Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks

Ivan Stojmenovic[1], Mark Russell[2] and Bosko Vukojevic[3]
[1]SITE, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada ivan@site.uottawa.ca, and DISCA, IIMAS, UNAM, Ciudad Universitaria, Mexico D.F. 04510, Mexico
[2]Dept. of Combinatorics & Optimization, University of Waterloo, Waterloo, Ont N2L 3G1, Canada
[3]SS8 Networks, Ottawa, Ontario, Canada

Abstract

In a localized routing algorithm, node A currently holding the message forwards it based on the location of itself, its neighboring nodes and destination. We propose to use depth first search (DFS) method for routing decisions. Each node A, upon receiving the message for the first time, sorts all its neighbors according to a criteria, such as their distance to destination, and uses that order in DFS algorithm. It is the first localized algorithm that guarantees delivery for (connected) wireless networks modeled by arbitrary graphs, including inaccurate location information. We then propose the first localized QoS routing algorithm for wireless networks. It performs DFS routing algorithm after edges with insufficient bandwidth or insufficient connection time are deleted from the graph, and attempts to minimize hop count. This is also the first paper to apply GPS in QoS routing decisions, and to consider the connection time (estimated lifetime of a link) as a QoS criterion. The average length of measured QoS path in our experiments, obtained by DFS method, was between 1 and 1.34 times longer than the length of QoS path obtained by shortest path algorithm. The overhead is considerably reduced by applying the concept of internal nodes.

Keywords: *Wireless networks, routing, quality of service.*

1. Introduction

Wireless networks consist of static or mobile hosts (or nodes) which can communicate with each other over the wireless links without any static network interaction. Each mobile host has the capability to communicate directly with another mobile hosts in its vicinity. They can also forward packets destined for other nodes. Examples of such networks are ad hoc local area networks, packet radio networks, and sensor networks. They are used in situations like disaster rescues, law enforcement, wireless conferences, battlefields, or monitoring objects in a remote environment.

The routing problem is the problem of finding a route for sending a message from a source to a destination. Routing becomes very difficult in wireless networks. In highly mobile situation, the flooding scheme is the most reliable for sending data packets. Since the link channel resource is very scarce and battery power is limited, more efficient schemes must be devised for moderate mobility rates. Surveys of routing protocols, which do not use GPS, are given in [BMJHJ, RT].

In a Quality-of-Service (QoS) routing problem, the selected path between source and destination should have sufficient resources to satisfy bandwidth or delay requirements in a dynamic multihop mobile environment. For example, such path is required if the image of an injured person in the field is routed to a medical specialist to obtain an advice on a first aid treatment. While delay and bandwidth are two major constraints in a wired environment, we believe that QoS in mobile wireless networks should consider the connection time on a path between source and destination as additional important requirement.

The desirable properties of any routing protocol include simplicity, loop-free operation, convergence after topological changes, storage overhead, computational and transmission overhead [MC]. The efficiency of a routing algorithm is measured by the delivery rate and the average hop count. The hop count is the number of transmissions needed by a routing method for a pair of source and destination. The delivery rate is the ratio of numbers of messages received by destination and sent by senders. For a shortest path algorithm, and for algorithms that guarantee delivery in a connected graph, this ratio is 1. Some of existing algorithms do not produce a unique path between a source and a destination (e.g. [BCSW, KV, S1]). Flooding rate is more appropriate measure for such algorithms. It is defined as the ratio of number of transmissions required by a method (in case of algorithms that produce a path, e.g. [SL1, BMSU] and this paper, it is equivalent to the hop count) and hop count of a shortest path algorithm.

Ad hoc networks are best modeled by the *minpower* graphs constructed in the following way. Each node A has its transmission range $t(A)$. Two nodes A and B in the network are neighbors (and thus joined by an edge) if the Euclidean distance between their coordinates in the network is less than the minimum between their transmission radii (i.e. $d(A,B) < \min \{t(A), t(B)\}$) [BCSW]. If all transmission ranges are equal, the corresponding graph is known as the *unit graph*. Method proposed in this paper is applicable to arbitrary graphs.

Ad hoc networks consist of autonomous nodes that run their routines in asynchronous fashion. The communication algorithms between nodes are therefore all distributed. However, [EGHK] defined the class of so called *localized* algorithms, as distributed algorithms where simple local node behavior achieves a desired global objective. Localized algorithms therefore resemble greedy sequential algorithms. They argued that localized algorithms may be necessary for sensor and ad hoc network coordination, and described localized algorithms for clustering and object location. In a localized routing algorithm, each node makes the decision to which neighbor(s) to forward the message based solely on the location of itself, its neighboring nodes, and destination. In addition, each node is allowed to perform local computation.

