Integrating Stability Estimation into Quality of Service Routing in Mobile Ad-hoc Networks

Weiying Zhu[a], *Student Member*, *IEEE*, Min Song[a], *Member*, *IEEE*, and Stephan Olariu[b], *Member*, *IEEE*
[a]Department of Electrical and Computer Engineering, [b]Department of Computer Science
Old Dominion University
Norfolk, VA 23529

Abstract—**One of the notoriously difficult problems in Quality of Service (QoS) routing in Mobile Ad-hoc NETworks (MANET) is to ensure that the established path for a connection does not break before the end of the data transmission. This paper addresses the issue of reducing path breakage during data transmission, even if geographic location information is not available. Using delay-constrained QoS routing as an example, we propose a novel algorithm that we call ticket-based probing with stability estimation (TBP-SE) as an enhancement for the multi-path distributed QoS routing scheme proposed in [4]. Models are created to estimate relative link and path stability. During path discovery, the estimated relative stability is used to direct tickets along paths featuring high stability. Among multiple detected paths, the one with the highest relative stability is selected. Through extensive simulations, we show that our algorithm significantly improves the stability of established paths in terms of average relative path stability, path breakage speed, and percentage of data transmissions completed before path breakage.**

Index Terms—**Quality of Service, mobile ad-hoc networks, QoS routing, path stability, wireless multimedia.**

I. INTRODUCTION

IN the past decade we have witnessed a phenomenal growth in the deployment of portable wireless devices and related services, including wireless multimedia. MANET, an archtypical infrastructure-less wireless packet network, enables these wireless devices to communicate with each other without the help of base stations or other pre-existing infrastructure. While a mobile node can communicate directly with the nodes lying within its transmission range, communication with the mobile nodes outside of the transmission range must necessarily be multi-hop and require the establishment of communication paths. It is well known that most of the multimedia applications require the establishment of communication paths that satisfy a number of negotiated parameters (such as delay or bandwidth), usually referred to QoS guarantees. Due to the dynamic nature of the network topology and imprecise network state information, a lot of problems remain before more efficient solutions are found for QoS routing in MANET. One of the problems is that the established path for a connection request may break before the end of data transmission. Under the assumption of constant velocity, the dynamic behavior of the communication links in a random mobility environment is theoretically analyzed in [16].

A. *State of the art*

In order to mitigate the problem of path breakage, the idea of soft QoS [3, 4, 10, 20, 21] was adopted in QoS provisioning in wireless networks and, in particular, in MANET. Soft QoS allows the negotiated QoS parameters to be violated for short, transient time periods. When the established path is broken, the source node is notified. Either a new connection request is initiated to re-transmit data or the path is rerouted to continue the interrupted data transmission. However, frequent breakages of established paths result in large delay and/or even transmission failure from the multimedia application point of view, and in increased routing message overhead and wastage of network resources.

The problem of frequent path breakage in MANET has attracted a great deal of well-deserved attention in the recent literature. In the context of geographic (location-based) routing [18], a scheme is proposed in [17] to predict future paths before existing paths break. This scheme can avoid path re-computation delay. But it does not reduce path breakage so that some problems such as transmission failure and big routing message overhead still exist. Additionally, geographic locations need to be obtained via GPS or similar devices and exchanged through periodically message update, which introduces extra system requirement and network resource in the context of non-geographic routing.

The QoS routing over cluster-based routing [22] is studied in [1]. The network is partitioned into logical clusters. Long lifetime of paths is pursued twofold in [1]. Firstly, the node with a larger absolute speed has a smaller chance to be elected as a cluster header. Paths are constructed along the virtual backbone formed by cluster headers. Secondly, among multiple discovered paths, the one with the maximal remaining power is selected. However, these two methods can not minimize path breakage efficiently. One reason is that the stability of a path is determined by the relative speeds between nodes forming the path instead of their absolute speeds. The other reason is that the effect of path breakage due to mobility is ignored during path selection. Moreover, geographic locations are also indispensable for cluster-based routing.

In [19], Wang et al. assign “Link-Stable-Time” for a link and “Path-Stable-Time” for a path. If the stable time of a path

1-4244-0476-2/06/$20.00 ©2006 IEEE.

is going to be expired, the source node will discover a new path in advance. However, no detailed method is given to estimate "Link-Stable-Time". Actually, because of the various movement models of mobile nodes, it is very hard to predict the exact life time of a path. The discovery in advance will be too late if the path breaks before "Path-Stable-Time", or will result in extra routing messages if the path breaks after the "Path-Stable-Time".

In the framework proposed in [6], a set of paths are detected to transmit data simultaneously. Each path is allocated a certain volume of data such that the path's lifetime is not less than the transmission time. A new set of paths are preemptively rediscovered just before the last set of paths complete data transmission. Both the correct data allocation and the timely path rediscovery highly rely on the accurate estimation on lifetime of links and paths. However, due to various mobility models and non-linear relationship between signal strength and distance, the lifetime estimation method proposed in [6] is not accurate most of the time in the real world.

