

On-Demand Location-Aided QoS Routing in Ad Hoc Networks *

Chun Huang, Fei Dai, and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

Abstract

With the development and application of position devices, location-based routing has received growing attention. However, little study has been done on QoS routing with the aid of location information. The existing location-based routing approaches, such as flooding-based routing schemes and localized routing schemes, have their limitations. Motivated by ticket-based routing, we propose an on-demand location-aided, ticket-based QoS routing protocol (LTBR). Two special cases of LTBR, LTBR-1 and LTBR-2, are discussed in detail. LTBR-1 uses a single ticket to find a route satisfying a given QoS constraint. LTBR-2 uses multiple tickets to search valid routes in a limited area. All tickets are guided via both location and QoS information. LTBR has lower overhead compared with the original ticket-based routing, because it does not rely on an underlying routing table. On the other hand, LTBR can find routes with better QoS qualities than traditional location-based protocols. Our simulation results show that LTBR-1 can find high quality routes in relatively dense networks with high probability and very low overhead. In sparse networks, LTBR-2 can be used to enhance the probability of finding high quality routes with acceptable overhead.

1 Introduction

An ad hoc network is a temporary network composed of wireless mobile hosts (nodes), where wireless links are setup or destroyed whenever one node moves in or out of transmission ranges of other nodes. Routing in ad hoc networks is challenging due to this dynamic nature. Existing routing protocols in ad hoc networks are either *proactive* or *on-demand*. Proactive protocols pre-compute and store paths to all possible destinations for each node in a routing table, and during the routing process, each intermediate node simply sends data to the next hops based on the

routing table. On-demand protocols compute the path upon each route request. Proactive protocols have low routing discovery overhead, however, they need to maintain routing information. On-demand protocols avoid the huge expense of updating routing tables, however, as flooding is always used to search paths, the routing overhead is high.

Quality-of-Service (QoS) routing has been motivated by multimedia applications, such as voice channels, live videos, and document transfer. QoS routing selects paths based on QoS metrics to satisfy specific requirements, such as end-to-end delay, delay jitter, bandwidth, and packet loss probability. These metrics can be classified into three classes [11]: additive, multiplicative, and concave. QoS routing is much more complicated than regular routing in that routing under multiple constraints is needed under many circumstances. Wang and Crowcroft [11] have shown that the problem of finding a path subject to constraints on two or more additive and multiplicative metrics is NP-complete. In ad hoc networks where the network topology changes arbitrarily, QoS routing is even more challenging.

With rapid development of positioning techniques, location-aided routing has received wide attention. Previous study shows that location information can effectively assist the routing process [1, 5, 6, 8, 9, 10]. However, existing location-based routing protocols have their limitations. Both localized algorithms and flooding-based algorithms have either unsatisfactory success rate or undesirable routing overhead. In addition, previous solutions are designed to deal with best-effort data traffic, giving no consideration to optimizing the QoS properties. The ticket-based probing scheme (TBP) [3] is a novel heuristic approach for solving QoS routing problems. In TBP, a ticket represents a permission for searching a path. The total number of tickets is determined at the start of routing and controls the exploring scope in the routing process. TBP has been shown to achieve as high a success rate as flooding, while incurring as low a routing overhead as single path routings. However, TBP is a proactive protocol where periodical update is applied to maintain the routing tables, which makes it deficient and unscalable.

*This work was supported in part by NSF grants CCR 0329741, ANI 0073736 and EIA 0130806. Contact address: jie@cse.fau.edu

Motivated by location-aided routing and ticket-based routing, we present a new QoS routing protocol called location-aided ticket-based routing (LTBR). By combining the low overhead nature of location-based routing and QoS awareness of ticket-based routing, LTBR can find high quality routes with relatively low maintenance overhead. Two special cases of LTBR are examined in this paper. LTBR-1 uses a single ticket to find a route that satisfies a given QoS constraint. This single ticket is guided by a location and QoS aware metric called progress-over-cost (i.e., progress toward the destination in unit cost). At each step of the probing, the ticket is relayed from one node to the next node that maximizes the progress-over-cost. In addition, a technique called backward checking is used to further enhance the QoS measure of a discovered route. LTBR-2 allows limited ticket duplication. That is, each node can send at most two copies of a received ticket to the selected successors. The two successors are selected within a certain range of directions, so that the dissemination of the tickets is restrained in a small area. On the other hand, the angle between two successors is larger than a minimum value, so as to avoid excessive redundancy within the searching area. Simulation results show that LTBR-1 has similar success ratio and overhead to single path location based protocols, such as GEDIR and MFR, and the found paths have better QoS measures. LTBR-2 has similar overhead and path quality to location based protocols using limited flooding, such as LAR, and achieves higher success ratio.