Non-localized algorithms, on the other hand, typically require the knowledge of the locations of all nodes in the network, and also the information about the existence of every edge in the graph. All non-localized routing algorithms proposed in literature are variations of shortest weighted path algorithm. Although ad hoc network is fairly accurately modeled by unit graphs, nodes that are at distance less than R may have an obstacle between them blocking the communication, while two nodes at distance that exceeds R by a small amount may still be able to communicate (or a node may even choose whether to use that possible but power demanding link). Any edge in the network that breaks or emerges (due to node mobility) may have impact on the whole network. In the worst case scenario, a path that used to be shortest may not even exist, and the algorithm does not offer immediately a reasonable alternative path unless the shortest (weighted) path algorithm is rerun. Therefore the maintenance of shortest weighted path requires that information about edges, in addition to location of nodes, is broadcast to the whole network, which is a significant quadratic communication overhead, and may cause significant delays in delivery. Next, some nodes in the sensor or ad hoc network may be temporarily inactive, and non-localized algorithms need to know which of the nodes are active to make their best decisions. The activity information puts additional demand on the information update. For example, static nodes may need to broadcast such information to the whole network whenever they change their activity status while, at the same time, they have no need to update their location with the rest of the nodes. In order to preserve battery power over long periods of time, nodes may change their activity on a regular basis. Thus, non-localized shortest path algorithms may not be the best choice for a routing algorithm even in case of static networks (e.g. some kinds of sensor networks). This paper deals solely with localized algorithms.

In accordance with [BCSW, BMSU, KV, KSU, SL1, SL2, TK], it is assumed that each node, in its routing table, contains the geographic location of all other nodes in network, including the time when the location for each node is established. The sender uses the latest available location about the position of the destination and attaches it to the message. Location updates and destination search problems are discussed in [BCSW, KV, S2, S1]. The proposed *DFS* algorithm guarantees delivery even if position of destination is not known accurately, and not consistent at various nodes.

Several localized routing algorithms are proposed recently in literature. One such method, *GEDIR* (geographic distance routing), based on the location information supplied by GPS is proposed in [SL1] and shown to have the best performance in its class. The hop count of *GEDIR* is very close to the hop count of the shortest path algorithm, and the success rate is very high for networks with high degrees. However, its success rate is low for low degree unit graphs. Full flooding approach guarantees delivery at the expense of unacceptable communication overhead. In addition, only one of recently proposed localized algorithms, *GFG* (and its variants) [BMSU] guarantees delivery, but only for wireless networks modeled by unit graphs. In addition, it requires accurate position of destination. Finally, *GFG* is a memoryless algorithm (that is, nodes do not need to memorize past traffic). The concept of internal nodes [WL, SSZ] is shown to improve the performance of routing algorithms, and will be applied also in this paper. All QoS routing algorithms for wireless networks proposed in literature are non-localized, and do not use GPS in their routing decisions.

This paper is organized as follows. Section 2 gives a literature review on GPS based localized routing in ad hoc networks, QoS routing in wireless networks, and the concept of dominating sets. *DFS* based routing algorithm is described in Section 3. Section 4 describes *DFS* based QoS routing algorithm, which is the first localized QoS routing algorithm, and the first one to apply GPS. Performance evaluation is given in section 4.

2. Literature review

Global Position System (GPS) provides location information (latitude, longitude and height) and time to hosts in ad hoc wireless network. GPS cards will be, in the near future, deployed in each car and possibly in every user terminal. For instance, Differential GPS offers accuracy of a few meters.

Stojmenovic and Lin [SL1] introduced the location based geographic distance routing *(GEDIR)* algorithm for a wireless network. When node *A* wants to send a message *m* to node *D*, it uses the location information for all its one hop neighbors and destination *D* to determine the neighbor *C* that is closest to *D* among all neighbors of *A*. The message is forwarded to *C*, and the same procedure is repeated until *D*, if possible, is eventually reached. In this basic *GEDIR* method, a node will stop forwarding the message if the best choice for it is the node from which the message arrived. *GEDIR* algorithm is inherently loop-free [SL1]. The proof is based on the observation that distances of nodes toward destination are decreasing. Other existing GPS based routing algorithms are reviewed and two of them compared experimentally with the *GEDIR* algorithm. One of methods, called *MFR* (most forward with progress) [TK] is comparable to *GEDIR* but is conceptually more sophisticated and requires more power [SL2], while the other, based on direction of edges [BCSW, KV, KSU], is not loop-free [SL1] and does not perform better in terms of success rates or hop counts [SL1]. More precisely, direction based methods [BCSW, KV] forward message from node *A* currently holding it to all nodes located between tangents from *A* to the circle that contains destination *D*. Ko and Vaidya [KV] proposed also *LAR2* scheme, in which source or each intermediate node *A* will forward the message to all nodes that are closer to the destination than *A* is. Experiments in [CL, SL1] indicate excessive flooding rates for these multipath methods, even if past traffic is memorized at each node.