B. *Our contributions*

It is still an open issue to reduce path breakage during data transmission in QoS routing in MANET, especially if no geographic location information is available. Ticket-based probing (TBP) [4] is an interesting multi-path distributed QoS routing scheme which, however, only considers a very simple and inefficient way to assess path stability.

The main contribution of this paper is to provide a significant extension of TBP. Specifically, we propose a new algorithm that we call Ticket-Based probing with Stability Estimation (TBP-SE) that integrates the stability criterion into path discovery and path selection. We propose models and metrics to estimate the relative stability for a link between two neighboring mobile nodes and the relative stability for a path between source and destination nodes. During path discovery, the estimated relative link and path stability is used to direct tickets along the paths with high stability. When multiple feasible paths are detected, the one with the highest relative stability is selected.

Extensive simulations are performed to compare TBP and TBP-SE. It is shown that TBP-SE does significantly improve the stability of selected paths in terms of average relative path stability, path breakage speed, and percentage of data transmissions completed before path breaking.

The remainder of this paper is organized as follows. Section II describes the delay-constrained QoS routing problem, TBP, and the limitations of TBP. Section III specifies the models to estimate the relative link stability and the relative path stability. Details of the TBP-SE algorithm are given in Section IV. Section V presents the simulation study. Finally, Section VI offers concluding remarks and open research problems.

II. DELAY-CONSTRAINED QOS ROUTING AND TICKET BASED PROBING

The set of mobile nodes in a MANET is denoted by V. It is assumed that all nodes have the same transmission range. The node that lies within the transmission range of node i is a neighbor of node i. Each node detects its neighbors through a neighbor-discovery protocol [13, 15]. Periodically, a beacon message is broadcasted by a node and received by all the nodes within its transmission range. By receiving the beacon messages, node i maintains the set of its neighbor nodes, N_i. The link between two neighbor nodes i and j, is denoted by $L_{i,j}$.

A. *QoS Routing with Delay-Constraint*

Given that the delay requirement is D, the goal is to detect a path from source node s to destination node d such that the delay on the path does not exceed D. Unless specified otherwise, path P represents a path with source node s, destination node d, and intermediate nodes r_1, r_2, …, and r_n in between, i.e., $P = s \to r_1 \to r_2 \to \ldots \to r_n \to d$. We have

$$delay(P) = delay(L_{s,r_1}) + \sum_{k=1}^{n-1} delay(L_{r_k,r_{k+1}}) + delay(L_{r_n,d}) \,. \quad (1)$$

A path that satisfies the delay-constraint is referred to as *feasible*.

B. *Ticket-Based Probing*

In general, in addition to the delay constraint in path discovery, there might be some other optimization objectives that need to be satisfied, such as least number of hops, least link utilization, and so on. When there are multiple objectives, it becomes an NP-complete problem [7]. In [4], a novel multi-path distributed QoS routing scheme, TBP, was proposed to detect the path that satisfies two objectives: delay-constraint and low cost. Cost is an abstract concept and could be a hop counter, a function of link utilization, or other QoS state metrics that should be as small as possible. For path P, cost is calculated as follows:

$$cost(P) = cost(L_{s,r_1}) + \sum_{k=1}^{n-1} cost(L_{r_k,r_{k+1}}) + cost(L_{r_n,d}) \,. \quad (2)$$

With TBP, probes carrying tickets are generated by the source node and travel in the network. Every probe that successfully arrives at the destination node detects one feasible path. The abstract of TBP is given as follows.

- Node i keeps the up-to-date local state information, delay($L_{i,j}$) and cost($L_{i,j}$), $\forall\, j \in N_i$.
- At time t, node i also maintains the imprecise end-to-end state information from node i to node k, $\forall\, k \in V$. The state variables include the minimum end-to-end delay, $D_{i\to k}(t)$, the least end-to-end cost, $C_{i\to k}(t)$, and the estimated maximum change of the end-to-end delay before the next update, $\Delta D_{i\to k}(t)$, $\forall\, k \in V$. The end-to-end state information is periodically updated by a distance-vector protocol [14].
- For a connection request, the source node generates a certain number of yellow tickets and a certain number of

green tickets.

- At a source or intermediate node, tickets are distributed among its neighbor nodes based on the accumulated path delay, the accumulated path cost, and the local and end-to-end state information. Toward the destination node, yellow tickets should be sent to the paths with low delay; and green tickets should be sent to the paths with low cost. If a neighbor node is allocated at least one ticket, a probe carrying all the tickets allocated to this neighbor node is sent to this neighbor node. If no feasible path exists, the delivery of probes is terminated.
- Finally, each of the probes that successfully arrive at the destination node detects a feasible path. Among them, the path with the lowest cost is selected.

C. *Limitation of Ticket-Based Probing*

In TBP [4], in order to reduce the probability that a path breaks when the topology changes, a link is tagged as either *transient* if it is just formed or *stationary* if it remains unbroken for a time period. Tickets are distributed only among stationary links whenever possible. However, this method does not efficiently reduce the probability that the selected path breaks because of three reasons.