Contributions of this paper include the new location and QoS aware metric that guides the dissemination of tickets, the backward checking mechanism that enhances the route quality, and techniques to use multiple tickets searching a limited area. The effectiveness of those mechanisms are also confirmed by the simulation study.

2 Related Work

Location-based routing. Stojmenovic [8] classified location-based routing schemes into two categories: *localized* approaches and *flooding-based* approaches.

Localized approaches are marked by single path strategy and greedy principle. In the routing process, one node always selects *one* neighbor as the successor to forward the message, based on the locations of the destination, the current node and the neighbors. MFR (Most Forward within Radius) [10], GEDIR (GEographic DIstance Routing) [9] and DIR (Compass Computing) [6] are three typical protocols. In MFR, a node always sends the packet to the neighbor with the greatest progress. In GEDIR, the packet is sent to the neighbor which is the closest to the destination. In DIR, a node selects a neighbor, such that the angle constructed by the selected neighbor, the source and the destination will be minimized.

Flooding-based approaches are marked by a flooding-based strategy, where messages are flooded through the whole network area or a portion of the network. LAR (Location-Aided Routing) [5] and DREAM (Distance Routing Effect Algorithm For Mobility) [1] are two flooding-based protocols. LAR has two schemes. In LAR1, messages are sent to neighbors within the request zone, which is defined as the minimum rectangle that contains the source and the expected zone, a circular area centered at the destination with radius $v(t_1 - t_0)$, where v is the velocity. In LAR2, messages are sent to neighbors which are closer to the destination than the current node. In DREAM, a direction range is selected at the source and every intermediate node, in such a way that the probability of finding the destination in this range is higher than a given threshold, and only neighbors of the current node within such direction range will receive the message.

The localized approach has the advantage of low routing overhead, but it cannot achieve a satisfactory success rate under some circumstances, such as in sparse networks. The flooding-based approach has relatively higher success rate. However, it is accompanied by high routing overhead. Besides, LAR1, LAR2 and DREAM do not have competitive performances in sparse networks, where it is not uncommon that the neighbors of a node are out of the predefined request zone or the direction range. Furthermore, in LAR1 and DREAM, the request zone or the directional range is primarily determined by the mobility of the destination. Other factors, such as the likelihood that intermediate nodes may deviate from this area, are ignored. This reduces the possibility of finding paths under certain circumstances. For example, when the destination is still, the direction range in DREAM will be a line segment connecting the source and the destination. Similarly, in LAR1, if the destination is still and has the same x or y coordinate with the source, the request zone also becomes a line segment. Overall, there is no existing location-based routing protocol that can achieve satisfactory performance in both low routing overhead and high success rate.

Boukerche et al [2] proposed using location information to reduce the flooding cost in on-demand route discovery processes. When the route request message for source s arrives at an intermediate node v , the angle $\angle svx$ will be calculated for each neighbor x . Only neighbors with the angle $\angle svx$ larger than a given *screening angle* are selected as successors. The number of successors can be reduced significantly with a larger screening angle. However, this approach still selects many successors in dense networks, and the selection of successors is not guided by the destination location.

Ticket-based probing. *Ticket-based probing scheme* (TBP) [3] is a distributed routing scheme to solve two QoS routing problems. One is the delay-constraint least-cost routing

problem (referred to as the DCLCR problem), which is to find a path that satisfies the delay constraint and has the least cost among all qualified paths. Another one is the bandwidth-constraint least-cost routing problem (referred as the BCLCR problem), which is to find a path that satisfies the bandwidth constraint and has least cost among all qualified paths. In TBP, an imprecise state model is first set up based on the distance vector scheme, then end-to-end state information for all destinations is stored and periodically updated in each node's routing table. For example, to solve DCLCR problem, node i keeps (*next-hop*, $D_i(d)$, $\Delta D_i(d)$) for each destination d , where *next-hop*, $D_i(d)$, $\Delta D_i(d)$ respectively represent the next hop, the end-to-end delay and expected maximum change of delay of the least-delay path from i to d . In the routing process, $D_i(d) + \Delta D_i(d)$ and $D_i(d) - \Delta D_i(d)$ are respectively used to estimate the worst and best expected end-to-end delay from i to d .

For a route request to node d with delay constraint D , source s first determines total number of tickets, N_0 , by the following rules: request is rejected if the possibility to find a feasible path is too low (i.e., $D < D_s(d) - \Delta D_s(d)$); N_0 is set to a small constant if such possibility is high enough (i.e. $D > D_s(d) + \Delta D_s(d)$). Otherwise, N_0 is set to a value between 1 and a constant such that for a more stringent QoS requirement, a larger N_0 is assigned. Then s distributes N_0 tickets among its neighbors in a way such that a neighbor with lower estimated end-to-end delay to the destination will be given more tickets. Then, s sends a probe with assigned number of tickets to each selected neighbor. Each probe contains information of the QoS route request and the path the ticket travels along. Any node receiving a probe will distribute its received tickets among the neighbors in the same way as s , except it finds the ticket violates the QoS constraints. Thus a probe arriving at the destination contains a path that satisfies the QoS constraints. Note that, at anytime in the routing process, the total amount of tickets contained in all probes for this request is N_0 .