Bose, Morin, Stojmenovic and Urrutia [BMSU] described a GPS based distributed routing algorithm which guarantees the delivery for wireless networks modeled by unit graphs, assuming only that the graph is connected. Important additional constraint is that the position of destination (as recorded by the sender node) is reasonably accurate (that is, the unit graph is not altered by imprecise information). *FACE* algorithm [BMSU] runs in two phases. In the first phase, a connected planar subgraph is constructed, using only localized criterion (an edge *AB* remains in the subgraph if $\angle ACB$ is acute for any common neighbor *C* of *A* and *B*). In the second phase, algorithm follows the faces of the planar subgraph in order of their intersection of line *SD* (where *S* is the source and *D* is the destination). Message may move from one face to the other whenever the edge it is sent on intersects the line *SD*. *GFG* algorithm combines *GEDIR* and *FACE* algorithm. *GEDIR* algorithm is applied as long as possible, until delivery or a failure. In case of failure, the algorithm switches to *FACE* algorithm until a node closer to destination than last failure node is found, at which point *GEDIR* is applied again.

Wang and Crowcroft [WC] considered a number of issues in QoS routing for various architectures that support multimedia applications such as digital video and audio. They examined the basic problem of QoS routing, namely, finding a path that satisfies multiple constraints, and its implications on routing metric selection, and presented three path computation algorithms for source routing and hop-by-hop routing. Although loss probability, cost, and delay jitter are very useful parameters, delay and bandwidth are the two most important metrics [WC]. The bandwidth considered is the residual bandwidth available for new traffic. The bandwidth of a path is defined as the minimum of the residual bandwidth of all links on the path or the *bottleneck bandwidth*. The delay has two basic components: queuing delay and propagation delay [WC]. Queuing delay is determined by bottleneck bandwidth and traffic characteristics. Since queuing delay is already reflected in the bandwidth metric, we only need to consider propagation delay in the delay metrics [WC]. This way, one can make sure that the two metrics are not interdependent. Bottleneck bandwidth and propagation delay can be viewed as the width and length of a path. The problem of QoS routing is then to find a path in the network given the constraints on its width and length.

We did not find any localized QoS routing algorithm for wireless networks in literature. Moreover, GPS was not previously used for distributed QoS routing in wireless networks. Toh [T] described associativity-based routing algorithm based on a new metric called degree of association stability. Each node periodically generates a beacon to signify its existence. For each beacon received, the associativity tick of the current node with respect to beaconing node is incremented. Association stability is defined by connection stability of one node with respect to another node over time and space. When multiple paths have the same overall degree of association stability, the route with minimum number of hops is selected. The query packets in [T] may contain QoS information. Special issue of IEEE J. on Selected Areas in Communications (Aug. 1999) featured four papers where QoS routing was discussed.

Let *G* be the graph that corresponds to given wireless network. A set is dominating if all the nodes in *G* are either in the set or neighbors of nodes in the set. Nodes that belong to a dominating set will be called *internal* nodes for *G* (of course, a different definition for dominating set leads to different set of internal nodes). Routing based on a connected dominating set is frequently used approach

[WL], where the searching space for a route is reduced to corresponding internal nodes. The routing process, in this approach, is divided into three steps. If source node is not an internal node, it forwards the packets to one of its adjacent internal nodes. This internal node then acts as a new source to route the packets in the reduced graph consisting of internal nodes only. Eventually, the packets reach the destination internal node which is either the destination node itself or neighbor of the destination node. In the later case, the destination internal node forwards the packets directly to the destination node. Such routing is suggested for shortest path and for dynamic source routing [WL], which do not use GPS.

Wu and Li [WL] proposed a simple and efficient algorithm for calculating connected dominating set in wireless networks. They introduced the concept of an *intermediate* node. A node A is an *intermediate* node if there exist two neighbors B and C of A that are not direct neighbors themselves. They introduced also two rules that considerably reduce the number of internal nodes in the network. Let $N(u)$ be the (open) set of all neighbors of node u, and let $N[u]=N(u) \cup \{u\}$ be the corresponding closed neighbor set. Suppose that each node has a unique *id* number. Let us define *inter-gateway* and *gateway* nodes as intermediate nodes that are not eliminated by Rule 1 and both rules, respectively.

Rule 1 [WL] is as follows. Consider two intermediate nodes v and u. If $N[v] \subseteq N[u]$ in G and $id(v) < id(u)$, then node v is not an inter-gateway node. In other words, if any neighbor of v is also a neighbor of u, and v is connected to u and has lower *id*, then any path via v can be replaced by a path via u, thus node v is not needed as internal node. We may also say that node v is 'covered' by node u. The number of internal nodes can be further reduced by applying Rule 2 [WL], as follows. Assume that, after applying Rule 1, u and w are two inter-gateway neighbors of a inter-gateway node v. If $N(v) \subseteq N(u) \cup N(w)$ in G and $id(v) = \min \{id(v), id(u), id(w)\}$, then node v is declared a non-gateway node. In other words, if each neighbor of v is a neighbor of u or w, where u and w are two connected neighbors of v, then v can be eliminated from the list of gateway nodes (when, in addition, v has lowest *id* among the three). The hop count between a source and destination node may increase by one in this process, since a segment *pvq* of the path between them is replaced by a segment *puwq*.

Since nodes know the location of all their neighbors, each node can determine whether or not it is an intermediate, inter-gateway or gateway node in $O(k^2)$ computation time (where k is the number of its neighbors), and without any message exchanged with its neighbors for that purpose.