First, it is based on an assumption that newly-formed links are more likely to be broken than links that have already existed for some time. This is certainly not always true. For example, node j just enters the transmission range of node i and moves slowly toward node i. Meanwhile, node k is moving away from node i and is going to leave the transmission range of node i soon. Obviously $L_{i,k}$ is much more likely to be broken than $L_{i,j}$ although $L_{i,j}$ is a new link and $L_{i,k}$ has existed for some time.

Second, links are only categorized in two groups. From the stability point of view, links belonging to the same group are not differentiated. For instance, with node i as reference, its neighbor node j is static and its neighbor node k is moving away. Since both $L_{i,j}$ and $L_{i,k}$ are stationary links, $L_{i,j}$ is not given any preference although $L_{i,k}$ is more likely to be broken than $L_{i,j}$.

Third, no QoS state metrics are used to indicate the accumulated path stability and provide reference for path selection. While selecting a path among multiple feasible paths, preference is not given to the path consisting of less transient links.

Therefore, one significant limitation of TBP is that the selected path may break soon due to the topology change while the feasible paths that exist longer are not selected. Then the transmission of the multimedia application is susceptible to be interrupted due to the breaking of the selected path although there might be other alternative paths that exist longer.

III. ESTIMATING THE RELATIVE PATH STABILITY

Given that the topology of a MANET is dynamic, it is very important to select among the feasible paths for a connection, a path that maximizes stability. As previously mentioned, path breakage during data transmission is apt to result in increased delay, transmission failure, significant routing message overhead, and wastage of network resource. Compared with other optimization objective, such as low link utilization, high path stability is far more elusive. Indeed, due to the unpredictable nature of node mobility, it is very difficult to predict the exact probability that a link or path will be broken in the near future. Nonetheless, based on historical and current distance between nodes, it is possible to estimate the relative stability of links and paths so as to provide guidelines for path discovery and selection.

In order to estimate the relative stability, the distance between neighbor nodes needs to be calculated first. For GPS-enabled nodes, their distance can be easily calculated using their location. For nodes without GPS, we propose a different methodology. At time t, when node i receives a beacon from its neighbor, node j, it measures the signal strength, $P_{i,j}(t)$. By knowing $P_{i,j}(t)$, based on the radio propagation model, an estimate of the distance between nodes i and j at time t, $Dis_{i,j}(t)$, can be calculated [5, 8, 12]. Assuming that T is the period with which beacon messages are broadcast, at time t, node i updates both $Dis_{i,j}(t - T)$ and $Dis_{i,j}(t)$, $\forall\, j \in N_i$.

A. *Estimating the Link Stability Rate*

We define a new QoS state metric, the *link stability rate*, to capture the relative stability of a link. For each of its neighbors, node i keeps the up-to-date local state information of link stability rate. The link stability rate of $L_{i,j}$ at time t is denoted as $S_{i,j}(t)$, where $j \in N_i$. Before giving the estimation model for $S_{i,j}(t)$, two thresholds for the distance need to be defined. One is the maximum distance, Dis_{max}, between two neighbors. When $Dis_{i,j}(t) > Dis_{max}$, nodes i and j cannot receive messages from each other and, consequently, the link $L_{i,j}$ is broken. Normally, Dis_{max} equals the transmission range of the nodes of the MANET. We have $Dis_{i,j}(t) \in [0, Dis_{max}]$, $\forall\, j \in N_i$. The other threshold is the minimum distance, Dis_{mi}, between two neighboring nodes. When $Dis_{i,j}(t) \le Dis_{min}$, nodes i and j are sufficiently close to each other and the link $L_{i,j}$ is sufficiently stable for the link stability rate to be considered maximal. We can see that $Dis_{min} \in [0, Dis_{max})$.

When a link is sufficiently stable, its link stability rate is set to the maximal value, η, which is a non-negative value less than 1. The minimal value of link stability rate is 0. At time t, $\forall\, j \in N_i$, a node i calculates $S_{i,j}(t)$ according to the following rules:

Rule 1: If $Dis_{i,j}(t) \le Dis_{min}$, then nodes i and j are very close and $L_{i,j}$ is considered to be sufficiently stable so that $S_{i,j}(t) = \eta$;

Rule 2: If node j is a new neighbor of node i, that is, at time $t - T$, node j is not the neighbor of node i, then $S_{i,j}(t) = 0$;

Rule 3: If $Dis_{i,j}(t) \le Dis_{i,j}(t - T)$, then nodes i and j are approaching each other and $L_{i,j}$ is considered to be sufficiently stable so that $S_{i,j}(t) = \eta$;

