J Comb Optim (2006) 11: 71–85
DOI 10.1007/s10878-006-5978-7

Reliable ad hoc routing based on mobility prediction*

Jian Tang · Guoliang Xue · Weiyi Zhang

Accepted: 25 October 2005

Abstract Reliability is a very important issue in Mobile Ad hoc NETworks (MANETs). Shortest paths are usually used to route packets in MANETs. However, a shortest path may fail quickly, because some of the wireless links along a shortest path may be broken shortly after the path is established due to mobility of mobile nodes. Rediscovering routes may result in substantial data loss and message exchange overhead. In this paper, we study reliable ad hoc routing in the urban environment. Specifically, we formulate and study two optimization problems. In the minimum Cost Duration-bounded Path (CDP) 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 (DCP) routing problem, we seek a maximum duration source to destination path with cost no greater than a given threshold. 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 DCP routing problem. In addition, we show that the proposed prediction and routing schemes can be easily applied for designing reliable ad hoc routing protocols. Simulation results show that our mobility prediction based routing algorithms lead to higher network throughput and longer average path duration, compared with the shortest path routing.

Keywords Reliable ad hoc routing · Mobility prediction · Mobile ad hoc network · QoS

*This research was supported in part by ARO grant W911NF-04-1-0385 and NSF grant CCF-0431167. The information reported here does not reflect the position or the policy of the federal government.

J. Tang (✉)· G. Xue · W. Zhang
Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287-8809
e-mail: {jian.tang, xue, weiyi.zhang}@asu.edu

1. 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 breakage 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 cannot support many Quality of Service (QoS) connection requests when path duration is a requirement. For example, a video stream may need to be transferred from a source node to a destination node 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 packet routing in such applications.

Originally, the MANET was proposed for military applications in the battlefield. However, future MANETs could be deployed in various environments. City-wide MANETs (Bai et al., 2003) have attracted research attentions recently because of its potential applications. Different from movements in the battlefield, movements in a 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 (Bai et al., 2003). Therefore, it is possible for us to make a relatively accurate prediction for mobility of mobile nodes, which will provide a good insight for finding reliable routing paths. In this paper, we study reliable ad hoc routing 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 not too high. We define two optimization problems for reliable routing in city-wide MANETs. They are the minimum Cost Duration-bounded Path (CDP) routing problem and the maximum Duration Cost-bounded Path (DCP) routing problem. We introduce the *waypoint graph* to model the city street 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 the link can last. Utilizing the prediction table, we present an algorithm to solve the CDP problem optimally and a heuristic algorithm for the DCP problem. We also discuss how to apply the duration prediction table and routing algorithms to design reliable ad hoc routing protocols.

The rest of this paper is organized as follows. We discuss related work in Section 2. We formally define the problems to be studied and necessary notations in Section 3. We describe our prediction and routing algorithms in Section 4. We present our simulation results in Section 5 and conclude the paper in Section 6.

2. Related work

Various mobility models have been proposed for MANETs in the literature. In Broch et al. (1998), the authors introduced the random waypoint model which turns out to be the most widely used mobility model. In this model, 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 destination randomly and repeats the whole process. Besides the random waypoint model, several other mobility models are proposed for special purposes. Reference Point Group Model (RPGM) was proposed in Hong et al. (1999) to characterize mobility behaviors in the battlefield. Recently, the freeway model and Manhattan model were introduced by Bai et al. (2003). In these two models, movements of nodes are highly restricted by roadways. The authors also evaluated the performance of various ad hoc routing protocols under different mobility models. A more recent paper (Sadagopan et al., 2003) analyzed the statistics of path duration under different mobility models and studied their impact on routing protocols.

