Reliable Routing in Mobile Ad Hoc Networks Based on Mobility Prediction

Jian Tang, Guoliang Xue *Senior Member, IEEE* and Weiyi Zhang
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-8809
Email:{jian.tang, xue, weiyi.zhang}@asu.edu

Abstract— **Reliability is a major issue in mobile ad hoc routing. Shortest paths are usually used to route packets in Mobile Ad hoc NETworks (MANETs). However, a shortest path may fail quickly, because some of the wireless links on the shortest path may be broken shortly after the path is established due to mobility of mobile nodes. Rediscovering routes can result in substantial data loss and communication overheads. In this paper, we consider a MANET in the urban environment. We formulate and study two optimization problems related to reliable routing in MANETs. In the Minimum Cost Duration-Bounded Path (MCDBP) routing problem, we seek a minimum cost source to destination path with duration no less than a given threshold. In the Maximum Duration Cost-Bounded Path (MDCBP) routing problem, we seek a maximum duration source to destination path with cost no greater than a given constraint. We use a waypoint graph to model the working area of a MANET and present an offline algorithm to compute a duration prediction table for the given waypoint graph. An entry in the duration prediction table contains the guaranteed worst-case duration of the corresponding wireless link. We then present an efficient algorithm which computes a minimum cost duration-bounded path, using the information provided in the duration prediction table. We also present a heuristic algorithm for the MDCBP routing problem. Our simulation results show that our mobility prediction based routing algorithms lead to better network throughput and longer average path duration, compared with the shortest path algorithm.**

I. INTRODUCTION

The Mobile Ad hoc NETwork (MANET) is different from traditional wireless networks in many ways. One of the basic differences is that a MANET is a multi-hop wireless network, i.e., a routing path is composed of a number of intermediate mobile nodes and wireless links connecting them. Since nodes can move at any

time, wireless links are prone to be broken. Any link break along an established routing path will lead to a path failure. A shortest path may fail sooner than another path connecting a given source and destination pair. Frequent routing discovery is costly and inefficient. Moreover, shortest path routing can not support many Quality of Service (QoS) connection requests when the path duration is a requirement. For example, a video stream could be required to be transferred from node s to node t without any interruption for 100 seconds in a multimedia application. Instead of shortest paths, more durable paths or paths with duration guarantees are preferred to be used for routing packets.

Originally, the MANET is proposed for military applications in the battlefield. However, future MANETs could be deployed in various environments. the city-wide MANET begins to attract research attentions recently ([1]) because of its application potential. Different from movements in the battlefield, movements in the city are highly restricted by roadways, i.e., the following movement rules must be obeyed: a vehicle or person can only move along roads, turn or stay at intersections. In addition, the driving speed of a vehicle on a specific road segment cannot exceed its prescribed speed limit. A similar mobility pattern is described in the Manhattan mobility model ([1]). Therefore, it is possible for us to make a relatively accurate prediction for mobility of mobile nodes, which will give a good insight for finding reliable routing paths. In this paper, we consider a MANET in the urban environment. As mentioned before, we are interested in QoS connection requests with duration requirements. In addition, we are also interested in finding a path whose duration is as long as possible but whose cost is as low as possible. We formulate two optimization problems for reliable routing in MANETs. They are the Minimum Cost Duration-Bounded Path (MCDBP) routing problem and the Maximum Duration Cost-Bounded Path (MDCBP) routing problem. We introduce the *waypoint graph* to model the

city map and present a prediction algorithm to compute a duration table for the given waypoint graph. Each entry in the table gives the worst-case duration of a corresponding wireless link, i.e., at least how long it can last. Based on the prediction table, we present an algorithm to solve the (MCDBP) problem optimally and a heuristic algorithm for the (MDCBP) problem.

The rest of this paper is organized as follows. We discuss related work in Section II. We formally define our problems and some notations in Section III. We describe our prediction and routing algorithms in Section IV. We present our simulation results in Section V. We conclude the paper in Section VI.

II. RELATED WORK

In [2], the authors introduce the random waypoint model which turns out to be the most widely used mobility model in the literature. In this model, at every instant, each mobile node chooses a random destination and moves toward it with a speed uniformly distributed in [0, V_{max}], where V_{max} is the maximum allowable speed for a node. After reaching the destination, the node stops for a random duration. It then chooses another random destination and repeats the whole process. Besides the random waypoint model, some other mobility models are proposed for special purposes. Reference Point Group Model (RPGM) is proposed in [3] to characterize mobility behaviors in the battlefield. Recently, the freeway model and Manhattan model are introduced by Bai *et al.* in [1]. In these two models, movements of nodes are highly restricted by roadways. The authors also evaluate the performance of various routing protocols for MANET under different mobility models. A more recent paper ([11]) analyzes the statistics of path duration under different mobility models and studies their impact on routing protocols.