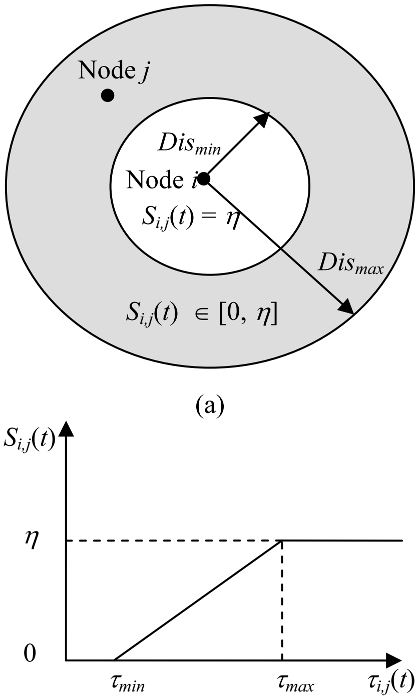
Rule 4: Let $\tau_{i,j}(t) = \dfrac{Dis_{max} - Dis_{i,j}(t)}{Dis_{i,j}(t) - Dis_{i,j}(t-T)}$ represent the expected duration, at time t, that the link $L_{i,j}$ remains unbroken. Two positive threshold values, τ_{min} and τ_{max}, are defined for $\tau_{i,j}(t)$, where $\tau_{max} > \tau_{min}$.

Rule 5: If $\tau_{i,j}(t) \geq \tau_{max}$, then $S_{i,j}(t) = \eta$;

Rule 6: If $\tau_{i,j}(t) \leq \tau_{min}$, then $S_{i,j}(t) = 0$;

Rule 7: $S_{i,j}(t) = \dfrac{\tau_{i,j}(t) - \tau_{min}}{\tau_{max} - \tau_{min}} \times \eta$.

The above rules are applied in sequence. Once $S_{i,j}(t)$ is determined, the remaining rules will not be applied. The estimation of $S_{i,j}(t)$ is illustrated in Figure 1, where (a) explains Rule 1, and (b) explains Rules 4, 5, 6, and 7.



(a)

(b)

Fig. 1. Estimation of the link stability rate for $L_{i,j}$

B. Estimating the Path Stability Rate

Another QoS state metric, *the path stability rate*, is meant to indicate the relative stability of a path between a source node and a destination node. As we know, if any one of the links that build up the path is broken, the whole path is broken. If a path is non-broken, all of its links must be non-broken. And the status of a link is independent of the status of other links. Therefore, for path P, the probability that a path is non-broken equals to the product of the probabilities that its links are non-broken, i.e.

$$\begin{aligned} &\Pr\{P \text{ is non-broken}\} \\ &\quad = \Pr\{L_{s,r_1} \text{ is non-broken}\} \times \\ &\quad \prod_{k=1}^{n-1} \Pr\{L_{r_k,r_{k+1}} \text{ is non-broken}\} \times \\ &\quad \Pr\{L_{r_n,d} \text{ is non-broken}\}. \end{aligned} \tag{3}$$

Based on the above relationship, the path stability rate of P at time t, $S_P(t)$, is defined as follows:

$$S_p(t) = S_{s,r_1}(t) \times \prod_{k=1}^{n-1} S_{r_k,r_{k+1}}(t) \times S_{r_n,d}(t) . \tag{4}$$

In addition, since the link stability rate is non-negative and less than one, this definition also makes a preference for the path with less number of hops.

IV. TICKET BASED PROBING WITH STABILITY ESTIMATION

In this section, we describe the modification to TBP in order to integrate stability estimation. The modified algorithm is called ticket based probing with stability estimation. To specify the algorithm, new QoS state information need be introduced; a new group of tickets, red tickets, need to be generated and forwarded; and, at the destination node, among all feasible paths, the one with the highest path stability rate is selected. Unless specified otherwise, the delay requirement of the connection request is denoted by D, the source node of the connection request is denoted as node s, and the destination node of the connection request is denoted as node d.

A. State Variables for Stability

At a generic node i in the network, two new groups of state variables are introduced to specify stability. One group of variables, $S_{i,j}(t)$, $\forall\ j \in N_i$, records the up-to-date link stability rates for outgoing links, which is updated by a neighbor-discovery protocol [13, 15]. The other group of variables, $S_{i\to d}(t)$, $\forall\ d \in V$, records the estimated largest end-to-end path stability rate from node i to node d, which is updated by a distance-vector protocol [14].

B. Determining the Number of Red Tickets

Similarly to the yellow tickets for detecting feasible paths and green tickets for detecting low-cost paths in TBP, a certain number of red tickets are generated to maximize the probability that a feasible path with high stability is detected. Red tickets prefer paths with higher path stability rates and have less chance to satisfy the delay-constraint than yellow tickets.

The number of red tickets is decided according to both D and the estimated end-to-end delay from node s to node d. At time t, given the maximum number of red tickets, Λ, and the threshold value specifying the sufficiently-large range for the delay requirement, θ, the number of red tickets, R_0, is determined as follows:

If $D \geq \theta \times (D_{s\to d}(t) + \Delta D_{s\to d}(t))$, then $R_0 = 1$. Because the delay requirement is sufficiently large, one red ticket suffices to detect a feasible path with high stability.