TBP is one of the few routing protocols that can achieve a high success ratio while producing low routing overhead. However, TBP is a proactive protocol where the routing relies on the pre-computed QoS state information, which is expensive to be maintained. Therefore, TBP is not scalable.

3 Protocol Overview

3.1 System model

We focus on QoS routing where an additive metric such as delay is used. Let $d(l)$ be the QoS metric of a link l and \oplus the additive operation, the QoS measure of path $p = (l_1, l_2, \dots, l_n)$ is $d(p) = d(l_1) \oplus d(l_2) \oplus \dots \oplus d(l_n)$. For example, when delay is used as the QoS metric, $d(l_i) \oplus d(l_j) = d(l_i) + d(l_j) + c$. c is the constant processing time

at each node. To simplify the discussion, c is assumed to be 0. A QoS routing request is a triple $Q(s, d, D)$, where s is the source, d is the destination, and D is a QoS constraint. A QoS routing protocol accepts such a request and finds a path $p = (s = l_1, l_2, \dots, l_n = d)$ such that $d(p) \leq D$.

For simplicity, we assume that each node in the network is aware of its geographical location. Each node v knows its neighbor set $N(v)$ and locations of its neighbors. In on-demand routing protocols, it is a challenging task for each node to obtain and maintain the location information of other nodes. Several proactive solutions exist for this *location management problem*. DREAM [1] maintains a location table at each node via controlled flooding. GLS (Grid Location Service) [7] and DLM (Distributed Location Management) [12] have location servers to answer location queries.

In our protocol, we use a simple on-demand location management scheme similar to the route discovery process in DSR [4]. When the location information of the destination is needed, the source broadcasts a location query message. On receiving the location query, the destination sends a location reply message back to the source. In order to reduce the flooding overhead, each node maintains a location cache that contains a list of known locations, and issues a location query only when the demanded location is missing in the location cache. Two techniques in DSR, expanding ring search and reply with cache, can be used to further reduce the overhead. The location information is also piggybacked in each normal packet for location cache update. In the following discussion, we assume the source and intermediate nodes in the QoS routing process know the location of the destination, and this location information is accurate enough to guide the route search.

3.2 Location-aided ticket-based routing

We propose location-aided ticket-based routing (LTBR), an on-demand and localized QoS routing protocol that combines location-based routing and ticket-based routing. In LTBR, a ticket is a special control message for searching QoS paths. A ticket contains the information of the QoS routing request and traversed path. In the routing process, one ticket is originally generated at the source and sent to neighbors selected by certain ticket assignment rules. Any intermediate node receiving a ticket either ignores it or sends tickets to successors selected in the same way as the source. A ticket will be forwarded until reaching the destination, or stopped at an intermediate node if it violates the QoS constraints. A ticket arriving at the destination indicates the success in finding a path.

The routing mechanism of LTBR resembles the ticket-based routing process in TBP [3], with two differences: (1) LTBR uses location information to guide ticket forwarding

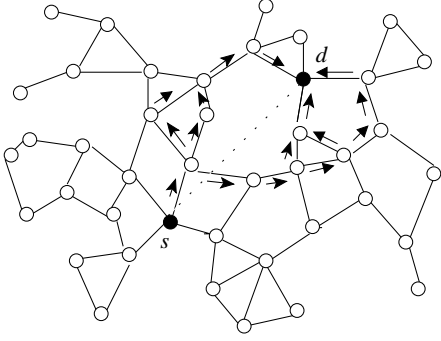


Figure 1. Location-aided ticket-based routing.

and does not need the underlying routing table in TBP; (2) LTBR allows issuing multiple tickets in the intermediate nodes to improve the success rate with controllable overhead. As shown in Figure 1, in the routing process for source s to destination d , each intermediate node uses certain ticket assignment rules to select successors. The selection is based mainly on QoS metrics of adjacent links, and locations of the destination d , current node, and neighbors of the current node. By such rules, the tickets are guided toward the destination. Hopefully one or more tickets will arrive at the destination after exploring different paths.

The routing process can be described briefly as follows. Upon QoS routing request $Q(s, d, D)$, the source s constructs a ticket that contains the QoS routing request and traversed path, and sends it to one or more neighbors selected by certain *ticket assignment rules*. An intermediate node v receiving a ticket t for Q will first check whether QoS constraints are violated and ignore it if they are. If t is the first ticket v receives for Q , v selects successors and forwards t to all of the successors after adjusting the traversed path information in the ticket.