Stojmenovic, Seddigh, and Zunic [SSZ] proposed to replace node *id*s with a record (*degree, x, y*), where *degree* is the number of neighbors of a node, and x and y are its two coordinates in the plane. In both rules from [WL], nodes compare first their degrees, and node with higher degree has greater chances of remaining an internal node. In case of ties, x-coordinate is used to resolve. If x-coordinates also happen to be the same, use y-coordinate for final decision. Such comparison rule will result in fewer remaining nodes in the graph. The information about the degree of neighboring nodes may be gathered together with their location.

The preliminary version of this paper was published as [SRV]. After the publication of [SRV], we have learned about a relevant and independent article [JPS]. Jain, Puri and Sengupta [JPS] proposed a single-path strategy that guarantees delivery called *geographic routing algorithm (GRA)*. It requires nodes to partially store routes toward certain destinations in routing tables. *GRA* applies greedy strategy in forwarding messages. However, sometimes node S may discover that it is closer to the destination D than any of its neighbors. That is, the packet may be 'stuck' at S. Under this condition, it starts the route discovery protocol. The route discovery finds a path from S to D and updates the routing tables toward D at any node on the path, with this information. After that the route discovery protocol is successfully completed, the stuck packet can be routed from S to D. The authors propose two route discovery strategies: *breadth first search* (which is equivalent to flooding) and *depth first search (DFS)*. *DFS* yields a single acyclic path from *S* to *D*. Each node puts its name and address on the route discovery packet *p*. Then it forwards *p* to a neighbor who has not seen *p* before. This neighbor is one of all the neighbors which minimize *d(S,y)+d(y,D)*, where *d(x,y)* is Euclidean distance between nodes *x* and *y*. If a node has no possibilities to forward the packet, it removes its name and address from the packet and returns the packet to the node from which it originally received it. Route discovery packets are kept for some time. If a node receives twice the same packet, it refuses it. The authors investigate routing table sizes and present methods for taking into account positional errors, node failures and mobility.

The concept of connection time (proposed in this article, with preliminary version in [SRV]) was independently discussed by Su, Lee and Gerla [SLG] for mobility prediction in wireless networks.

3. DFS based routing algorithms

The *DFS* based routing algorithm performs *DFS* search in a given graph in distributed way. The property that enables its use in routing is the fact that *DFS* creates a path in the graph without making any jumps from a node to another node that is not it's neighbor. In *DFS* based algorithm, nodes are 'colored' as white or gray (because of distributed behavior of our algorithm, the third color is not needed). Initially all nodes are white (that is, if message *id* is not found in local memory, white color is assumed). The process of visiting nodes coincides with sending messages between nodes. There is always only one copy of message in the graph, and thus a path is created. The sender node *S* begins routing and colors itself as a gray node. Gray nodes are nodes that are visited (that is, they received message at least once). Each message that is sent from a node *B* to a node *A* has one bit that indicates whether the message is forwarded or returned. Node *A* receiving the message then acts according to that bit.

White node *A*, upon receiving forwarded message for the first time, changes its color to gray, and orders its neighbors according to distance from destination (the neighbors which are closer to destination are preferred). The only exception is that node *B*, that sent message to *A*, is ignored. Thus node *A* should memorize, together with the message *id*, also neighbor *B* that forwarded that message. The message is then forwarded to the first choice *C* among neighbors. If there is no choice, message is returned to *B*.

Gray node *A*, upon receiving forwarded message from any node *B*, will reject the message immediately. That is, the message will be immediately returned to *B*. If node *A* sends message to node *B*, and node *B* rejects the message, it is counted as two hops in the simulation (*A* to *B* and *B* back to *A*). Gray node *A*, upon receiving a returned message from node *C*, will forward the message to the next choice *E* in its sorted list of neighbors, if such a neighbor exists. If *A* has no more neighbors in its list, message will be returned to the neighbor *B* which sent the message to *A* (and which was memorized for that purpose).

Fig. 1 gives an example of a *DFS* routing path ***SABC***_A_***C***_SCB_***A***_CA_***S***_C_***SEFGHID,*** where bold and regular letters correspond to forwarded and returned/rejected messages, respectively (sender node is marked).

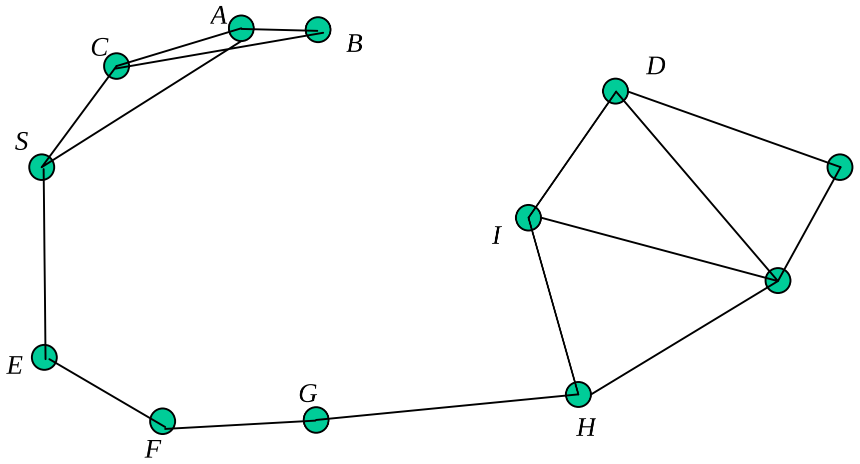

Figure 1. DFS routing path ***SABC**A**C**SCB**A**CA**S**C**SEFGHI**D* and QoS path *SEFGHID*