Mobile ad hoc routing has been extensively studied before. The basic ad hoc routing protocols can be divided into three categories: *proactive*, *reactive* (on-demand) and *hybrid*. Proactive routing protocols are based on periodic exchanges of control messages, some of which are sent throughout the whole network when network topology changes. In this way, every node is able to maintain a consistent view of network topology at any time. DSDV (Perkins, 2001) and OLSR (Clausen et al., 2001) fall into this category. Reactive (on demand) routing protocols will flood route discovery messages upon arrival of a connection request, and will choose a shortest path to route packets from the given source node to the destination node. The well-known on-demand routing protocols include AODV (Perkins and Royer, 1999) and DSR (Johnson et al., 2003). Proactive routing protocols can immediately provide the routes when a connection request arrives at the cost of high overheads due to frequent control message exchanges. On the contrary, on-demand routing protocols reduce the overheads at the cost of increased latency of route discovery. Hybrid ad hoc routing protocols such as ZRP (Perkins, 2001) are the middle ground in this tradeoff. Many research efforts have been made to enhance reliability of ad hoc routing. Toh proposed an associativity-based long-lived routing (ABR) protocol in Toh (1999). In Lim et al. (2002), the authors proposed link stability comparison models for routing protocols. They showed the properties of these models and proposed an enhanced link stability estimation model which can be used to find routes with longer lifetimes. In Lou et al. (2004), Papadimitratos et al. (2002), Wang et al. (2000), and Wu and Harms (2002), multiple path routing algorithms are proposed to improve reliability.

Mobility prediction has also been applied to design efficient routing protocols for MANETs before. In Stojmenovic et al. (2000), the authors proposed to apply Global Positioning System (GPS) in QoS routing decisions, and predict the connection time (estimated lifetime) of wireless links. In Su et al. (2001), a simple mechanism was proposed to predict durations of wireless links in a MANET by assuming directions and speeds of end nodes of wireless links would not change in the future. The methods for applying this prediction mechanism to existing unicast and multicast routing protocols are also discussed in Lee et al. (2001) and Su et al. (2001). Simulations were used to show the performance enhancement by their mobility prediction scheme. Jiang et al. (2001) introduced a prediction-based link availability estimation. They also proposed to use their estimation algorithm to develop a metric for path selection in terms of path reliability, which was 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 the worst-case link durations for MANETs in the urban environment. (2) Based on our prediction, we present an efficient routing algorithm which can find minimum cost paths with required duration guarantees. We also present a heuristic which can find more durable paths compared to shortest paths.

3. Problem definition

We model the working area of the network using a waypoint graph $G_W(V_W, E_W)$. Every vertex in G_W 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_1, w_2) and (w_2, w_1), into the graph and with cost set to the Euclidean distance between the two end waypoints. Here we use two directed edges to distinguish two different moving directions. We study a MANET $G(V, E)$ with mobile node set V and wireless link set E. 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 transmission range $R > 0$. There is an undirected link e 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 an edge weighting function, $C(e)$, which assigns a cost value for each link e 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 e with end nodes u and v (denoted by $D(e)$) is the time period during which node u and node v are within the transmission range of each other. A wireless link will be broken if the Euclidean distance between its two end nodes becomes greater than R.

Definition 1. Let $e_1, e_2, \ldots, e_p$ be the links of a path P. Then the **duration of path** P is $D(P) = \min_{1 \le j \le p} D(e_j)$, where $D(e_j)$ is the duration of link e_j. The **cost of path** P is $C(P) = \sum_{j=1}^{p} C(e_j)$, where $C(e_j)$ is the cost of link e_j.

Now we are ready to formulate the optimization problems we are going to study for reliable routing in MANETs. We are given a source node s and a destination node t, together with a duration threshold $DT > 0$ and a cost threshold $CT > 0$.

Definition 2 (CDP). A *duration-bounded path* is a path from s to t such that $D(P) \ge DT$. The **minimum Cost Duration-bounded Path (CDP)** routing problem seeks a path P from s to t with minimum cost among all duration-bounded paths.

Definition 3 (DCP). A *cost-bounded path* from s to t is a path such that $C(P) \le CT$. The **maximum Duration Cost-bounded Path (DCP)** routing problem seeks a path P from s to t with maximum duration among all cost-bounded paths.