If $\theta \times (D_{s\to d}(t) + \Delta D_{s\to d}(t)) > D \geq D_{s\to d}(t)$, then

$$R_0 = \left\lceil \frac{\theta \times (D_{s\to d}(t) + \Delta D_{s\to d}(t)) - D}{\theta \times (D_{s\to d}(t) + \Delta D_{s\to d}(t)) - D_{s\to d}(t)} \times (\Lambda - 1) + 1 \right\rceil .$$

Because the delay requirement is still larger than the estimated minimum end-to-end, as the delay requirement becomes smaller, the number of red tickets is increased in order to increase the probability that a feasible path with high stability is detected.

If $D_{s\to d}(t) > D \ge D_{s\to d}(t) - \Delta D_{s\to d}(t)$, then $R_0 = \left\lceil \frac{D - (D_{s\to d}(t) - \Delta D_{s\to d}(t))}{\Delta D_{s\to d}(t)} \times \Lambda \right\rceil$. Because the delay requirement is smaller than the estimated minimum end-to-end, it is considered *stringent* and the main objective is to detect a feasible path instead of optimizing path stability. As the delay requirement becomes smaller, the number of yellow tickets increases. Hence, the number of red tickets decreases in order to reduce the routing overhead.

If $D < D_{s\to d}(t) - \Delta D_{s\to d}(t)$, then $R_0 = 0$. If the delay requirement is too stringent to be satisfied, then no ticket is issued and the connection request is rejected.

The above rules for calculating R_0 are illustrated in Figure 2.

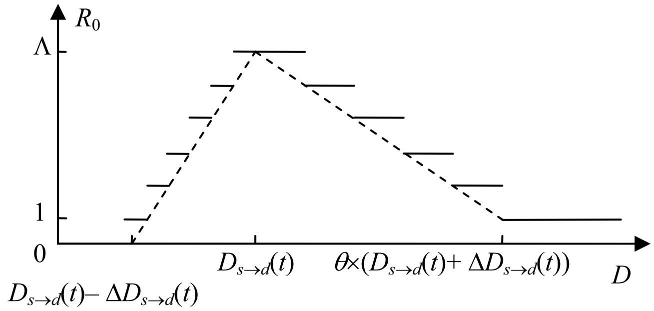


Fig. 2. The number of red tickets as a function of delay requirement

C. Distributing Red Tickets among Neighboring Nodes

When a source node generates a certain number of red tickets or an intermediate node receives a certain number of red tickets carried by a probe, it forwards them toward the destination node by selecting a set of candidate neighbor nodes. The guideline of distributing red tickets among the neighbor nodes is to forward red tickets to the paths with high stability. Given that the source node or the intermediate node is node *i*, at time *t*, a certain number, *X*, of red tickets generated or received by node *i* are distributed among the neighbor nodes of node *i* in three steps.

First, the candidate neighbor nodes that are eligible for receiving tickets need to be selected. When the probe moves toward the destination node, it records the accumulated path delay, *delay*(P^{tr}), and the accumulated path stability rate, *stab*(P^{tr}), of the path it has traversed so far. The set of nodes that the probe has traversed is denoted as V^{tr}. Notice that if node *i* is a source node, *delay*(P^{tr}) = 0, *stab*(P^{tr}) = 1, and V^{tr} is empty. A candidate neighbor node, for instance, node *j*, should satisfy two criteria: (1) the probe has not traversed node *j*; and, (2) the minimum path delay from the source node to the destination node via node *j* is expected to be less than the delay requirement. For the set of the candidate neighbor nodes that are eligible for receiving tickets, R_i, we have:

$$R_i = \{ j \mid delay(P^{tr}) + delay(L_{i,j}) + D_{j\to d}(t) - \Delta D_{j\to d}(t) \le D,\ j \in N_i - V^{tr} \}\ . \tag{5}$$

If R_i is empty, then the forwarding of tickets is terminated at node *i* and no probe is sent out.

Second, the red tickets need to be distributed among the candidate neighbor nodes. Since the purpose of red tickets is to detect a feasible path with high stability, more red tickets are sent to the neighbor node on the path with higher path stability rate. For the number of red tickets distributed to a candidate neighbor node, X_j, $\forall j \in R_i$, we have:

$$\overline{X}_j = \frac{S_{i,j}(t) \times S_{j\to d}(t)}{\sum_{j' \in R_i} (S_{i,j'}(t) \times S_{j'\to d}(t))} \times X\ . \tag{6}$$

$X_j = \lceil \overline{X}_j \rceil$ for larger $\overline{X}_j$, $X_j = \lfloor \overline{X}_j \rfloor$ for smaller $\overline{X}_j$, and $\sum_{j \in R_i} X_j = X$.

Third, a new probe is sent to node *j* if $X_j > 0$, $\forall j \in R_i$. This probe carries X_j red tickets and other colored tickets allocated to node *j* if any. The accumulated variables are updated as follows:

- *delay*(P^{tr}) = *delay*(P^{tr}) + *delay*($L_{i,j}$);
- *stab*(P^{tr}) = *stab*(P^{tr}) × $S_{i,j}(t)$;
- $V^{tr} = V^{tr} + \{i\}$.