The pseudo code of *DFS* based routing algorithm can be given as follows:

```
Procedure DFS-forward(B, A, D);
    B forwards message to A;
    If A=D then message delivered else {
        If A is white node then
            {A is colored as gray node; A memorizes B;
             A sorts all neighbors (except B) according to
            distance from D as $C_1$, $C_2$, …, $C_k$;
        If $k>0$ then DFS-forward(A, $C_1$, D)
                else DFS-return(A, B, D)
                        else DFS-return(A,B,D); }
Procedure DFS-return(C,A,D);
    C returns message to A;
    A sorts all neighbors (except memorized neighbor B)
        according to distance from D as $C_1$, $C_2$, …, $C_k$;
    Let L be the index such that $C=C_L$;
    If $L < k$ then DFS-forward(A, $C_{L+1}$, D) else
        {If A=S then D not connected to S
                else DFS-return(A,B,D);}
Program DFS-routing(S,D);
    Let C be the neighbor of S which is closest to D;
    DFS-forward(S, C, D);
```

The basic *DFS* routing algorithms can be improved in several ways. The concept of internal nodes may be applied, resulting in reduced size of graphs (details of graph reduction size are given in [SSS]), and consequently fewer edges and nodes to be traversed by *DFS*. Nodes may also memorize, in addition to one neighbor (the one message came from for the first time) as required by *DFS* algorithm, any neighboring node that sent message which was then rejected. It turns out that in the current version of algorithm nodes may attempt each other as possible neighbors, which increases number of failures. However, this approach involves possibly non-constant amount of required memory at each node, and will not be pursued in our experiments.

4. DFS based QoS routing algorithms

In this section we shall discuss the extension of *DFS* based routing algorithm to provide QoS support. As already mentioned, only propagation delay is an independent delay requirement [WC]. In wireless networks, propagation delay is proportional to hop count between two nodes. Therefore we approximate delay QoS requirement with hop count, and will attempt to minimize hop count in our algorithm. We shall adopt the bottleneck bandwidth model [WC] in our algorithm. Bandwidth of an edge is determined by the number of available time slots for communication between the endpoints, taking current traffic (reserved time slots) into consideration. Edges which do not have sufficient bandwidth are simply deleted from the graph that models wireless network, and *DFS* routing is performed on so defined subgraph. Each node on the path memorizes the previous and next node, which suffices to use QoS path in both directions between source and destination. Nodes that return or reject the message during the path search are deleted from the QoS path.

While loss probability is low in wired networks, due to the connection stability, wireless networks pose additional challenge. Toh [T] addressed that challenge by considering the degree of association stability as QoS parameter. The availability of GPS may provide a more precise parameter, which we introduce here. The connection time is defined as the estimated duration of a connection between two neighboring nodes. Neighboring nodes frequently update their location to each other, and this information may be used to estimate the direction and speed of their movements. In turn, this suffices to estimate the connection time. Let A and B be the two neighboring nodes which move at speeds a and b, respectively. Here, A and B are position vectors while a and b are directional vectors. At time t, they move to new positions $A'= A+at$ and $B'= B+bt$. They will loose their connection when the distance between them becomes $>R$, where R is the radius of corresponding unit graph (or the smaller of their transmission radii in case of minpower graphs). The time t when the connection will be lost can be estimated by solving quadratic equation $|A'B'|=|B-A+(b-a)t|=R$.

The connection time may be treated as constraint in two ways. It can be used as hard constraint, hoping that a QoS path will be found that will be available for the whole duration of multimedia transmission. The bottleneck connection time is then the minimal connection time between two edges on a path. Clearly, this requirement can be handled in the same way as the bottleneck bandwidth, by eliminating edges whose estimated connection time is below the threshold. If such a stable path cannot be found, the required connection time for the whole transmission may be divided into few time fragments (whose duration depends on the mobility rate of the network nodes). A stable path is used in each time fragment, and, simultaneously, a new path is created for the next time fragment, using updated location information. This approach differs from the path reconstruction one suggested in recent literature, whose goal is to replace a particular broken link with an alternate path between the two nodes.

In summary, when a QoS path between a source S and destination D is required in a mobile wireless network, the following steps are performed:

- Destination search, by applying a method that depends on the location update scheme;
- Destination receives QoS requirements and position of source, and may initiate DFS routing algorithm (links that do not satisfy bandwidth and connection time requirement are ignored) which will construct DFS path;

- Source learns QoS path toward destination and may begin using it.