When destination d receives the first ticket for Q , it waits for a period of time to receive all possible tickets for Q . If at least one received ticket fulfills the QoS requirements, d picks one ticket t_1 from all received tickets which has the best QoS measure, constructs a route reply r and sends it to the source. The route reply contains field *returnPath*, which is copied from reversed $t_1.traversedPath$ and guides the reply route to the source. Any node receiving route reply r updates the remaining part of $r.returnPath$ with the path information stored in the optimal ticket, and relays it to the next hop along $r.returnPath$. If a route reply is not received by the source after a certain period of time, this routing process is viewed as failed.

The effectiveness (i.e., success ratio) and efficiency (i.e., routing overhead) of LBTR is determined by the ticket assignment rule that decides the number and selection of suc-

cessors in forwarding tickets. We focus on two ticket assignment schemes: one-ticket LBTR where a single ticket is used for each QoS request, and two-ticket LBTR where each intermediate node (including the source) can send at most two tickets to two neighbors.

4 Special Cases

4.1 One-ticket LTBR

One-ticket LTBR (LTBR-1) is a single path routing protocol, where at each step at most one successor is selected. LTBR-1 resembles GEDIR and MFR in that every node selects a successor based on the location information. However, with two enhancements, a hybrid metric called *progress-over-cost* and an optimization mechanism called *backward checking*, it improves QoS properties of the found paths significantly, which is confirmed by our simulation results.

Algorithm 1 ASSIGN-ONE-TICKET (u, v, d)

```

1: if  $d \in N(v)$  then
2:   return  $\{d\}$ 
3: else
4:    $N_1(v) \leftarrow N(v) - \{u\}$ 
5:   if  $N_1(v) \neq \emptyset$  then
6:      $x_1 \leftarrow \text{BEST}(v, N_1(v), d)$ 
7:     return  $\{x_1\}$ 
8:   else
9:     return  $\emptyset$ 

```

Algorithm 1 describes the ticket assignment rule of LTBR-1. When the source or an intermediate node v has a ticket to destination d , it first checks if d is in its neighbor set $N(v)$. If it is, the ticket is sent directly to the destination (steps 1–2). If v has at least one neighbor other than the predecessor u (u is empty if v is the source), it selects the “best” neighbor x_1 as its successor and forwards the ticket to x_1 ; otherwise no successor is selected, and the ticket is dropped (steps 4–9). v ’s selection of the “best” successor from the set of candidates $N_1(v)$ ($N(v)$ by excluding predecessor u) with respect to the destination d is based on the hybrid metric described in the next subsection.

4.2 A location and QoS aware metric

In localized location-based routing protocols, every intermediate node selects the successor by applying greedy principle on a certain metric. In LTBR, a metric called *progress-over-cost* is used, which combines the consideration for both location and QoS metrics. Given the current node v , destination d , and v ’s neighbor x , the *geographic progress* $p(v, x, d)$ of successor x is the projection of line vx onto line vd . For example, in Figure 2,

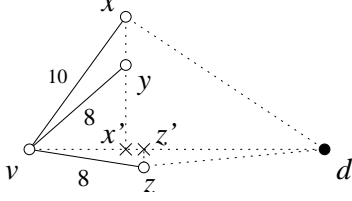


Figure 2. Progress-over-cost.

$p(v, x, d) = p(v, y, d) = |vx'|$ and $p(v, z, d) = |vz|$. The progress-over-cost of successor x is defined as

$$p_d(v, x, d) = \frac{p(v, x, d)}{d(v, x)}$$

where $d(v, x)$ is the additive QoS metric of link (v, x) .

Progress-over-cost is easy to understand. By viewing the QoS metric as the cost we must pay to move to the destination, the 1-hop progress/cost ratio can be considered as the progress toward the destination in unit cost. Naturally, neighbors with greater progress/cost ratio are preferred. In Figure 2, node v has three neighbors x , y , and z . Since $p_d(v, x, d) < p_d(v, y, d) < p_d(v, z, d)$, z is most preferred in selecting successor for v . In Algorithm 1, BEST $(v, N_1(v), d)$ simply returns the node x in $N_1(v)$ with the maximum progress-over-cost $P_d(v, x, d)$.