D. Path Selection at the Destination Node

When a probe arrives at the destination, a path is detected. Notice that by virtue of (5), the path satisfies the delay-constraint. Recall that, as stated in Section 2, it is far more important than other optimization objectives for a path to have high stability. Therefore, among all the feasible paths, the one with the highest path stability rate is selected as the primary path. As a result, there is no need to generate and forward green tickets for detecting low cost paths.

V. SIMULATION

A. The Mobility Model

In our simulations, the *random waypoint* model [2, 9, 11] is adopted to simulate the movement of mobile nodes. The random waypoint model is a commonly used mobility model in MANET [9]. Every node in a two-dimension $L \times W$ rectangular area is assigned an initial location, chosen independently and uniformly. With each of the mobile nodes, a direction, a speed, and a moving duration are randomly assigned. The direction is chosen uniformly within [0, 2π); the speed is chosen uniformly within $(0, 2\times V_{mean}]$; and the moving duration is chosen uniformly within $(0, 2\times T_{mean}^{mov}]$. After the node moves for the assigned duration, it pauses. The pause duration is chosen uniformly within (0, $2\times T_{mean}^{pause}$]. After that, a new direction, speed and moving duration are selected and the pattern iterates.

The mobility rate is defined as the ratio of T_{mean}^{mov} to (T_{mean}^{mov} + T_{mean}^{pause}). The delay requirement is uniformly distributed within [D_{min}, D_{max}]. For every connection request, the data transmission duration is chosen uniformly within [T_{min}^{trans}, T_{max}^{trans}].

B. Performance Metrics

Four performance metrics are defined as follows for performance evaluation and comparison:

- Average path stability rate $\triangleq$ $\frac{\text{total path stability rate of all seleted paths}}{\text{number of accepted connection requests}}$. A connection request is accepted if one path is selected for it. The average path stability rate indicates how efficiently paths with high stability rate are detected by forwarding red tickets. However, it cannot indicate the efficiency of estimation on the relative link and path stability. Hence, the percentage of non-broken paths and the success ratio are introduced.
- Percentage of non-broken paths (t) $\triangleq$ $\frac{\text{number of selected paths that exist for at least time } t}{\text{number of selected paths}}$.

  Due to the topology change, a path is going to break some time after it is selected for a connection request. If a path remains non-broken time t after it is selected, it exists for at least time t. The increase of the percentage of non-broken paths reflects the decrease of the path breakage speed.
- Success ratio $\triangleq$ $\frac{\text{number of connections whose data transmissions are completed}}{\text{number of connection requests}}$.

  In [4], the success ratio is defined as the ratio of the number of accepted connection requests to the number of connection requests. However, as we know, in MANET, even if a feasible path is selected for a connection request, data transmission might fail due to the path breaking. Only if the selected path remains non-broken until data transmission is completed, the connection request is going to succeed. The success ratio is re-defined here.
- Average message overhead $\triangleq$ $\frac{\text{number of messages sent}}{\text{number of connection requests}}$. When a node sends a probe to over a link, one message is counted. For instance, if a probe arrives at the destination node via a path consisting of l links, l messages are counted for detecting this path. The average message overhead evaluates the message overhead per connection request for QoS routing.

C. Simulation Results

Like TBP, neither the flooding algorithm nor the shortest-path algorithm considers the factor of stability while selecting a path. Compared with the flooding algorithm and the shortest-path algorithm, TBP is a better compromise between the success ratio and the message overhead. Hence, we only use TBP as a comparison base in this paper. Extensive simulations were performed to compare the performance of TBP-SE with TBP. All simulations were run for 10,000 seconds to collect data for statistical elaboration. 2000 independent connection requests were generated and the moment that a connection request was initiated is randomly chosen. This subsection describes the typical simulation results. The values of the system parameters are illustrated in Table 1.

TABLE 1. VALUES OF SYSTEM PARAMETERS

System parameter	Value
L	20 meters
W	20 meters
Number of mobile nodes	50
Dis_{max} (Transmission range)	5 meters
Dis_{min}	0.3 meter
τ_{min}	10 seconds
τ_{max}	500 seconds
η	0.9
Φ (Maximum number of yellow tickets)	5
Ω (Maximum number of green tickets)	4
Λ	4
θ	1.7
V_{mean}	0.3 meter/second
T_{mean}^{mov}	50 seconds
Average link delay	50 milliseconds
Imprecise rate of link delay	0.1
Average link cost	100
Imprecise rate of link cost	0.1
D_{min}	30 milliseconds
D_{max}	1000 milliseconds
T_{min}^{trans}	0.5 second
T_{max}^{trans}	10 seconds

Figure 3 compares the average path stability rates of TBP-SE and TBP as a function of mobility rate. As the mobility rate increases, the average path stability rates of both algorithms decrease while TBP-SE can always detect paths with much higher stability rate than TBP.

Fig. 3. Average path stability rate as a function of mobility rate

The percentage of paths that remain non-broken time *t* after they were selected is shown in Figure 4 as a function of *t*. Three groups of curves are plotted with mobility rate (MR) being low (0.1), mild (0.4), and high (0.8), respectively. In each group, when *t* equals to 0, the percentage of non-broken paths of TBP-SE is the same as TBP. The percentage of non-broken paths of TBP drops much faster than that of TBP-SE as *t* increases. Before both of them drop to zero, for a given *t*, the percentage of non-broken paths of TBP-SE is obviously bigger than that of TBP. This shows that TBP-SE does detect paths that remain non-broken for longer duration than TBP.