Mobile ad hoc routing has been extensively studied in the literature. The well-known on-demand routing protocols, including AODV ([10]), DSR ([7]) and so on, basically will flood route discovery messages upon arrival of a connection request, and will choose a shortest path to route packets from the given source to the destination. A landmark routing protocol is proposed specially for group mobility in [9]. Reliable routing in MANETs has also been well studied before. Toh proposes an associativity-based long-lived routing (ABR) protocol in [14]. Routes selected by the protocol are likely to be long-lived and hence there is no need for frequent route recoveries. In [5], the authors propose link stability comparison models for routing algorithms. They show properties of these models and propose an enhanced link stability estimation model to find a route

with longer lifetime. Multiple path routing algorithms are also proposed to improve reliability in [8], [15] and [16].

Mobility prediction has also been applied to design efficient routing algorithms for MANETs before. [12] is the first paper to apply Global Positioning System (GPS) in QoS routing decisions, and to consider and predict the connection time (estimated lifetime) of wireless links. In [13], the authors propose a simple mechanism to predict durations of wireless links in a MANET by assuming directions and speeds of end nodes of wireless links will not change in the future. The methods for applying this prediction mechanism to existing unicast and multicast routing protocols are also described in [13] and [4]. They use simulations to show the performance enhancement by their mobility prediction scheme. The authors of [6] introduce a prediction-based link availability estimation. They also propose to use their estimation algorithm to develop a metric for path selection in terms of path reliability, which is shown to improve network performance by simulations.

Our work is different from all previous work in the following ways: (1) We propose an offline algorithm to predict link durations in the worst-case for the city-wide MANET. (2) We present an efficient routing algorithm which can find minimum cost paths with required duration guarantees based on our prediction algorithm. (3) We also present a heuristic algorithm which can find relatively durable paths, compared to shortest paths. We also study the tradeoff between path cost and path duration through simulations.

III. PROBLEM STATEMENTS

As mentioned before, we study a MANET in the urban environment. We model the working area of the network using a waypoint graph $WG(V, E)$. Every vertex in WG is a waypoint which has a specific location in the Euclidean plane and corresponds to an intersection of two or more roads. For any pair of waypoints, w_1, w_2, if there exists a road segment directly connecting them, we will add two directed edges, w_1w_2 and w_2w_1 into the graph and their costs are the Euclidean distance between the two end waypoints. We use two directed edges to distinguish two different moving directions. We study a MANET $G(N, L)$ with mobile node set N and wireless link set L. We assume that every mobile node is aware of its location which can be obtained from GPS or some other location service systems. We also assume that all mobile nodes have the same fixed communication range R. There is an undirected link i connecting node u and v in G if and only if the Euclidean distance between u and v is no more than R. There is a cost function, $C(i)$,

which assigns a cost value for each link i in G. This cost value could be the transmission cost, the delay of the link, etc, or a combination of these parameters. Similarly, the *duration* of a wireless link i with end nodes u and v (denoted by $D(i)$) is the period during which node u and node v are within the communication range of each other. A wireless link will be broken if the Euclidean distance between its two end nodes becomes greater than R. If one of its end node is currently out of communication range of another or the link is broken when 1 time unit elapses, then its duration is 0.

Definition 1: Let $i_1, i_2, \ldots, i_p$ be the links of the path P. Then the **Duration of a Path P** is $D(P) = \min_{1 \le j \le p} D(i_j)$, where $D(i_j)$ is the duration of link i_j.

Similarly, we will have the definition for the path cost.

Definition 2: The **Cost of a Path P** is $C(P) = \sum_{j=1}^{p} C(i_j)$, where $C(i_j)$ is the cost of link i_j.

Now we are ready to formulate two optimization problems for reliable routing in MANETs.

Definition 3: Given a source node s and a destination node t, together with a duration threshold $DT > 0$, a **Duration-Bounded Path** is a path from s to t such that $D(P) \ge DT$.

Definition 4: Given a source node s and a destination node t, together with a duration threshold $DT > 0$, the **Minimum Cost Duration-Bounded Path (MCDBP)** routing problem seeks a path P from s to t with minimum cost among all **Duration-Bounded Paths**.

Definition 5: Given a source node s and a destination node t, together with a cost threshold $CT > 0$, a **Cost-Bounded Path** from s to t such that $C(P) \le CT$.

Definition 6: Given a source node s and a destination node t, together with a cost threshold $CT > 0$, the **Maximum Duration Cost-Bounded Path (MDCBP)** routing problem seeks a path P from s to t with maximum duration among all **Cost-Bounded Paths**.

IV. Reliable Ad Hoc Routing

In this section, we will present a complete routing scheme to support reliable routing, which includes an offline prediction algorithm and two routing algorithms. After running our prediction algorithm, we will have a link duration prediction table. By looking up this table, we can find a worst-case duration value for each possible link. Then our routing algorithms can be employed to find reliable paths to route packets. Before describing our routing algorithms, we talk about our prediction algorithm first.