Note that *DFS* routing may be also applied in the first step, since it guarantees destination detection (in connected wireless networks) even if destination location is not accurately known. The method is successful, and its performance may be improved by applying suitable location update schemes.

5. Performance evaluation

We have designed experiments that evaluated the *DFS* routing algorithm independently from the location update or destination search schemes. This is achieved by assuming that the destination location is accurately recorded at each node (that is, destination search was successful). Although nodes may move, the node currently holding the message only needs to know the current locations of neighbors which are active. Thus, with no significant loss of output accuracy, we may assume that the network is static and all nodes are active.

The experiments were carried on random connected unit graphs, defined as follows [SL1, SL2]. Each of *n* nodes is chosen by selecting its *x* and *y* coordinates at random in the interval [*0,100*). In order to control the average node degree *d,* we sort all *n(n-1)/2* (potential) edges in the network by their length, in increasing order. The radius *R* that corresponds to chosen value of *d* is equal to the length of *nd/2*-th edge in the sorted order. Dijkstra's shortest path algorithm (from one source to all nodes) can be used to test whether a graph is connected.

The experiments measured average hop counts of *DFS* based routing algorithm (including rejected and returned messages), and compared with hop count of the shortest path (*SP*) algorithm. Graphs are classified according to the pair *(n,d).* Next, the same *DFS* based algorithm was used for routing and QoS routing, the difference being that some edges, not satisfying bandwidth and connection time criteria, are eliminated from graph. If the remaining subgraph is also parameterized by pair *(n,d)* (where *d* is the average degree of the remaining subgraph) then no specific QoS routing experiments are needed (the same tables may be used). The difference is that effective hop count (*EHC*) is measured (instead of hop count), which is the *DFS* path obtained by eliminating nodes that rejected or returned messages (more precisely, eliminating corresponding edges). That is, only the path to be used by QoS routing tasks remains. The *EHC* of a path can be calculated as *EHC= hop_count - 2* reject_count - 2* return count.* Table 1 gives the average hop counts (including reject and return messages) for *DFS* routing algorithm. Number of nodes *n* range from 20 to 100 while degree *d* is between 3 and 11.

(n,d)	3	4	5	6	7	8	9	10	11
20	5.17	4.36	3.05	2.43	2.03	1.80	1.64	1.53	1.45
30	11.69	8.32	6.15	4.03	3.12	2.58	2.17	1.99	1.91
40	13.84	12.48	8.79	6.33	4.13	3.40	3.16	2.43	2.22
50	21.33	16.93	13.00	8.23	6.83	4.38	3.38	2.98	2.72
60	22.95	22.14	14.60	10.75	6.39	5.47	4.44	3.25	2.99
70	27.83	27.81	21.61	14.88	9.78	5.83	4.73	3.82	3.18
80		31.46	24.05	15.69	11.34	7.20	5.45	4.36	3.76
90		39.24	28.98	19.23	12.44	8.01	5.76	4.47	3.93
100		43.57	32.87	21.05	13.22	8.43	6.89	5.05	4.65

Table 1. Average hop counts for connected unit graphs with *n* nodes and degree *d*

(n,d)	3	4	5	6	7	8	9	10	11
20	1.36	1.46	1.22	1.12	1.05	1.03	1.01	1.00	1.00
30	2.25	2.10	1.85	1.41	1.25	1.14	1.04	1.01	1.04
40	2.25	2.54	2.20	1.83	1.36	1.25	1.27	1.05	1.02
50	3.20	2.99	2.77	2.07	1.97	1.41	1.19	1.13	1.10
60	2.88	3.42	2.79	2.41	1.64	1.57	1.41	1.11	1.09
70	3.13	3.81	3.73	3.09	2.29	1.56	1.37	1.19	1.06
80		3.93	3.81	2.98	2.42	1.77	1.47	1.28	1.18
90		4.50	4.28	3.38	2.56	1.82	1.45	1.22	1.15
100		4.71	4.54	3.48	2.56	1.83	1.65	1.31	1.28

Table 2. Ratios of average hop counts of *DFS* and *SP* algorithms

Table 2 gives the ratios of these hop counts and hop counts of *SP* algorithm (i.e. flooding rates). These ratios are close to optimal (1) for higher degree graphs, and reasonable for low degree graphs. The largest ratio found was 4.71 for *n=100, d=4*. Table 3 gives the *EHCs* for *DFS* based QoS routing. Table 4 presents ratios of *EHCs* and hop counts by *SP* algorithm. The largest ratio found is 1.34, meaning that *DFS* does provide almost shortest path routes for QoS routing in localized manner. This excellent ratio is obtained for the critical case of low degree graphs, while higher degree graphs gave almost perfect routes, with ratios very close to 1.