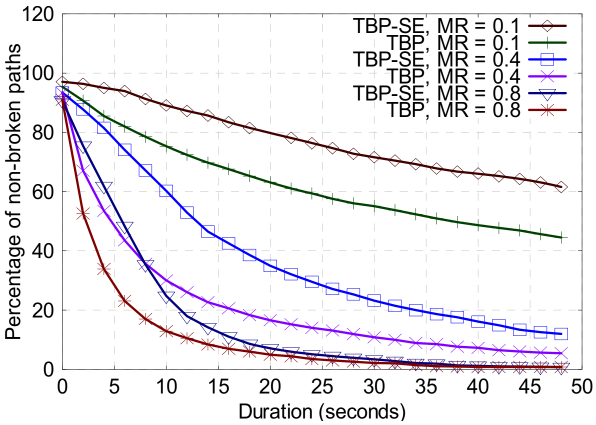


Fig. 4. Percentage of non-broken paths as a function of duration

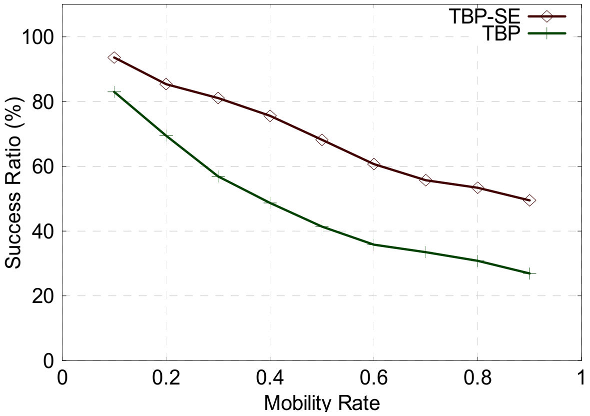


Fig. 5. Success ratio as a function of mobility rate

The success ratio as a function of mobility rate is illustrated in Figure 5. Compared with TBP, TBP-SE significantly increases the probability that data transmission is completed before the selected path breaks. This improvement is very helpful for decreasing rerouting delay, transmission failure, and wastage of network resources.

Figure 6 plots the average message overhead as a function of mobility rate. We can see that the curves of TBP-SE and TBP are very close to each other. From the QoS routing message overhead per connection request point of view, TBP-SE and TBP are similar to each other. However, considering that TBP-SE reduces transmission interruption due to topology change, it reduces the number of connection requests for re-transmission and/or re-routing. Thus the overall QoS routing message overhead decreases.

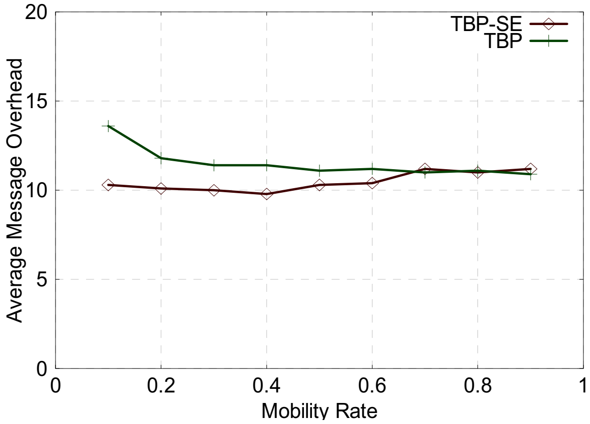


Fig. 6. Average message overhead as a function of mobility rate

VI. CONCLUDING REMARKS AND OPEN PROBLEMS

We have proposed models to estimate the relative stability of links and paths. Based on stability estimation, a distributed QoS routing scheme, TBP, is enhanced. The factor of stability is integrated into the procedure of path discovery and selection such that the path with longer lifetime can be found for a QoS connection request. The delay-constrained QoS routing problem is used as an example to present our algorithm, TBP-SE. Four performance metrics are defined to compare TBP-SE and TBP. Extensive simulations indicate that TBP-SE significantly increases the average path stability rate, decreases the path breakage speed, and increases the success ratio. Meanwhile, the average QoS message overhead per connection request of TBP-SE is kept at the same level as TBP. These results prove the efficiency of our stability estimation model and our method to detect paths with high stability.

The results reported in this paper can be extended in two ways. First, if other optimization objectives need to be considered, such as low link utilization, more groups of colored tickets can be generated. Each group of tickets has the same color and is directed toward the destination node according to one specific optimization objective. For every optimization objective, an objective variable, like the path stability rate for the optimization objective of high path stability, is accumulated while the probe travels. At the destination node, based on a weighted combination of all objective variables, the primary path is selected among all the feasible paths. Second, the estimation models of relative link stability and relative path stability can be integrated into other QoS routing schemes such as geographic routing and cluster-based routing. Addressing these, and similar issues, promises to be an exciting area for future work.