A. The Prediction Algorithm

In order to predict the duration, we need to create an augment graph $AWG(V_A, E_A)$ based on the way-point graph WG by adding some new *Landmarks* into every road segment of WG. The distance between every two consecutive landmarks is the same and is called a distance unit. The vertex set V_A of AGW corresponds to the union of waypoints in WG and newly added landmarks, and the edge set E_A corresponds to the union of those separated road segments. However, in order to decide how many landmarks need to be added for a road segment, we must introduce the concept of *Role*. A mobile node can be a walking person, a running person, a vehicle or anything you want to define, which is called the *Role* of a mobile node. Based on the role, we can decide the speed of that node on a specific road segment. According to practical experiences, a vehicle normally moves as fast as the speed limit, so we can obtain its speed on a specific road segment by simply looking up the corresponding speed limit table of the given waypoint graph. However, people runs/walks at roughly the same constant speed on different road segments. No matter which role of a node is, we need to guarantee that the number of landmarks on a road segment must be a multiple of the number of landmarks it passes within one time unit. Therefore, once the waypoint graph, all types of roles and their speeds on different road segments are known, we can compute the minimum number of landmarks needed to be added into every road segment. We also have to assume that initially every mobile node will be at some vertices of AGW. Although this may not be exactly true in practice, it is a fairly close approximation. Actually the prediction precision can be improved by adding more landmarks. However, this will increase the time complexity of computation.

We label every waypoint, road segment in WG and landmark in AGW. In the following, the road segment always means the whole road segment between two waypoints, not landmarks. The LandmarkID can uniquely identify a vertex on graph AGW since if the vertex is a waypoint, we assign a negative value to LandmarkID whose absolute value is that WaypointID. We may also note that the mobile node will only move in two directions if it is on a road segment and can stay or go to any outgoing road segment if it is on a waypoint. Since the waypoint graph is a bidirectional graph, the SegmentID can represent moving directions of a mobile node.

Now we are ready to introduce the concept of a *Possible Link*. For every pair of vertices in AGW whose Euclidean distance is no more than the communication range R, we will have one possible link whose two ends correspond to those two vertices. A triple (RoleID, LandmarkID, SegmentID) will be sufficient to uniquely identify any possible mobile node on AGW, which are called

the *Mobility Parameters* of a mobile node. So every possible link for the given AGW can be represented by a 6-tuple, ($RoleID_u$, $LandmarkID_u$, $SegmentID_u$, $RoleID_v$, $LandmarkID_v$, $SegmentID_v$). In this way, we can identify a finite number of possible links for a given waypoint graph WG (AWG is constructed based on WG) and we call the set of possible links on WG, $PL(WG)$. No matter how a MANET $G(N, L)$ is deployed on the waypoint graph WG, for each node in G, we will have a vertex in AGW corresponding to it and for each wireless link in G, we will have a possible link from $PL(WG)$ corresponding to it. For example, suppose $RoleID = 1$ represents the vehicle and $RoleID = 2$ represents the walking person, then $(1, -1, 1, 2, 35, 14)$ represents a possible link corresponding to a wireless link of G whose one end node is a vehicle at the waypoint 1 moving along road segment 1 and another end node is a walking person at the landmark 35 going along the road segment 14.

The duration prediction table will be indexed by a 6-tuple ($RoleID_u$, $LandmarkID_u$, $SegmentID_u$, $RoleID_v$, $LandmarkID_v$, $SegmentID_v$). Every entry of the table corresponds to a possible link in $PL(WG)$ and indicates at least how long this possible link can last. Since it is hard to directly compute the duration prediction table. An auxiliary table, AD_Table, is used to assist the computation. The AD_Table is indexed by a 7-tuple ($RoleID_u$, $LandmarkID_u$, $SegmentID_u$, $RoleID_v$, $LandmarkID_v$, $SegmentID_v$, $duration$). Every entry of the table corresponds to a possible link in $PL(WG)$ and indicates if this possible link can last *duration* time units in the worst case by YES or NO. We propose Algorithm 1 to compute the AD_Table. Once we obtain the AD_Table, we can have the duration prediction table by a simple transformation.

In Algorithm 1, we use N_ROLE, N_LM, N_SEG, MAX_D to denote the number of roles, the number of vertices in AWG, the number of road segments and the max possible duration respectively. In addition, $(R_u, L_u, S_u, R_v, L_v, S_v, d)$ is the simpler representation for ($RoleID_u$, $LandmarkID_u$, $SegmentID_u$, $RoleID_v$, $LandmarkID_v$, $SegmentID_v$, $duration$). R_u^+ L_u^+, S_u^+, R_v^+, L_v^+ and S_v^+ denote one of next possible roles, locations and directions after one step movement from R_u, L_u, S_u, R_v, L_v and S_v.