(n,d)	3	4	5	6	7	8	9	10	11
20	3.83	3.22	2.60	2.22	1.97	1.78	1.64	1.53	1.45
30	6.23	4.53	3.77	3.05	2.63	2.34	2.12	1.97	1.85
40	7.17	5.71	4.62	3.83	3.22	2.86	2.59	2.35	2.19
50	8.44	6.75	5.70	4.55	3.97	3.35	2.97	2.72	2.52
60	9.43	7.87	6.25	5.29	4.31	3.85	3.39	3.00	2.80
70	10.46	9.05	7.57	6.16	5.04	4.09	3.71	3.31	3.04
80		9.85	8.16	6.62	5.53	4.61	4.01	3.62	3.32
90		10.98	9.10	7.27	5.96	4.92	4.30	3.84	3.54
100		11.87	9.61	7.73	6.38	5.33	4.63	4.12	3.80

Table 3. Effective hop counts for *DFS* based QoS routing

(n,d)	3	4	5	6	7	8	9	10	11
20	1.01	1.08	1.04	1.02	1.02	1.02	1.01	1.00	1.00
30	1.20	1.14	1.13	1.07	1.05	1.03	1.01	1.01	1.01
40	1.17	1.16	1.16	1.11	1.06	1.05	1.04	1.02	1.01
50	1.26	1.19	1.21	1.15	1.14	1.08	1.05	1.03	1.02
60	1.18	1.21	1.20	1.19	1.11	1.11	1.08	1.03	1.02
70	1.18	1.24	1.31	1.28	1.18	1.10	1.07	1.04	1.02
80		1.23	1.29	1.26	1.18	1.13	1.08	1.06	1.04
90		1.26	1.34	1.28	1.23	1.12	1.08	1.05	1.03
100		1.28	1.33	1.28	1.23	1.15	1.11	1.07	1.04

Table 4. Ratios of effective hop counts and hop counts by *SP* algorithm

	3	4	5	6	7	8	9	10	11
All nodes	3.13	3.81	3.73	3.09	2.29	1.56	1.37	1.19	1.06
intermediate	2.42	2.74	3	2.54	2.12	1.59	1.28	1.15	1.06
inter-gateways	1.92	2.16	2	1.73	1.37	1.38	1.15	1.06	1.03
gateways	1.82	1.81	1.80	1.64	1.33	1.24	1.16	1.16	1.10

Table 5. Flooding rates for *n*=70 and each type of internal nodes

	3	4	5	6	7	8	9	10	11
All nodes	1.18	1.24	1.31	1.28	1.18	1.1	1.07	1.04	1.02
intermediate	1.14	1.15	1.24	1.2	1.17	1.11	1.06	1.03	1.02
inter-gateways	1.1	1.13	1.14	1.14	1.08	1.1	1.04	1.02	1.01
gateways	1.09	1.11	1.17	1.18	1.12	1.12	1.1	1.1	1.07

Table 6. Ratios of *EHC*s and hop counts for *SP* for *n*=70 and each type of internal nodes

The performance of *DFS* routing algorithm was also measured on each of three types of internal nodes. The reduction in the average hop counts was notable, especially for lower degrees. Table 5 gives the ratios of average hop counts of *DFS* routing on these types of internal nodes and hop counts of *SP* method (that is, flooding rates), for *n*=70 and graph degrees from 3 to 11. Note that the hop count of *SP* algorithm for intermediate and inter-gateway nodes is the same as hop count for the set of all nodes, while the hop counts on gateway nodes somewhat increases. Flooding rate for gateway nodes is measured with respect to hop count of *SP* algorithm applied on all nodes (not only gateway nodes). The improvement after introducing intermediate nodes concept (the simplest one to apply) is notable, especially for low degree graphs. Inter-gateway nodes show further improvements. On the other hand, gateway nodes showed some improvements for low degrees graphs, but not for higher degree graphs. Thus their application, according to our results, is not justified.

Table 6 presents the ratios of effective hop counts and hop counts of SP algorithm for n=70 nodes and degrees 3 - 11, for each kind of internal nodes.

6. Conclusions

Our experiments have shown a very competitive results for *DFS* based routing, in terms of the average hop counts. Moreover, the length of produced QoS path compare very favorably with the length of QoS path found by applying shortest path algorithm. Depending on graph degree, the average length of QoS path was between 1 and 1.34 times longer than the one obtained from shortest path algorithm. Since *DFS* based routing algorithm is localized, while shortest path algorithm is non-localized, our experimental results have suggested *DFS* based routing as a practical method for QoS routing in wireless networks. The overhead in finding QoS path, which comes from rejected and returned messages in *DFS*, are considerably reduced by applying the concept of internal nodes. The values of this flooding rate (as measure of overhead) were always below 5 (and below 3 for intergateway and gateway nodes), thus confirming *DFS* based routing as a competitive routing method.

The hop count in DFS based routing scheme can be improved further in the following way. The current algorithm considers each neighbor *B* at each node *A*. However, if *B* has already received the message before, it will reject it. The current hop count includes this communication as two messages: forward from *A* to *B*, and reject from *B* to *A*. The proposed modification is to eliminate such communication as follows. If *B* has sent already at least once the same message to any neighbor, then that message was heard by all its neighbors, not only intended one. Therefore *A* will learn that *B* already received the message, and will eliminate *B* from its list of candidate neighboring nodes. *A* will therefore immediately try the next best choice.