4. Reliable ad hoc routing

In this section, we will present an offline prediction algorithm and two routing algorithms. After running our prediction algorithm, we will obtain a link duration prediction table. By looking up this table, we can find a worst-case duration value for each possible wireless link in the MANET. Then we present our routing algorithms which compute reliable routing paths based on the duration prediction table. We also discuss applications of our prediction table and our routing algorithms in ad hoc routing protocol design.

4.1. The prediction algorithm

In order to predict the link duration, we need to create an auxiliary directed graph $G_A(V_A, E_A)$ based on the waypoint graph G_W by adding some *landmarks* into every road segment of G_W. The distance between every two consecutive landmarks is the same and is called a *distance unit*. The vertex set V_A of G_A corresponds to the union of waypoints in G_W and newly added landmarks. The edge set E_A of G_A corresponds to the union of those separated road segments. However, in order to decide how many landmarks need to be added for a specific road segment, we need to introduce the concept of *role*. A mobile node can be *a walking person*, *a running person*, or *a vehicle*, which are called the *roles* of a mobile node. Based on its role, we can decide the speed of a 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. Normally, people run/walk at roughly the same constant speed on different road segments. Of course, we can define more roles to make the prediction more precise, e.g. a fast/slow running person. No matter which role of a node is in, we assume that the number of landmarks on a road segment is a multiple of the number of landmarks it passes within one time unit. *Although this may not be exactly true in practice, we can always find a close approximation*. Therefore, once we know the waypoint graph and all types of roles and their speeds on different road segments, we can compute the minimum number of landmarks required to be added into every road segment. We also assume that initially every mobile node will be at some vertices of G_A. The prediction precision can be improved by adding more landmarks. However, this will increase the complexity of the system.

We label every waypoint, road segment in G_W and landmark in G_A. In the following, a road segment always means the whole road segment between two waypoints, which may contain several landmarks. For two vertices in G_W corresponding to two landmarks, they may have the same $LandmarkID$. However, a tuple (*LandmarkID*, *SegmentID*) can uniquely identify a vertex on graph G_W if the vertex corresponds to a landmark. If a vertex in G_A corresponds to a waypoint, its *LandmarkID* will be a negative value whose absolute value is its *WaypointID*. We can imagine a waypoint as a special landmark. We also note that a 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. Therefore, a triple ($RoleID$, *LandmarkID*, *SegmentID*) will be sufficient to uniquely identify any possible mobile node, which are called the *mobility parameter* of the mobile node.

Now we are ready to introduce the concept of *possible link*. For each pair of vertices in G_A whose Euclidean distance is no more than the transmission range R, we will have one possible link whose two ends correspond to the pair of vertices. So every possible link for the given graph G_A 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 G_W (note that there is a unique G_A corresponding to a given G_W) and we denote the set of possible links on G_A by $L(G_A)$. No matter how a MANET $G(V, E)$ is deployed on the waypoint graph G_W, for each node in G, we will have a vertex in G_A corresponding to it. Similarly, for each wireless link in G, we will have a possible link in $L(G_A)$ corresponding to it. For example, suppose $RoleID = 1$ represents a walking person and *RoleID* = 3 represents a vehicle, then (1, 8, 13, 3, 7, 7) represents a possible link corresponding to a wireless link of G whose one end node u is a walking person at the landmark (8, 13) moving along road segment 13 and whose other end node is a vehicle at the landmark (7, 7) moving along road segment 7.

The duration prediction table will be indexed by a 6-tuple ($RoleID_u$, $LandmarkID_u$, $SegmentID_u$, $RoleID_v$, $LandmarkID_v$, $SegmentID_v$). Each entry of the table corresponds to a possible link in $L(G_A)$ 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. 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 ($RoleID_u$, $LandmarkID_u$, $SegmentID_u$, $RoleID_v$, $LandmarkID_v$, $SegmentID_v$, $duration$). Each entry of the table corresponds to a possible link in $L(G_A)$ and indicates (by YES or NO) whether this possible link can last $duration$ time units in the worst case. We propose Algorithm 1 to compute the AD_Table. Once we obtain the AD_Table, we can compute the duration prediction table by a simple transformation.