Basically Algorithm 1 is a dynamic programming algorithm. In **step_1**, we initialize the AD_Table. In **step_2**, we compute all possible R_u^+ L_u^+, S_u^+, R_v^+, L_v^+ and S_v^+ according to the movement rules, i.e., the node can move in two directions if it is on a road segment and can stay or move to any outgoing road segment if it is on a waypoint. Of course, the role of a node will never change during

---

Algorithm 1 AD_Table Computation Algorithm

INPUT: $AGW(V_A, E_A)$, N_ROLE, N_LM, N_SEG, MAX_D
OUTPUT: AD_Table.

step_1 **forall** R_u,L_u,S_u,R_v,L_v,S_v,d **do**
    if ($d == 0$) **then**
      AD_Table(R_u,L_u,S_u,R_v,L_v,S_v,d)=YES;
    else
      AD_Table(R_u,L_u,S_u,R_v,L_v,S_v,d)=NO;
    endif
    endfor
step_2 **for** $d = 1$ **to** MAX_D **do**
    forall R_u,L_u,S_u,R_v,L_v,S_v, s. t.
    AD_Table(R_u,L_u,S_u,R_v,L_v,S_v,d-1)==YES
    do
    AD_Table(R_u,L_u,S_u,R_v,L_v,S_v,d)=YES;
    forall R_u^+,L_u^+,S_u^+,R_v^+,L_v^+,S_v^+ **do**
      if AD_Table(R_u^+,L_u^+,S_u^+,
                R_v^+,L_v^+,S_v^+,d-1)==NO
      then
        AD_Table(R_u,L_u,S_u,R_v,L_v,S_v,d)=NO;
        break;
      endif
    endfor
    endfor
    endfor

---

the whole procedure. We iteratively increase duration d and we test all possible R_u^+, L_u^+, S_u^+, R_v^+, L_v^+ and S_v^+ to see if the link can survive the next step movement. Since we are making worst-case prediction, the link is considered not able to survive if it will break in one of possible cases. If we have a YES, we will know that this possible link can at least last d time units. When we have the AD_Table, we can construct the duration prediction table, DP_Table, as follows. For each entry in DP_Table, DP_Table(R_u,L_u,S_u,R_v,L_v,S_v) is equal to the maximum integer d such that AD_Table(R_u,L_u,S_u,R_v,L_v,S_v,d) = YES. If the values of all related entries in AD_Table is NO, then DP_Table(R_u,L_u,S_u,R_v,L_v,S_v) = 0.

Let us use N_Degree to denote the maximum outgoing degrees of waypoints. Based on our assumptions, we have at most $N\_ROLE^2 \times N\_LM^2 \times N\_Degree^2$ possible links. Computing the AD_Table will take $O(N\_ROLE^2 \times N\_LM^2 \times N\_Degree^2 \times MAX\_D)$ and constructing DP_Table from AD_Table will take linear time. We can decrease the total number of entries in the DP_Table by eliminating symmetric ones. Even with this, the size of the table could still be very large for some large waypoint graphs and the computation time

could be very long. However, the duration prediction table can be computed offline once a waypoint graph is given. Since the waypoint graph does not change often, the table needs not to be recomputed frequently. Therefore the waypoint graph and the corresponding duration prediction table can be stored in every mobile node and used repeatedly. So even the table computation is a costly procedure, it will not influence the efficiency of our routing scheme.

B. The Routing Algorithms

Now we are ready to present our routing algorithms. Firstly, we present an algorithm which is able to optimally solve Minimum Cost Duration-Bounded Path routing (MCDBP) problem defined in Section III.

Algorithm 2 MCDBP Routing Algorithm

INPUT: MANET $G(N, L)$, the mobility parameters (R_u, L_u, S_u) of all n nodes $v_1, v_2, \ldots, v_n$, their communication range R and the cost function C. A connection request ρ with source $s(\rho)$, destination $t(\rho)$, along with a duration threshold $DT > 0$.

OUTPUT: Either block the request or establish a s–t paths with minimum total cost among all those whose duration is at least DT.

---

step_1 Construct a graph G_B in the following way. The set of vertices N_B of G_B contains all n mobile nodes $v_1, v_2, \ldots, v_n$ in G. The set of undirected edges L_B of G_B contains all pairs $(u, v) \in N_B \times N_B$, such that $D(uv) \geq DT$, where $D(uv) = DP\_Table(R_u, L_u, S_u, R_v, L_v, S_v)$. The cost of the link uv is assigned to $C(uv)$.

step_2 Run Dijkstra's algorithm on graph G_B to find an s–t path with minimum total cost.

step_3 **if** such a path cannot be found in **step**_2
          Block the connection request ρ.
       else
          Return the found path.

We will have the following theorem.

Theorem 1: The worst case running time of Algorithm 2 is $O(n^2)$. Whenever an s–t path with duration at least DT exists, Algorithm 2 finds such a path with minimum total cost.