4.3 Backward checking

Backward checking is a technique to find paths with better QoS properties. Consider a QoS routing request $Q(u, v, 11)$ in Figure 3 (a). Suppose u selects v as the only successor. The routing fails, although there exists a path (u, w, v) that satisfies the QoS constraint. It fails because node u knows only $d(u, v)$ and $d(u, w)$ but not $d(w, v)$ when it makes the decision. However, the failure could be avoided by doing a simple check when the ticket arrives at v , which has knowledge of $d(w, v)$. We propose the 1-hop backward checking technique as follows. Each ticket memorizes neighbor set $N(u)$ of the last traversed node u . After a ticket arrives at a successor v , v checks if there is a node $w \in N(u) \cap (N(v))$, such that $u \rightarrow w \rightarrow v$ has less QoS measure than path $u \rightarrow v$. If such a node w is found, it adjusts the traversed path of the ticket by replacing $u \rightarrow v$ with $u \rightarrow w \rightarrow v$. If multiple nodes exist, select one with the best QoS metric. In order to increase the probability of finding a path with better QoS property, 2-hop backward checking can also be used. That is, each ticket contains neighbor sets $N(u')$ and $N(u)$ of the last two traversed nodes u and u' , and the successor v also checks a better path (u', w, v) that has a better QoS property than the traversed path (u', u, v) .

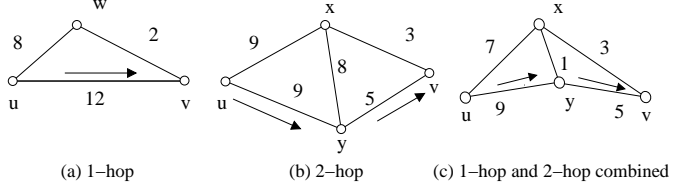


Figure 3. Backward checking.

Figure 3 (a) illustrates the 1-hop backward checking. When ticket t arrives at v , v finds that path (u, w, v) has a smaller QoS measure, and adjusts t 's traversed path by replacing (u, v) by (u, w, v) . Figure 3 (b) illustrates the 2-hop backward checking. After ticket t travels from u to y to v , B finds that path (u, x, v) is better than path (u, y, v) , and adjusts t 's traversed path by replacing (u, y, v) by (u, x, v) . Figure 3 (c) gives a more complicated example. u first selects y as the only successor and sends it ticket t . When t arrives at y , y changes the path in t from (u, y) to (u, x, y) by applying 1-hop backward checking. Then y forwards t to its successor v . When t arrives at v carrying path (u, x, y, v) , v changes path to (u, x, v) by applying 2-hops backward checking.

Backward checking can effectively make up for the weaknesses of localized routing. As the successors are selected based on the knowledge of 1-hop neighborhood, the decision may be "shortsighted". Backward checking examines multiple paths while traversing only one path, and makes better decisions based on a broader view beyond its 1-hop neighborhood. Backward checking can be also applied in other localized routing protocols such as GEDIR and MFR.

4.4 Two-ticket LTBR

Two-Ticket LTBR (LTBR-2) lies between flooding and single path routing, where at most two successors are allowed to be selected at each step. LTBR-2 uses an extra ticket at each step, thus the routing process could explore more paths, and both the success ratio and QoS properties could be enhanced. It has been observed that the effect of using a second ticket is significant in sparse networks. The ticket assignment algorithm of LTBR-2 (Algorithm 2) selects at most two successors, and has two implementation parameters α and β that control the directions of the ticket dissemination.

The first parameter α controls the $\angle xvd$, the angle between the destination d and the successor x , with respect to the current node v . A neighbor x satisfying $\angle xvd \leq \alpha$ is called a *preferred neighbor*. Usually, only preferred neighbors are considered in ticket assignment. Non-preferred neighbors are considered only when the preferred neighbors

Algorithm 2 ASSIGN-TWO-TICKETS (u, v, d, α, β)

```

1: if  $d \in N(v)$  then
2:   return  $\{d\}$ 
3: else
4:    $N_1(v) \leftarrow \{i \mid i \in N(v) \wedge \angle ivd \leq \alpha\} - \{u\}$ 
5:   if  $N_1(v) = \emptyset$  then
6:      $x_1 \leftarrow \text{BEST}(v, N_1(v), d)$ 
7:      $N_2(v) \leftarrow \{j \mid j \in N_2(v) \wedge \angle x_1vj > \beta\}$ 
8:     if  $N_2(v) \neq \phi$  then
9:        $x_2 \leftarrow \text{BEST}(v, N_2(v), d)$ 
10:      return  $\{x_1, x_2\}$ 
11:    else
12:      return  $\{x_1\}$ 
13:    else {if the preferred neighbor set is empty}
14:       $N_1(v) \leftarrow N(v) - \{u\}$ 
15:      if  $N_1(v) \neq \emptyset$  then
16:         $x_1 \leftarrow \text{BEST}(v, N_1(v), d)$ 
17:        return  $\{x_1\}$ 
18:      else
19:        return  $\emptyset$ 

```

are unavailable. For example, the three gray nodes in Figure 4 (a2) are node v 's preferred neighbors. Using such a parameter is based on one assumption. That is, a good path (the one with good QoS measure) from node s to d is not far away from line sd . Such a path has smaller hop counts than a path which has traversed far from line sd , and usually has smaller end-to-end QoS measure. Therefore, by limiting the exploring scope to a small area around line sd , we can find good paths with relatively low overhead. It is also observed that, if at each step the successor x satisfies $\angle xvd \leq \alpha$, the whole exploring scope of the routing process will be limited to a small area around sd , as shown in Figure 4 (a1).