Algorithm 1 AD_Table Computation Algorithm

INPUT: $G_A(V_A, E_A)$, N_{role}, N_{lm}, N_{seg}, D_{max}.

OUTPUT: AD_Table.

```
Step_1 forall possible link (Ru,Lu,Su,Rv,Lv,Sv) and d do
         if (d = 0) then
           AD_Table(Ru,Lu,Su,Rv,Lv,Sv,d):=YES;
         else
           AD_Table(Ru,Lu,Su,Rv,Lv,Sv,d):=NO;
         endif
       endfor

Step_2 for d := 1 to Dmax do
         forall possible link (Ru,Lu,Su,Rv,Lv,Sv), s. t.
           AD_Table(Ru,Lu,Su,Rv,Lv,Sv,d − 1)=YES
           do
           AD_Table(Ru,Lu,Su,Rv,Lv,Sv,d):=YES;
           forall Ru+,Lu+,Su+,Rv+,Lv+,Sv+ do
             if AD_Table(Ru+,Lu+,Su+,Rv+,Lv+,Sv+,d − 1)
                =NO or no corresponding entry
             then
               AD_Table(Ru,Lu,Su,Rv,Lv,Sv,d):=NO;
               break;
             endif
           endfor
         endfor
       endfor
```

In Algorithm 1, we use N_{role}, N_{lm}, N_{seg}, and $D_{\max}$ to denote the number of roles, the number of vertices in G_A, 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 possible combination of role, location and direction, after one time unit.

Algorithm 1 is a dynamic programming algorithm. In **Step_1**, we initialize the AD_Table. In **Step_2**, for a specific possible link (R_u, L_u, S_u, R_v, L_v, S_v), we compute all possible

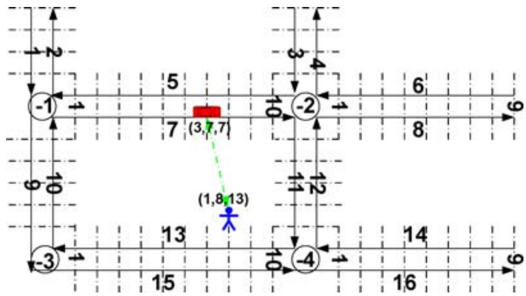

Fig. 1 A possible link $(1, 8, 13, 3, 7, 7)$

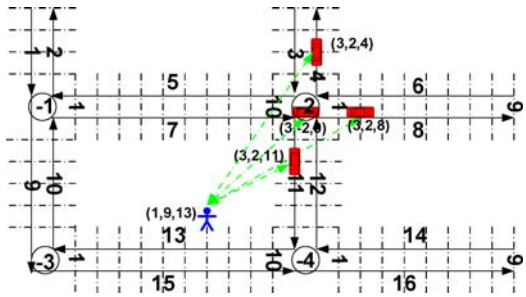

Fig. 2 All possible R_u^+, L_u^+, S_u^+, R_v^+, L_v^+, S_v^+ after one time unit

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. The role of a node will never change during the entire procedure. We iteratively increase duration d and test all next possible R_u^+, L_u^+, S_u^+, R_v^+, L_v^+ and S_v^+ to see if the considered link can last one more time unit. If in all those next possible cases, we find that the corresponding possible links can last at least $d - 1$ time units, we know that the possible link we are considering $(R_u, L_u, S_u, R_v, L_v, S_v)$ can last at least d time units. Otherwise, i.e., if testing fails (get a NO) in any of possible cases, we know that this link does not guarantee to last d time units. Obviously, this is a worst-case prediction.