PROOF. In the worst-case, the number of links m in a MANET is $O(n^2)$. Looking up the duration prediction table for a specific link uv will take constant time since it is indexed by the end nodes of possible links. So **step**_1 takes $O(n^2)$ time. In **step**_2, the Dijkstra algorithm will

take $O(m + n \log n)$, i.e., $O(n^2)$. Therefore, the time complexity of Algorithm 2 is $O(n^2)$. The correctness of the algorithm lies in the fact that our prediction algorithm gives an guaranteed worst-case duration for each link. The graph G_B constructed in **step**_1 only includes those links whose duration is greater than or equal to the given duration threshold DT. So the Dijkstra algorithm in **step**_2 guarantees to find a path with minimum cost and with duration at least DT if it exists.   □

Knowing mobility parameters of each mobile node is the prerequisite for successfully running Algorithm 2. Since each mobile node is aware of its location and the waypoint graph, it can figure out all necessary mobility parameters about itself. In order to gather mobility parameters from all other nodes to the source node, an inquiry procedure can be initiated by the source node s when a connection request arrives. Basically, inquiry messages will be flooded throughout the whole network. Any node receiving the inquiry message will then pack its mobility parameters into a rely message and send it back along the same path the received inquiry message traverses but in the opposite direction. The communication overhead can be decreased by in-network aggregation, i.e, some intermediate mobile nodes can aggregate reply messages from all its downstream mobile nodes and only send back a single reply message to the source node.

Algorithm 3 is a heuristic algorithm for the MDCBP routing problem. it uses Algorithm 2 as a subroutine for $O(\log n)$ times while bisecting the set of possible duration values. So the total running time of Algorithm 3 is $O(n^2 \log n)$. If our prediction algorithm gives the *actual* duration of each link, rather than the worst-case duration, then this algorithm will give an optimal solution for MDCBP problem. However, our prediction is a worst-case prediction. Therefore, the actual duration of a link may be much longer than the predicted value. We cannot guarantee that the found path by our algorithm has a longer duration than other candidate $s - t$ paths. That is the reason why we claim it to be a heuristic algorithm. However, our prediction scheme provides an estimation for the reliability of a wireless link. Hopefully, wireless links with longer worst-case durations will last longer. In the next section, we use simulations to show that paths found by Algorithm 3 are actually reliable in most of cases.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our algorithms via simulations. We randomly generate grid-like waypoint graphs. Figure 1 shows a sample waypoint graph. In these preliminary simulations, all waypoint

Algorithm 3 MDCBP Routing Algorithm

INPUT: MANET $G(N, L)$, mobility parameters (R_u, L_u, S_u) of all n nodes $v_1, v_2, \ldots, v_n$, their communication range R and the cost function C. A connection request ρ with source $s(\rho)$, destination $t(\rho)$, along with a cost threshold $CT > 0$.

OUTPUT: Either block the request or establish a s–t path with maximum duration among all those whose cost is at most CT.

step_1 Compute the set of distinct values $0 \leq D_1 < D_2 < \cdots < D_k$ such that for every pair of nodes (u, v) whose Euclidean distance is not greater than R, there is some index i ($1 \leq i \leq k$) such that $D(uv) = D_i$, where $D(uv) = DP\_Table(R_u, L_u, S_u, R_v, L_v, S_v)$. Let $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$.

step_2 Use bisection on $\mathcal{D}$ to find the largest D_i such that the solution to corresponding **MCDBP** routing problem computed by Algorithm 2 has cost no more than CT.

step_3 **if** such a value cannot be found in **step_2**

       Block the connection request ρ.

  else

       Output the corresponding solution.