The second parameter β controls the $\angle x_1vx_2$, the angle between the first successor x_1 and the second successor x_2 . When selecting x_2 , we expect $\angle x_1vx_2$ to be greater than a proper threshold β , so that the exploring scope of the two tickets will not share a large common area. Thus, their combined exploring scope is large and the possibility of finding a good path is high. Theoretically, a large α allows a node to send tickets to neighbors in a wide range, and a large β allows a node to split the tickets into two different directions thus a large scope of the network area can be explored. Therefore, large α and β increase the success ratio, as well as produce high routing overhead. In dense networks, relatively small α and β should be used to minimize overhead. In sparse networks, large α and β are necessary for a high success ratio.

The ticket assignment rules of LTBR-2 are as follows: (1) If the destination d is a neighbor of the current node v , d is the only successor. (2) If v has no neighbor except the predecessor u , no successor will be selected. (3) If v 's preferred neighbor set $N_1(v)$ is not empty, v picks the preferred neighbor x_1 with the best metrics as the first successor. If v

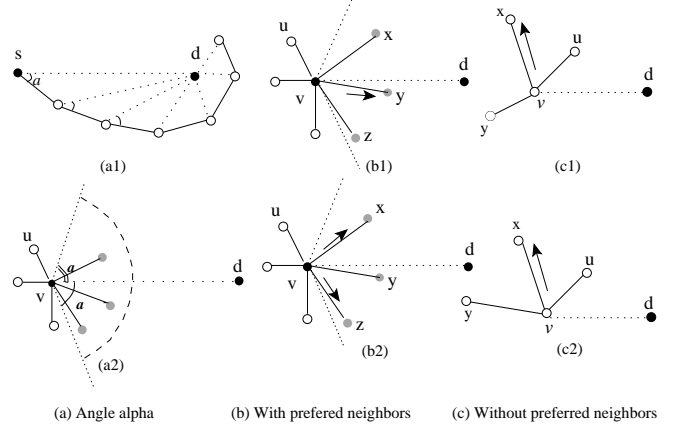


Figure 4. Ticket assignment in LTBR-2.

has other preferred neighbors satisfying $\angle x_2vx_1 > \beta$, v selects the best x_2 as the second successor. (4) If v 's preferred neighbor set is empty, the best non-preferred neighbor is selected as the only successor.

Figure 4 gives examples for this algorithm, where node d, v, u are respectively the destination, the current node and the predecessor. Figure 4 (b) shows the scenarios when the preferred neighbor set is not empty and $\beta = 60^\circ$. In Figure 4 (b1), when y is selected as the first successor, no other successor is selected since $\angle xvy$ and $\angle zvy$ are less than β . In Figure 4 (b2), when x is selected first, z is also selected since $\angle zvx$ is greater than β . Figure 4 (c) shows the scenarios when the preferred neighbor set is empty and $\beta = 60^\circ$. In this case, only one successor (x) will be selected, whether the angle between two non-preferred neighbors is larger than β , (as in Figures 4 (c1)) or not (as in Figures 4 (c2)).

5 Simulation

We have developed a simulator that implements various routing protocols, including Flooding (FLD), LTBR-1 (LTB1), LTBR-2 (LTB2), GEDIR (GDR), MFR, LAR1, and LAR2. We conduct the simulations in static ad hoc networks. Each network is randomly generated by deploying 500 nodes in a 60×60 area following the uniform distribution. All nodes in the network have the same transmission range r . Each wireless link is assigned a random QoS metric within $(3, 21)$. All protocols are simulated with the same set of QoS routing requests. To eliminate the disturbance on evaluation from the definitions of QoS constraints (i.e., if the QoS constraints are set too tight, all routing protocols may fail, and vice versa), we let all route requests have unlimited QoS constraints.

Among implemented protocols, Flooding uses no loca-

tion or QoS information to guide the dissemination of tickets. LTBR-1 and LTBR-2 use progress-over-cost in selecting successors to forward tickets. GEDIR, MFR, LAR1 and LAR2 use location information only. The following performance measures are compared: (1) *SuccRa*: $\frac{100n}{n_0}$, where n is the number of successful routings in a certain protocol, and n_0 is the success number in Flooding. (2) *QoSMea*: the average end-to-end QoS measure of the found paths. (3) *MsgSnt*: the average number of messages sent per routing process. (4) *MsgRec*: the average number of messages received per routing process. (5) *AvgHop*: the average hop number of found paths. The simulations are repeated until 95% confidence intervals of all average values are within $\pm 5\%$.