We will use the a simple example to illustrate the computation of AD_Table. Figure 1 shows a part of a waypoint graph. The directed edges and circles represent the corresponding road segments and waypoints, respectively. We use dotted lines to represent the corresponding landmarks for better illustration. All landmarks, waypoints and road segments are labeled as shown in the figure. Suppose we consider the aforementioned possible link $(1, 8, 13, 3, 7, 7)$ (fig. 1) and start from $d = 0$. We assume the vehicle and the person move at the speeds of 6 and 1 distance unit(s) per time unit respectively and assume the distance between any two landmark/wapoints in the figure is less than R. According to the moving rules, the mobility parameter of the person node must be $(1, 9, 13)$ and the mobility parameter of the vehicle node may be any one of $(3, 2, 4)$, $(3, 2, 8)$, $(3, 2, 11)$ or $(3, -2, 0)$ after one time unit (fig. 2). Here, 0 in the SegmentID field means that the vehicle stays at the waypoint 2. Since $AD\_Table(1, 9, 13, 3, 2, 4, 0)$, $AD\_Table(1, 9, 13, 3, 2, 8, 0)$, $AD\_Table(1, 9, 13, 3, 2, 11, 0)$ and $AD\_Table(1, 9, 13, 3, -2, 0)$ are all equal to YES, we conclude that $AD\_Table(1, 8, 13, 3, 7, 7, 1) := \text{YES}$. We repeat this process and can finally find the worst-case duration of the possible link $(1, 8, 13, 3, 7, 7)$.

Algorithm 2 CDP Routing

INPUT: MANET $G(V, E)$, the mobility parameter $(R_v, L_v, S_v)\ \forall\ v \in V$, R and the link cost function C, connection request $\rho = (s, t, DT)$.

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

Step_1 Construct a graph $G_B(V, E_B)$ in the following way. The set of vertices of G_B contains all n mobile nodes $v_1, v_2, \ldots, v_n$ in G. The set of undirected edges E_B of G_B contains all pairs $(u, v) \in N_B \times N_B$, such that $d(u, v) \le R$ and $D(u, v) \ge DT$, where $D(u, v) = DP\_Table(R_u, L_u, S_u, R_v, L_v, S_v)$. The cost of the link (u, v) is set to $C(u, v)$.

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 then**
  Block the connection request ρ.
else
  Output the found path.
endif

After we obtain the AD_Table, we can compute 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. Let us use N_{deg} to denote the maximum outgoing degree of all waypoints in G_W. Based on our assumptions, we have at most $N_{\text{role}}^2 \times N_{\text{lm}}^2 \times N_{\text{deg}}^2$ possible links. Computing the AD_Table will take $O(N_{\text{role}}^2 \times N_{\text{lm}}^2 \times N_{\text{deg}}^2 \times D_{\text{max}})$ and constructing DP_Table from AD_Table will take linear time in terms of the table size. The sizes of the tables could be very large for relatively large waypoint graphs and the computation time could be long. However, the duration prediction table can be computed offline once a waypoint graph is given. The waypoint graph changes only if the corresponding city street map changes, which does not happen frequently. Therefore, the duration prediction table needs not to be recomputed frequently. The waypoint graph, the corresponding auxiliary graph, and the duration prediction table can be stored in every mobile node and used repeatedly.

4.2. The routing algorithms

Now we are ready to present our routing algorithms. Firstly, we present an algorithm which is able to optimally solve the minimum Cost Duration-bounded Path (CDP) routing problem defined in Section 3. The algorithm is formally presented as Algorithm 2. We 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 (u, v) takes 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 $O(m + n \log n)$ time, which is no more than $O(n^2)$. 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. □

Algorithm 3 DCP Routing

INPUT: MANET $G(V, E)$, the mobility parameter (R_v, L_v, S_v) $\forall$ $v \in V$, R and the link cost function C, connection request $\rho = (s, t, CT)$.

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

Step_1 Compute the set of distinct values $0 \leq D_1 < \cdots < D_k$ such that for every pair of nodes (u, v) such that $d(u, v) \leq R$ and there is some index i $(1 \leq i \leq k)$ such that $D(u, v) = D_i$, where $D(u, v) = DP\_Table(R_u, L_u, S_u, R_v, L_v, S_v)$. Let $\mathcal{D} := \{D_1, D_2, \ldots, D_k\}$.

Step_2 Use binary search on $\mathcal{D}$ to find the largest D_i such that the solution to corresponding **CDP** routing problem computed by Algorithm 2 has a cost no more than CT.