Fig. 1. A Sample Waypoint Graph

graphs used have 5 blocks in the vertical direction and 5 blocks in the horizontal direction. The distance between two blocks is a random value ranging from about 90 meters to 270 meters. The role, initial locations and moving directions of mobile nodes are randomly generated. Each node randomly chooses a waypoint as its destination, moves along the shortest path on the waypoint graph to the destination. After it reaches the destination, it stays there for a while, which is also determined by a random value uniformly distributed from 18 to 30 time units. In all simulations, one time unit is equal to 10 seconds. Then it randomly chooses another destination and repeats the above procedure. The number of roles is 2. One type of node represents the walking person moving at a speed of 2.235 m/s. Another represents the vehicle. We assign the vehicle speed limits of all road segments to be 4 times moving speed of the walking person. Every mobile node has the same fixed communication range, 250 meters. In all simulations, we employ the *hop count* as the metric to measure path cost. It is a commonly used metric since it represents the number of transmissions or the delay of a routing path.
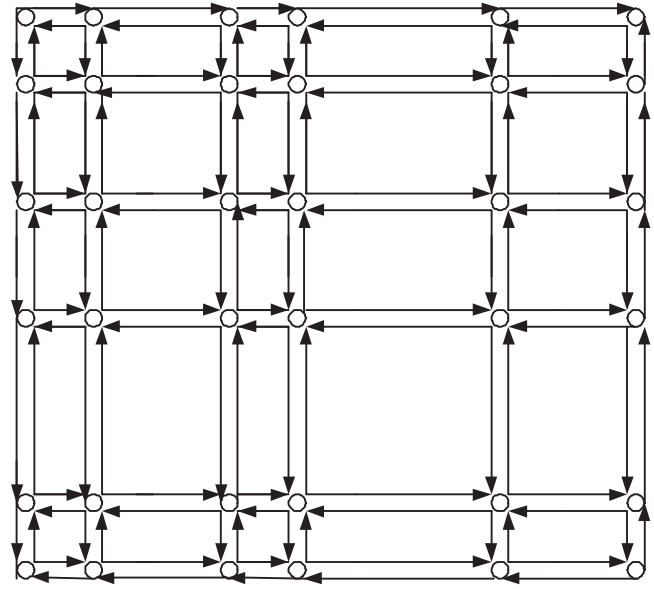
In first two experiments, we compare the performance of our Algorithm 2 with the Shortest Path (SP) algorithm in terms of network throughput. Every 30 time units, 10 connection requests are injected into the network, whose sources and destinations are randomly chosen. Totally, 1000 connection requests will be generated in each run. The duration threshold is randomly picked as 1 or 2 time units. When a connection request arrives, each algorithm will be invoked to compute a single path for routing. If the algorithm fails to find a path or the worst-case duration of the found path cannot satisfy the given duration threshold, the request will be rejected. We count the total number of successfully established connections and use it to represent network throughput. In the first scenario, we create a mobile network with 60 nodes and run the simulation on 10 different randomly generated waypoint graph. In the second scenario, we run the simulation on one waypoint graph, but randomly generate 5 different mobile networks with 40, 60, 80, 100, 120 nodes respectively. Results are shown in the following two tables.

Table I and II show the percentage of the number of successfully established connections against the total number of connection requests (1000) by our algorithms and by the shortest path algorithms. Intuitively, network throughput given by both algorithms will become higher on relatively dense waypoint graphs (distances between blocks are relatively small) since wireless links in dense waypoint graphs are not easy to break. Our simulation results testify it since we make the waypoint graph denser and denser from trial 1 to 10 by controlling generation

TABLE I
Network Throughput on Different Waypoint Graphs

WG	SP(%)	MDCBP(%)	Increase(%)
1	16.9	35.6	110.7
2	17.1	39.5	123.2
3	24.1	53.4	121.6
4	22.8	54.1	137.3
5	27.9	65.8	135.8
6	35.4	75.7	113.8
7	32.5	79.2	143.7
8	44.7	92.4	106.7
9	49.6	99.3	100.2
10	48.7	99.3	103.9
Avg	32.0	69.4	116.8

TABLE II
Throughput of Networks with Different Sizes

Size	SP(%)	MDCBP(%)	Increase(%)
40	19.0	31.2	64.2
60	24.1	53.4	121.6
80	27.6	72.0	160.9
100	26.1	82.8	217.2
120	29.2	89.3	205.8
Avg	25.2	65.7	160.9

total number of connection requests (1000). The cost threshold in our Algorithm 3 is set to be $bound\_ratio * MH$, where MH is the minimum hop count for the given source and destination pair in the network. We do simulations firstly on different waypoint graphs and then on mobile networks with different sizes. Simulation results are presented in the following 6 tables, in which each entry is the average over 1000 connection instances.

TABLE III
Duration Increase on Different Waypoint Graphs

WG	∞(%)	1.0(%)	1.2(%)	1.5(%)	2.0(%)
1	6.7	4.1	6.4	11.4	8.2
2	5.9	10.3	8.7	10.4	8.8
3	-8.4	4.1	4.3	2.0	-5.1
4	17.0	10.8	12.1	19.9	15.4
5	5.2	6.3	7.9	8.5	6.9
6	3.4	3.7	6.2	8.1	3.8
7	-2.6	11.5	11.4	9.9	2.4
8	-12.6	7.7	7.9	0.9	-9.9
9	6.6	3.2	4.2	8.1	6.3
10	-1.3	8.6	9.4	7.0	2.4
Avg	2.0	7.0	7.9	8.6	3.9

TABLE IV
Hop Count Increase on Different Waypoint Graphs

WG	∞(%)	1.0(%)	1.2(%)	1.5(%)	2.0(%)
1	22.5	0.0	3.5	13.1	19.8
2	37.5	0.0	2.7	18.3	30.2
3	34.3	0.0	3.2	15.8	26.4
4	23.9	0.0	4.2	12.0	18.4
5	31.5	0.0	3.8	15.4	24.5
6	21.8	0.0	2.9	11.0	17.8
7	45.1	0.0	0.9	19.0	36.1
8	55.1	0.0	0.3	22.6	41.5
9	18.0	0.0	3.3	10.9	15.2
10	49.5	0.0	0.7	21.9	40.2
Avg	33.9	0.0	2.6	16.0	27.0

parameters. We can see that network throughput become higher and higher no matter which algorithm is used.