LTBR vs. Flooding. We first compare LTBR with several existing protocols. As shown in the following table, compared with Flooding, LTBR-1 achieves high success ratio with very low overhead (in terms of *MsgSnt* and *MsgRec*) in dense networks ($r = 6$). In sparse networks ($r = 4$), LTBR-2 has a high success ratio, while its overhead is still one magnitude lower than Flooding. Flooding can always find the optimal path, but the quality of the found paths (in terms of *QoSMea* and *AvgHop*) are very close for all three protocols. The only drawback of LBTR-1 is that it has a low success ratio in sparse networks. Since LBTR-1 has the lowest overhead, we suggest using LBTR-1 in dense networks and LBTR-2 in sparse networks. The selection of α and β in LTBR-2 are different in different networks, as will be discussed later in this section.

	$r = 6$			$r = 4$		
	FLD	LTB1	LTB2	FLD	LTB1	LTB2
SuccRa	100	98	99	100	38	92
QoSMea	51	63	55	93	104	97
MsgSnt	499	9	56	449	11	144
MsgRec	6142	9	105	2169	11	253
AvgHop	9	9	9	11	11	11

LTBR-1 vs. GEDIR and MFR. The following table compares the performance of LTBR-1 with two single path protocols GEDIR and MFR. All three protocols have similar success ratio and overhead. On the other hand, paths found by LTBR-1 have better (lower) QoS measures than the other two protocols. Obviously, using the QoS-aware metrics and backward checking in LTBR-1 increases the chances of finding “high quality” paths.

	$r = 6$			$r = 4$		
	GDR	MFR	LTB1	GDR	MFR	LTB
SuccRa	98	98	98	38	40	38
QoSMea	98	98	63	131	130	105
MsgSnt	7	7	9	10	10	11
MsgRec	7	7	9	10	10	11
AvgHops	7	7	9	10	10	11

LTBR-2 vs. LAR. We compare LTBR-2 with two location-based protocols using limited flooding: LAR1 and LAR2.

As shown in the following table, LTBR-2 has lower overhead than LAR in dense networks, and higher success ratio in sparse networks. The quality of the found paths are close for different protocols. In our simulation, only limited flooding is used in both LAR1 and LAR2; that is, the failure of finding a valid route will not trigger a global flooding.

	$r = 6$			$r = 4$		
	LTB2	LAR1	LAR2	LTB2	LAR1	LAR2
SuccRa	99	75	99	84	24	55
QoSMea	55	58	53	125	128	125
MsgSnt	59	84	175	97	75	62
MSgRec	112	918	1142	172	348	187
AvgHop	9	9	9	14	14	14

α and β in LTBR-2. Now we check the effects of several parameters that affect the performance of LTBR. The following tables show the effect of two parameters α and β in LTBR-2 in (1) dense networks ($r = 6$)

α	60	90	90	135	135	135	180	180	180
β	30	30	60	30	60	90	30	60	90
SuccRa	99	99	99	100	100	100	100	100	100
QoSMea	60	58	59	58	59	60	58	59	61
MsgSnd	51	67	83	70	98	155	70	100	177
MsgRec	94	126	157	133	186	289	132	188	327
AvgHop	9	10	10	10	10	10	10	10	10

and (2) sparse networks ($r = 4$)

α	60	90	90	135	135	135	180	180	180
β	30	30	60	30	60	90	30	60	90
SuccRa	46	63	64	78	82	84	81	89	93
QoSMea	125	121	121	120	121	121	121	121	123
MsgSnd	30	45	47	63	76	87	75	112	204
MsgRec	47	77	82	113	136	154	137	200	357
AvgHop	13	14	14	14	14	14	14	14	14

Simulation results show that, in dense networks, LTBR-2 achieves desirable success ratio and QoS properties of discovered paths for all listed parameter combinations. In sparse networks, there is not an ideal combination of α and β . The selection must be made based on specific requirements. $(180^\circ, 90^\circ)$ is the best choice for seeking best success ratio. When the message overhead is also a concern, $(135^\circ, 60^\circ)$, $(135^\circ, 90^\circ)$ or $(180^\circ, 60^\circ)$ are better choices. Through this section, we set (α, β) to $(75^\circ, 30^\circ)$ in networks with $r \geq 6$, $(135^\circ, 30^\circ)$ when $r = 5$, and $(180^\circ, 90^\circ)$ when $r \leq 4$.

Backward checking. The following table compares the average QoS measures of the found paths using different routing protocols with backward checking (C) or without (N). Column R is the reduced ratio $((N-C)/N)$.