Step_3 **if** such a value cannot be found in **Step_2** **then**
  Block the connection request ρ.
else
  Output the found path.
endif

Algorithm 3 is a heuristic algorithm for the DCP routing problem. It uses Algorithm 2 as a subroutine for $O(\log n)$ times while using binary search on 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 the DCP problem. However, our prediction is a worst-case prediction. Therefore, the actual duration of a link may be 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 will use simulations to show that paths found by Algorithm 3 are actually reliable in most cases.

How to apply the prediction table and proposed routing algorithms in routing protocol design is a very important issue. The most direct way is to use them with the proactive routing protocols such as OLSR (Clausen et al., 2001). By piggybacking the mobility parameters in their control messages, the message exchange mechanisms of proactive routing protocols can be used for every mobile node to learn the current network topology and mobility statuses of other nodes. After gathering those information, our routing algorithms can be employed to find reliable routing paths. As mentioned in the above, the message exchange overhead of a proactive routing scheme is high, especially for large size networks with relatively high mobility. Actually, our prediction and routing schemes can also work with the on-demand routing protocols. As argued in Papadimitratos et al. (2002), the flooding based route discovery mechanisms of on-demand routing protocols such as AODV (Perkins and Royer, 1999) and DSR (Johnson et al., 2003), together with some enhanced discovery schemes like diversity injection (Pearlman and Haas, 1999), can be used to discover a partial view of the underlying network topology and node statuses with low message exchange overheads when a connection request arrives. Based on this information, we can construct the auxiliary graph G_B and apply our routing algorithms. In addition, we can also use our prediction table during

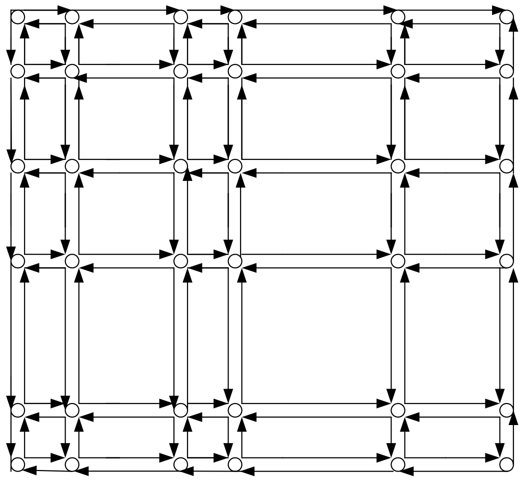

Fig. 3 A sample waypoint graph

the route discovery of on-demand routing protocols. Taking the DSR and CDP routing as an example, when an intermediate node receives a RREQ (Route REQuest) message, it can check whether the partial path from source node recorded in this RREQ message is still feasible in terms of path duration based on the prediction table and the mobility parameters of its precursor and itself. If not, it drops the packet. Otherwise, it attaches its ID in the packet and broadcasts it to all its neighbors. Based on all received RREQ packets, the destination node can pick a feasible path with minimum cost and send the RREP (Route REPly) message back to the source node along the found path.

5. Performance evaluation

In this section, we evaluate the performance of our algorithms via simulations. We randomly generate grid-like waypoint graphs. Figure 3 shows a sample waypoint graph. In the simulations, all generated waypoint graphs 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 90 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 (in terms of Euclidean distance) 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 in [18, 30] time units. Then it randomly chooses another destination and repeats the above procedure. The number of roles is 2. One role is a walking person moving at a speed of 1 distance unit per time unit. The other is a vehicle. We assign the vehicle speed limits of all road segments to be 4 distance units per time unit. Every mobile node has the same fixed transmission range, 250 meters. In all simulations, we employ *hop count* as the metric to measure path cost, i.e., the shortest path algorithm algorithm always finds minimum hop count paths. It is a commonly used metric since it represents the number of transmissions or the delay of a routing path.