From Table II, we can see that with the shortest path algorithm, the increase of network size does not change the throughput too much because link durations are totally ignored when computing paths. Even in large size mobile networks, paths found by it may still include links which will break soon in the future. However, our algorithm considers and predicts link durations. In large size mobile networks, it will be able to get more chances to have durable links when computing paths, i.e., gain more chances to satisfy given duration thresholds and improve throughput. We find out that with regards to network throughput, our algorithm outperforms the shortest path algorithm more than 100% on average.

In the other two experiments, our Algorithm 3 is compared with the shortest path algorithm in terms of path durations and hop counts. Similar to last experiments, we totally inject 1000 connection requests with random sources and destinations, 10 requests each time. The durations and hop counts of paths are counted in simulations. Besides these two metrics, we also introduce another metric called the *failure ratio* which is the ratio between the number of times durations of paths computed by our algorithms are actually less than those of paths founded by the shortest path algorithm and the

Those tables shows the average path duration increase, path hop count increase and failure ratios given by our Algorithm 3 (cost bound_ratio is set to $\infty, 1.0, 1.2, 1.5$ and 2.0 respectively) compared to the shortest path algorithm. If the $bound\_ratio = \infty$, it basically means that we do not put any constraint on hop count when computing the path. So our algorithm will find a path with hopefully longest duration. An interesting observation is that paths found by our algorithm with $bound\_ratio = \infty$ do not perform very well. In Table III, on 4 different waypoint graph instances, the average duration of paths given by our algorithm by setting $bound\_ratio = \infty$ are even shorter than those of paths computed by the

TABLE V
Failure Ratios on Different Waypoint Graphs

WG	∞(%)	1.0(%)	1.2(%)	1.5(%)	2.0(%)
1	10.4	5.7	6.1	7.9	9.4
2	23.3	11.6	12.4	13.8	20.2
3	22.4	8.0	9.2	14.3	21.6
4	10.8	5.7	6.3	7.9	10.5
5	21.0	10.9	12.3	16.9	19.5
6	6.7	2.6	2.9	5.3	6.5
7	38.7	15.8	17.3	23.6	35.5
8	45.6	26.8	24.0	35.9	43.7
9	10.2	5.3	5.4	7.3	10.0
10	34.3	20.3	20.7	26.8	34.4
Avg	22.3	11.3	11.7	16.0	21.1

TABLE VI
Duration Increase on Networks with Different Sizes

Size	∞(%)	1.0(%)	1.2(%)	1.5(%)	2.0(%)
40	2.2	0.5	2.4	4.5	1.1
60	6.7	4.1	6.4	11.4	8.2
80	14.0	8.5	14.0	21.6	18.3
100	18.4	10.5	15.3	21.0	19.5
120	23.8	14.7	16.1	28.2	28.2
Avg	13.0	7.7	10.8	17.3	15.1

TABLE VII
Hop Count Increase on Networks with Different Sizes

Size	∞(%)	1.0(%)	1.2(%)	1.5(%)	2.0(%)
40	15.8	0.0	2.1	8.6	14.6
60	22.5	0.0	3.5	13.1	19.8
80	35.1	0.0	4.9	16.5	27.9
100	39.2	0.0	4.7	20.7	32.0
120	47.7	0.0	5.5	23.1	36.4
Avg	32.1	0.0	4.1	16.4	26.1

TABLE VIII
Failure Ratios on Networks with Different Sizes

Size	∞(%)	1.0(%)	1.2(%)	1.5(%)	2.0(%)
40	6.4	2.3	2.9	3.6	6.5
60	10.4	5.7	6.1	7.9	9.4
80	15.0	8.0	7.2	8.3	12.5
100	16.7	11.4	12.4	13.8	17.3
120	16.3	8.3	10.9	10.0	15.0
Avg	13.0	7.1	7.9	8.7	12.1

the shortest path algorithm. Furthermore, costs of those paths are 33.9% more than that of the shortest paths on average. We can also see that the average failure ratio is as high as 22.3% from table V. That is to say, when setting $bound\_ratio = \infty$, our algorithm fails to come up with a solution as good as the one given by the shortest path algorithm with probability about 22.3%. One reason

