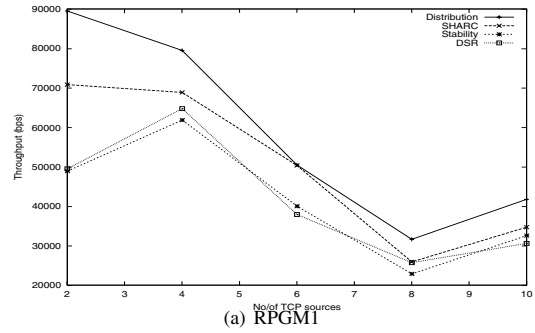


Fig. 6. Throughput Vs No/ of Sources, 10m/sec

99% of the errors are less than 40s and 90% are less than 15s. Therefore, the estimator provides reasonably robust input for enhancing route selection. We have also evaluated the estimation for other mobility patterns and speeds. The results are similar.

For our evaluations of SHARC, we compare SHARC with three other algorithms. As we have combined both hop-count and stability into the route computation, we use as baseline, the performance of *DSR*, which is hop-count based, and a DSR like algorithm, which uses the stability metric only. We label the performance of the algorithm that uses stability metric as *stability* in the plots. We have also simulated DSR with the best of the link caching schemes [8]. As there was no significant improvement compared to the baseline DSR simulated, we will not show the results here.

The third algorithm, which we label as *Distribution* in the plots, is similar to SHARC except for how residual lifetime values are obtained. Instead of estimating the residual lifetime values from past history, the values are obtained from previous simulations with the same parameters including the random seeds used. The lifetime distributions are embedded in all the nodes at the start of the simulation so that when

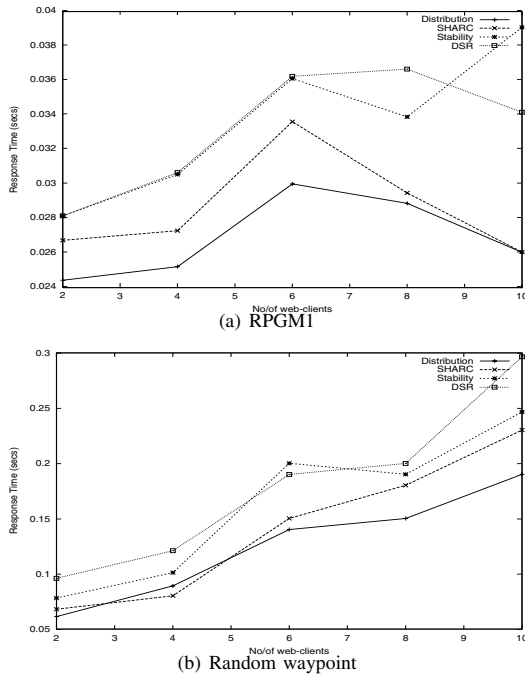


Fig. 7. Response Time Vs No/of Sources, 1m/sec

a node receives a request packet, it can base its decision on the exact neighbor lifetime distributions.

Figure 6 shows the throughput of long-lived TCP with varying number of sources and correspond to maximum speed of 10m/sec. It can be seen from the figure that in most of the cases, SHARC always performs better than the baseline cases of *stability* and *DSR*. At low speed (1 m/sec), we found that the improvement varies from 10% to 30%. However at higher speeds (10 m/sec), we can see that the improvement varies from 10% to 45%. It is interesting to note that in many cases, an algorithm that takes into account stability only performs worst than DSR. This is consistent with previous works like SSA, and work by Sridhar et. al., [10]. *However, when stability is added as an enhancement to the hop-count metric, performance can be improved.*

The plots also show that throughput values of SHARC are closer to the algorithm that operates with complete knowledge of residual lifetimes. While the estimation error is large in the beginning of the experiment, as the simulation time progresses and node collects more and more information of neighbor lifetimes, the estimates can be much better. Nevertheless, even with the rough approximations obtained using historical data, it is possible to perform close to the ideal algorithm in some cases. The performance gap is between 1% to 10%. The results show that it is not essential to have good approximation of the residual lifetimes to obtain good performance, a rough approximation suffice.