In the first two experiments, we compare the performance of our CDP routing algorithm (Algorithm 2) with the Shortest Path (SP) routing algorithm in terms of network throughput. Every 30 time units, 10 connection requests are injected into the network, whose source and destination nodes are randomly chosen. Totally, 1000 connection requests will be generated

Table 1 Network throughput on different waypoint graphs

WG	SP (%)	CDP (%)	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 2 Throughput of networks with different sizes

Size	SP (%)	CDP (%)	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

in each run. The duration threshold is randomly picked as 1 or 2 time units. When a connection request arrives, each algorithm will be used to compute a single source to destination path for routing. If the algorithm fails to find such 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 ratio between the number of successfully established connections and the total number of generated connection requests (1000) and use it to represent the network throughput. In the first scenario, we create a mobile network with 60 nodes and run simulations on 10 different randomly generated waypoint graph. In the second scenario, we run simulations on one waypoint graph, but randomly generate 5 different mobile networks with 40, 60, 80, 100, 120 nodes respectively. Results are shown in the Tables 1 and 2.

Intuitively, the 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 parameters. We can see that the network throughput becomes higher and higher no matter which algorithm is used. From Table 2, 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 relatively 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 gain more chances to have durable links when computing paths, i.e., gain more chances to satisfy the given duration thresholds and eventually improve throughput. We find out that with regards to network throughput, our algorithm outperforms the shortest path algorithm more than 100% in most cases.

In the other two experiments, our DCP routing algorithm (Algorithm 3) is compared with the shortest path routing algorithm in terms of path durations and hop counts. Similar to last experiments, we again totally inject 1000 connection requests with random sources and destinations, 10 requests each time unit. 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 our DCP algorithm finds paths whose actual durations are less than the corresponding minimum hop count paths, and the total number of connection requests (1000). The cost threshold in our DCP routing algorithm is set to be $bound\_ratio \times MH$, where MH is the minimum hop count for a given source and destination node pair in the network and *bound_ratio* can be any number no less than 1.0. We perform simulations firstly on different waypoint graphs and then on mobile networks with different sizes. Simulation results are presented in the following six tables.

These tables show the average path duration increase, path hop count increase and failure ratios given by our DCP routing algorithm (cost *bound_ratio* is set to ∞, 1.0, 1.2, 1.5 and 2.0 respectively) compared to the shortest path routing. Here, by "average", we mean an average over 1000 paths since totally we have 1000 connection requests in each run. If the $bound\_ratio = \infty$, it means that we do not put any constraint on hop count when computing a 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 3, on 4 different waypoint graph instances, the average duration of paths given by our algorithm with $bound\_ratio = \infty$ is even shorter than that of the corresponding minimum hop count paths. Furthermore, the average hop counts of those paths are 33.9% more than that of the minimum hop count paths. We can also see that the average failure ratio is as high as 22.3% from Table 5. That is to say, when setting $bound\_ratio = \infty$, for 22.3% connection requests, our algorithm fails to come up with solutions as good as (in terms of path duration) those given by the shortest path algorithm. This is due to the fact that our duration prediction is a worst-case prediction, i.e., the actual duration of a link will probably last longer than the predicted value in the simulation. Therefore, paths found by our algorithm are not guaranteed to last longer than those minimum hop count paths. On the other hand, by setting $bound\_ratio = \infty$, the found path will include more number of hops than

Table 3 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 4 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

Table 5 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

the corresponding minimum hop count 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 can decrease its hop count and hopefully prolong its duration. The simulation results verify it. If this bound ratio is set to 2.0, we obtain similar results as those 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 node 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, which is less than 5% in most cases (Tables 4 and 7). In addition, for about 90% of the connection requests, paths founded by our algorithm with proper bound ratio setting will last at least as long as those minimum hop

Table 6 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 7 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 8 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

count path without or with minor cost (hop count) increase, and the average path duration improvement is around 8%.