for this is due to the fact that our prediction is a worst-case prediction, i.e, the actual duration of a link will probably last much longer than the predicted value in the simulation. Therefore, paths found by our algorithm are not guaranteed to last longer than those found by the shortest path algorithm. On the other hand, By setting $bound\_ratio = \infty$, the found path will include more number of hops than the shortest path. Generally, the more hop count a path has, the more likely it will break soon since any link breakup will fail the whole path. So if we restrict the hop count of the path somehow, we definitely can decrease its hop count and hopefully we can prolong its duration. If this bound ratio is set to 2.0, we obtain similar results as the one with $bound\_ratio = \infty$. But if it is set to 1.0, the failure ratio is reduced to roughly 10% or even less in some network instances and the duration increase is improved to be about 7% on average without increasing the hop count at all. Firstly, we may note that we definitely will not increase the path hop count by setting $bound\_ratio = 1.0$. The reason for duration improvement is that in mobile networks, especially in relatively dense networks, there will exist several paths with the same minimum number of hops for a given source and destination pair. Our algorithm can choose one of them with hopefully long duration based on our prediction. If this bound ratio is 1.2, our algorithm can also find paths with longer average duration but minor hop count increase, less than 5% from Table IV and VII. In addition, with probability about 90% , paths founded by our algorithm with proper bound ratio setting will last at least as long as those by the shortest path algorithm without or with minor increase of hop count, and the average duration improvement is around 8%.

VI. CONCLUSIONS

In this paper, we have presented an offline algorithm to predict the worst-case duration of possible wireless links. Based on our prediction algorithm, we then present an efficient algorithm for computing a path connecting a source node and a destination node, which has minimum total cost and duration no smaller than a given threshold. We also present a heuristic algorithm to find a path with maximum actual duration and cost no more than a given threshold. Simulation results show that our first routing algorithm improves the network throughput by more than 100% and that our heuristic algorithm can improve the average path duration by about 8% without or with minor cost increase.

In the future, we intend to design efficient broadcasting and multicasting algorithms for city-wide MANETs based on our mobility prediction scheme. We are also

going to develop prediction-based multipath routing algorithms to improve reliability further and also support fault-tolerance. In addition, We will investigate new and more precise prediction schemes to predict the actual wireless link duration.

REFERENCES

[1] F. Bai, N. Sadagopan, and A. Helmy, IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks, *Proceedings of IEEE Infocom'2003*, March 2003, pp. 825–835.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, *Proceedings of ACM Mobicom'1998*, October 1998, pp. 85–97.

[3] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, A group mobility model for ad hoc wireless networks, *Proceedings of ACM/IEEE MSWiM*, August 1999, pp. 53–60.

[4] S-J Lee, W. Su, M. Gerlaw,Wireless ad hoc multicast routing with mobility prediction, *Mobile Networks and Applications*, Vol. 6(2001), pp. 351–360.

[5] G. Lim, K. Shin, S. Lee, H. Yoon, J. S. Ma, Link stability and route lifetime in ad hoc wireless networks, *Proceedings of IEEE ICPPW'2002*, August 2002, pp. 116–123.

[6] S. Jiang, D. He, J. Rao, A prediction-based link availability estimation for mobile ad hoc networks, *Proceedings of Infocom'2001*, March 2001, pp. 1745–1752.

[7] D. B. Johnson, D. A. Maltz, Y-C Hu, The dynamic source routing protocol for mobile ad hoc networks (DSR), *IETF Draft*, April, 2003.

[8] P. Papadimitratos, Z. J. Haas, E. G. Sirer, Path set selection in mobile ad hoc networks, *Proceedings of IEEE MobiHoc'2002*, June 2002, pp. 1–11.

[9] G. Pei, M. Gerla, X. Hong, LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility, *Proceedings of ACM MobiHoc'2000*, June 2000, pp. 11–18.

[10] C. E. Perkins, E. M. Royer, Ad-hoc on-demand distance vector routing, *Proceedings of IEEE WMCSA'99*, February 1999, pp. 90–100.

[11] N. Sadagopan, F. Bai, B. Krishnamachari, A. Helmy, PATHS: analysis of PATH duration statistics and their impact on reactive MANET routing protocols,*Proceedings of ACM Mobicom'2003*, September 2003, pp. 245–256.

[12] I. Stojmenovic, M. Russell, and B. Vukojevic, Depth first search and location based localized routing and QoS routing in wireless networks, *Proceedings of IEEE International Conference on Parallel Processing*, August 2000, pp. 173-180.

[13] W. Su, S-J Lee, M. Gerla, Mobility prediction and routing in ad hoc wireless networks, *Iternational Journal of Network Management*, Vol. 11(2001), pp. 3–30.

[14] C. K. Toh, Long-lived ad hoc routing based on the concept of Associativity *IETF Draft*, March 1999.

[15] L. Wang, L. Zhang, O. Yang, Y. Shu and M. Dong, Multipath source routing in wireless ad hoc networks, *Proceedings of IEEE CCECE'2000*, May 2000, pp. 479–483.

[16] K. Wu, J. Harms, Multipath routing for mobile ad hoc networks,*IEEE ComSoc/KICS Journal of Communications and Networks, Special Issue on Innovations in Ad Hoc Mobile Pervasive Networks*, Vol. 4(2002), pp. 48–58.