REFERENCES

[1] J. N. Al-Karaki and A. E. Kamal, “Stability Helps Quality of Service Routing in Wireless Ad Hoc Networks,” *Proc. of 2004 IEEE International Conference on Performance, Computing, and Communications*, 2004, pp. 329–336.

[2] Bettstetter, H. Hartenstein, and X. Perez-Costa, “Stochastic Properties of the Random Waypoint Mobility Model,” *Wireless Networks*, Kluwer Academic Publishers, vol. 10, no. 5, September 2004, pp. 555–567.

[3] L. Chen and W. Heinzelman, “QoS-aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, March 2005, pp. 561–572.

[4] S. Chen and K. Nahrstedt, “Distributed Quality-of Service Routing in Ad Hoc Networks”, *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, August 1999, pp. 1–18.

[5] X. Chen, M. Faloutsos, and S. Krishnamurthy, “Distance Adaptive (DAD) Broadcasting for Ad Hoc Networks,” *Proc. of IEEE Military Communications Conference (MILCOM 2002)*, vol. 21, no. 1, October 2002, pp. 878–882.

[6] S. K. Das, A. Mukherjee, S. Bandyopadhyay, D. Saha, and K. Paul, “An Adaptive Framework for QoS Routing through Multiple Paths in Ad Hoc Wireless Networks,” *Journal of Parallel and Distributed Computing*, vol. 63, no. 2, February 2003, pp. 141–153.

[7] M. Garey and D. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*, New York: W. H. Freeman, 1979.

[8] V. Kawadia and P.R. Kumar, “Power Control and Clustering in Ad Hoc Networks,” *Proc. of IEEE INFOCOM 2003*, vol. 1, March–April 2003, pp. 459–469.

[9] J.-Y. Le Boudec, “On the Stationary Distribution of Speed and Location of Random Waypoint,” *IEEE Trans. on Mobile Computing*, vol. 4, no. 4, July–August 2005, pp. 404–405.

[10] W. Liu, X. Chen, Y. Fang, and J.M. Shea, “Courtesy Piggybacking: Supporting Differentiated Services In Multihop Mobile Ad Hoc Networks,” *IEEE Trans. on Mobile Computing*, vol. 3, no. 4, October–December 2004, pp. 380–393.

[11] W. Navidi and T. Camp, “Stationary Distributions for the Random Waypoint Mobility Model,” *IEEE Trans. on Mobile Computing*, vol. 3, no. 1, January–March 2004, pp. 99–108.

[12] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks*, Prentice Hall, 2002.

[13] M. R. Pearlman, Z. J. Haas, P. E. Sholander, and S. S. Tabrizi, “On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks,” *Proc. of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC 2000)*, 2000, pp. 3–10.

[14] Perkins and P. Bhagwat, “Highly Dynamic Destination-sequenced Distance Vector Routing (DSDV) for Mobile Computers,” *Proc. of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM’94)*, August–September, pp. 234–244.

[15] R. Ramanathan, “On the Performance of Ad Hoc Networks with Beamforming Antennas,” *Proc. of the 2001 ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC 2001)*, October 2001, pp. 95–105.

[16] P. Samar and S. B. Wicker, “Network Communication Capacity and Behaviors: On the Behavior of Communication Links of a Node in a Multi-hop Mobile Environment,” *Proc. of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC 2004)*, May 2004, pp. 145–156.

[17] S. H. Shah and K. Nahrstedt, “Predictive Location-based QoS Routing in Mobile Ad Hoc Networks,” *Proc. of 2002 IEEE International Conference on Communications (ICC 2002)*, vol. 2, April–May 2002, pp. 1022–1027.

[18] Son, A. Helmy, and B. Krishnamachari, “The Effect of Mobility-Induced Location Errors on Geographic Routing in Mobile Ad Hoc and Sensor Networks: Analysis and Improvement Using Mobility Prediction,” *IEEE Trans. on Mobile Computing*, vol. 3, no. 3, July–September 2004, pp. 233–245.

[19] Y. Wang, H. Lin, and S. Chang, “Interfering-aware QoS Multipath Routing for Ad Hoc Wireless Network,” *Proc. of 18th International Conference on Advanced Information Networking and Applications*, 2004, vol. 1, pp. 29–34.

[20] L. Xiao, J. Wang, and K. Nahrstedt, “The Enhanced Ticket-Based Routing Algorithm,” *Proc. of 2002 IEEE International Conference on Communications (ICC 2002)*, vol. 25, no. 1, April 2002, pp. 2222–2226.

[21] Y. Yang and R. Kravets, “Contention-aware Admission Control for Ad Hoc Networks,” *IEEE Trans. on Mobile Computing*, vol. 4, no. 4, July–August 2005, pp. 363–377.

[22] Younis, S. Fahmy, and P. Santi, “Robust communications for sensor networks in hostile,” *Proc. of Twelfth IEEE International Workshop on Quality of Service (IWQOS 2004)*, June 2004, pp. 10–19.