Next, in order to evaluate the performance with respect to delay values, we consider the response time of web-traffic. We randomly select a node to be a HTTP server, and associate a cache node with it. We use the web-traffic model of ns-2 [14], in which pages are maintained as page pool, with configurable parameters such as page objects, expiry

time, request interarrival times etc. In our simulation we use page with just a single object, with average expiry time of 5 secs. The interarrival time of request from the client has an average value of 10 secs.

Figure 7 shows the response time with varying number of web-clients. Figure 7(a) corresponds to single group (RPGM1) mobility model, whereas Figure 7(b) corresponds to random waypoint. Both the models operate in low-speed (1m/sec). The results are similar to the throughput plots. SHARC has lower delay values compared to both the baseline algorithms. The improvements are 10% to 40% over DSR and 5% to 50% over a stability-only algorithm.

V. CONCLUSIONS

In this work, we study link stability with respect to different mobility models and the impact of link stability on routing. From the simulations, we found that residual link lifetime is a function of current link age, mobility speed and mobility pattern, and does not vary monotonically with age. Nevertheless, the residual lifetime can still be useful if it can be estimated approximately. We proposed a stability and hop-count based routing algorithm, called *SHARC*, which finds the most stable route among the set of shortest hop routes. Performance evaluation of SHARC shows that it performs better than purely stability based and purely hop count based algorithms in terms of throughput of long-lived flows and response time of short data transfers.

REFERENCES

- [1] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. In *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2002.
- [2] C-K. Toh. "Associativity-Based Routing for Ad Hoc Mobile Networks," *International Journal on Wireless Personal Communications*, Vol. 4 No. 2, March 1997, pp. 103-139.
- [3] R. Dube, C. D. Rais, K. Y. Wang, S. K. Tripathi, "Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks," *IEEE Personal Communications Magazine*, vol. 4, no. 1, Feb. 1997, pp.36-45.
- [4] M. Gerharz, C. D. Waal, M. Frank, P. Martini, "Link stability in mobile wireless ad hoc networks," *Proc. of 27th IEEE Conference on Local Computer Networks*, pp. 30-39, Tampa, Florida, Nov 2002.
- [5] S. Agarwal, A. Ahija, J. P. Singh, R. Shorey, "Route-lifetime Assessment Based Routing protocol for Mobile Ad Hoc Networks," *Proc. IEEE International Conference on Communications*, 2000, Volume 3, pp. 1697-1701.
- [6] N. Sadagopan, F. Bai, B. Krishnamachari and A. Helmy, "PATHS: Analysis of path duration distributions in MANET and their impact on routing protocols", in *Proc. ACM MobiHoc 2003*, Annapolis, MD, June 2003, pp. 245-256.
- [7] Z. Cheng and W. Heinzelman, "Exploring long lifetime routing in ad hoc networks," *Proceedings of the 7th ACM international symposium on modeling, analysis and simulation of wireless and mobile systems (MSWIM 2004)*, Oct., 2004.
- [8] Hu Y-C., Johnson D. B., "Caching Strategies in On-Demand Routing Protocols for wireless Ad Hoc Networks," in *proceedings of MOBI-COM 2000*, August 2000, Boston, MA, USA.
- [9] David B. Johnson, David A. Maltz, and Josh Broch. "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc networks", in *Ad Hoc Networking*, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.
- [10] Sridhar K. N., Lillykutty Jacob, "Interplay of Service differentiation and routing protocols: A performance study," in *proceedings of IPCCC 2004*, April 2004, Phoenix.
- [11] <http://toilers.mines.edu/>
- [12] <http://mile.usc.edu/important/>
- [13] <http://web.informatik.uni-bonn.de/IV/mitarbeiter/dewaal/bonnmotion/>
- [14] <http://www.isi.edu/nsnam/ns/>