	$r = 6$			$r = 5$			$r = 4$		
	N	C	R	N	C	R	N	C	R
GDR	98	71	0.28	129	100	0.23	130	107	0.18
MFR	98	71	0.28	128	100	0.22	129	107	0.17
LTB1	71	63	0.11	105	94	0.11	116	104	0.10
LTB2	59	55	0.07	96	90	0.06	132	125	0.05

Simulation results show that, backward checking does improve QoS properties of found paths. Note that, the improvement is higher in networks with higher densities. The improvement in non-QoS-aware routing protocols, such as GEDIR and MFR, is greater than that in LTBR. The improvement for LTBR-1 is also greater than LTBR-2. This is because the paths found by LTBR already have quite good QoS properties. In LTBR-2, the routing process has already searched multiple paths, leaving less room for improvement.

All enhancements. At last we evaluate the benefit of three proposed enhancements: progress-over-cost, backward checking and the ticket-based parallel routing scheme. The following table shows reduced ratios of the average QoS measure when these enhancements are applied to GEDIR. Columns (PoC) and (BkChk) are the results when progress-over-cost and backward checking are separately applied. Column (Both) described the results when both of them are applied (equivalent to LTBR-1). Column (ALL) describes the results when all the three enhancements are applied (equivalent to LTBR-2).

r	PoC	BkChk	Both	All
7	0.32	0.31	0.40	0.48
6	0.28	0.28	0.36	0.44
5	0.15	0.23	0.28	0.36
4	0.11	0.18	0.20	0.26
3	0.06	0.12	0.13	0.14

Simulation results show that the quality of found paths are improved by using progress-over-cost, backward checking, or parallel routing. The effect is more significant in dense networks. In addition, those three enhancements can be combined together to achieve the best result.

6 Conclusion

In this paper, we have proposed the generic mechanism of location-aided, ticket-based QoS routing (LTBR) and its two special cases, LTBR-1 and LTBR-2. Our approach differs from the traditional location-based routing protocols in that two important enhancements to localized QoS routing, the location-and-QoS-aware metric and backward checking, are introduced, and a parallel routing scheme is implemented. Our approach differs from the original ticket-based probing scheme in its on-demand feature. The tickets are dynamically generated based on timely position and QoS information; no routing table maintenance is needed. Simulation results show that, with the two enhancements, LTBR-1 beats existing localized location-based routing protocols in finding a path with better QoS properties. LTBR-2 achieves performance close to flooding with lower routing overhead in networks with any density. Moreover, in LTBR-2, by adjusting parameter α and β , tradeoff is allowed between the success ratio and the routing overhead.

Currently, our focus is the performance of LTBR in static networks, and routing with single additive QoS metric. It would be interesting to find out if LTBR still maintains its high success ratio in mobile ad hoc networks, where the destination location is inaccurate due to node movement. Our future work includes the performance evaluation of LTBR in dynamic networks and location-aided routing with multiple QoS constraints. In addition, an adaptive routing scheme will be explored, where the number of tickets issued at each node is adjusted dynamically to balance the success ratio and routing cost under different network density and mobility situations.

References

- [1] S. Basagmo, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (DREAM). *Proceedings of MOBICOM*, pages 76–84, 1998.
- [2] A. Boukerche and S. Rogers. GPS query optimization in mobile and wireless ad hoc networks. *Proceedings of IEEE Symposim on Computers and Communications*, pages 198–203, 2001.
- [3] S. Chen and K. Nahrstedt. Distributed quality-of-service routing in ad hoc networks. *IEEE Journal Selected Areas in Communication*, 17(8):1488–1505, Aug 1999.
- [4] D. B. Johnson, D. A. Maltz, Y. C. Hu, and J. Jetcheva. The dynamic source routing protocols for mobile ad hoc networks (DSR). *Internet Draft*, Feb. 2002.
- [5] Y. B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *Proceedings of MOBICOM*, pages 66–75, 1998.
- [6] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. *Proceedings 11th Canadian Conf. Computational Geometry*, Aug 1999.
- [7] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. *Proceedings of MOBICOM*, pages 120–130, March 2000.
- [8] I. Stojmenovic. Position-based routing in ad hoc networks. *IEEE Communications Magazine*, 39(2):142–148, Feb. 2001.
- [9] I. Stojmenovic and X. Lin. GEDIR: Loop-free location based routing in wireless networks. *Proceedings IASTED Int'l Conf. Parallel and Distriuted Computing and Systems*, pages 1025–1028, Nov. 1999.
- [10] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.
- [11] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 14(7):1228–1234, Sep. 1996.
- [12] Y. Xue, B. Li, and K. Nahrstedt. A scalable location management scheme in mobile ad-hoc networks. *Proceedings LCN 2001. 26th Annual IEEE Conference on Local Computer Networks, 2001*, pages 102–111, 2001.