The hop count can be further improved by applying Gabriel graph or *RNG* concepts [BMSU, SSS] instead or in conjunction with the dominating sets applied in this paper.

Further experiments will include the performance of *DFS* based localized routing algorithms which use power, cost and power-cost metrics instead of constant transmission power between two nodes. These metrics and corresponding localized routing algorithms (whose major problem is that delivery is not guaranteed) are described in [SL2]. The distance of neighboring nodes may be replaced by progress, direction, estimated transmission power, or quality of service (delay, bandwidth) criteria in routing decisions, as further applications of *DFS* based localized routing algorithm with guaranteed success. Finally, selected location update schemes will be combined with mentioned *DFS* based routing algorithm and simulated on moving nodes.

References

[BCSW] S. Basagni, I. Chlamtac, V.R. Syrotiuk, B.A. Woodward, A distance routing effect algorithm for mobility (DREAM), Proc. MOBICOM, 1998, 76-84.

[BMSU] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, Routing with guaranteed delivery in wireless networks, 3rd Int. Work. on Discr. Alg. and Methods for Mobile Computing and Comm., Seattle, 1999, 48-55; **ACM/Kluwer Wireless Networks**, 7, 6, November 2001, 609-616.

[BMJHJ] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, Proc. MOBICOM, 1998, 85-97.

[CL] D. Camara and A.F. Loureiro, A novel routing algorithm for ad hoc networks, Proc. HICSS, Hawaii, January 2000.

[EGHK] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next century challenges: Scalable coordination in sensor networks, MOBICOM 1999, Seattle, 263-270.

[JPS] R. Jain, A. Puri and R. Sengupta, Geographical routing using partial information for wireless ad hoc networks, IEEE Personal Communication, February 2001, 48-57.

[KSU] E. Kranakis, H. Singh and J. Urrutia, Compass routing on geometric networks, Proc. 11th Canadian Conf. on Comput. Geometry, Vancouver, Aug. 1999.

[KV] Y.B. Ko and N.H. Vaidya, Location-aided routing in mobile ad hoc networks, MOBICOM, 1998, 66-75.

[MC] J.P. Macker and M.S. Corson, Mobile ad hoc networking and the IETF, Mobile Computing and Communications Review, 2, 1, 1998, 9-14.

[RT] E.M. Royer and C.K. Toh, A review of current routing protocols for ad hoc mobile wireless networks, IEEE Personal Communications, April 1999, 46-55.

[S1] I. Stojmenovic, Voronoi diagram and convex hull based geocasting and routing in ad hoc wireless networks, Computer Science, SITE, University of Ottawa, TR-99-11, December 1999; www.site.uottawa.ca/~ivan.

[S2] I. Stojmenovic, Location updates for efficient routing in wireless networks, in: **Handbook of Wireless Networks and Mobile Computing**, John Wiley & Sons, 2002, 451-471.

[SLG] W. Su, S.J. Lee, M. Gerla, Mobility prediction in wireless networks, Proc. IEEE MILCOM, October 2000.

[SL1] I. Stojmenovic and Xu Lin, Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks, **IEEE Transactions on Parallel and Distributed Systems**, 12, 10, October 2001, 1023-1032.

[SL2] I. Stojmenovic and Xu Lin, Power-aware localized routing in wireless networks, **IEEE Transactions on Parallel and Distributed Systems**, Vol. 12, No. 11, November 2001, 1122-1133.

[SRV] I. Stojmenovic, M. Russell, and B. Vukojevic, Depth first search and location based localized routing and QoS routing in wireless networks, **IEEE International Conference on Parallel Processing**, August 21-24, 2000, Toronto, 173-180; SITE, University of Ottawa, TR-00-01, January 2000.

[SSS] M. Seddigh, J. Solano Gonzalez and I. Stojmenovic, RNG and internal node based broadcasting algorithms for wireless one-to-one networks, **ACM Mobile Computing and Communications Review**, 5, 2, April 2001, 37-44.

[SSZ] I. Stojmenovic, M. Seddigh and J. Zunic, Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks, **IEEE Transactions on Parallel and Distributed Systems**, Vol. 13, No. 1, January 2002, 14-25.

[T] C.K. Toh, Associativity-based routing for ad-hoc mobile networks, Wireless Personal Communications 4, 1997, 103-139.

[TK] H. Takagi and L. Kleinrock, Optimal transmission ranges for randomly distributed packet radio terminals, IEEE Transactions on Comm., 32, 3, 1984, 246-257.

[WC] Z. Wang and J. Crowcroft, Quality-of-service routing for supporting multimedia applications, IEEE J. on Selected Areas in Communications, 14, 7, 1996, 1228-1234.

[WL] J. Wu and H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, Proc. DIAL M, Seattle, Aug. 1999, 7-14.

This research is supported by NSERC and CONACYT research grants.