6. Conclusions

In this paper, we have studied the reliable ad hoc routing in the urban environment. We formulated two optimization problems, namely, minimum Cost Duration-bounded Path (CDP) routing problem problem and maximum Duration Cost-bounded Path (DCP) routing problem. We have presented an offline algorithm to predict the worst-case duration of possible wireless links. Based on our prediction, we then presented an efficient algorithm to optimally solve the CDP problem which computes a path connecting a given source node and a destination node, whose total cost is minimum and whose duration is no smaller than a given threshold. We also proposed a heuristic algorithm for the DCP problem which computes a path with hopefully long duration and with cost no more than a given threshold. In addition, we have shown that the proposed prediction and routing schemes can be easily applied to design reliable ad hoc routing protocols. Simulation results showed that compared with the shortest path routing, our CDP routing algorithm improves the network throughput by more than 100% in most cases and that our DCP routing algorithm improves the average path duration by about 8% without increasing or with slight increase of the average path cost.

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 support fault-tolerance at the same time. In addition, we will investigate new and more precise schemes to predict the actual wireless link duration.

References

Bai F, Sadagopan N, Helmy A (2003) IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks. In: Proceedings of IEEE INFOCOM, pp. 825–835

Broch J, Maltz DA, Johnson DB, Hu YC, Jetcheva J (1998) A performance comparison of multi-hop wireless ad hoc network routing protocols. In: Proceedings of ACM MobiCom, pp. 85–97

Clausen T, Jacquet P, Laouiti A, Muhlethaler P, Qayyum A, Viennot L (2001) Optimized link state routing protocol for ad hoc networks. In: Proceedings of IEEE INMIC, pp. 62–68

Hong X, Gerla M, Pei G, Chiang CC (1999) A group mobility model for ad hoc wireless networks. In: Proceedings of ACM/IEEE MSWiM, pp. 53–60

Jiang S, He D, Rao J (2001) A prediction-based link availability estimation for mobile ad hoc networks. In: Proceedings of INFOCOM, pp. 1745–1752

Johnson DB, Maltz DA, Hu YC (2003) The dynamic source routing protocol for mobile ad hoc networks (DSR). IETF Draft

Lee SJ, Su W, Gerlaw M (2001) Wireless ad hoc multicast routing with mobility prediction. Mobile Networks and Applications 6:351–360

Lim G, Shin K, Lee S, Yoon H, Ma JS (2002) Link stability and route lifetime in ad hoc wireless networks. In: Proceedings of IEEE ICPPW, pp. 116–123

Lou W, Liu W, Fang Y (2004) SPREAD: Enhancing data confidentiality in mobile ad hoc networks, In: Proceedings of IEEE INFOCOM, pp. 2404–2413

Papadimitratos P, Haas J, Sirer EG (2002) Path set selection in mobile ad hoc networks. In: Proceedings of IEEE MobiHoc, pp. 1–11

Pearlman MR, Haas ZJ (1999) Improving the performance of query-based routing protocols through °diversity injection. In: Proceedings of IEEE WCNC, pp. 1548–1552

Pei G, Gerla M, Hong X (2000) LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility. In: Proceedings of ACM MobiHoc, pp. 11–18

Perkins CE (2001) Ad hoc networking, Addison-Wesley

Perkins CE, Royer EM Ad-hoc on-demand distance vector routing. In: Proceedings of IEEE WMCSA, pp. 90–100

Sadagopan N, Bai F, Krishnamachari B, Helmy A (2003) PATHS: Analysis of PATH duration statistics and their impact on reactive MANET routing protocols. In: Proceedings of ACM MobiCom, pp. 245–256

Stojmenovic I, Russell M, Vukojevic B (2000) Depth first search and location based localized routing and QoS routing in wireless networks. In: Proceedings of IEEE ICPP, pp. 173–180

Su W, Lee SJ, Gerla M (2001) Mobility prediction and routing in ad hoc wireless networks. Iternational Journal of Network Management 11:3–30

Toh CK (1999) Long-lived ad hoc routing based on the concept of associativity. IETF Draft

Wang L, Zhang L, Yang O, Shu Y, Dong M (2000) Multipath source routing in wireless ad hoc networks. In: Proceedings of IEEE CCECE, pp. 479–483

Wu K, Harms J (2002) Multipath routing for mobile ad hoc networks. IEEE Journal of Communications and Networks, Special Issue on Innovations in Ad Hoc Mobile Pervasive Networks 4:48–58