Location Based Localized Alternate, Disjoint, Multi-path and Component Routing Schemes for Wireless Networks

Xu Lin[1]
[1]Cognos Incorporated, 3755 Riverside Drive, P.O. Box 9707, St. 'T', Ottawa, ON Canada K1G 3Z4
Xu.Lin@cognos.com

Mouhsine Lakshdisi[2]
[2]SITE, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada,
mlakhdis@site.uottawa.ca

Ivan Stojmenovic[2,3]
[3]DISCA, IIMAS, UNAM, Dirrection Circuito Escolar s/n, Coyoacan, Mexico D.F. 04510, Mexico
ivan@site.uottawa.ca

ABSTRACT

In this paper we propose four schemes that improve the performance of greedy routing method. In the *alternate* method, the *i*-th received copy of message *m* is forwarded to *i*-th best neighbor, according to the selected criterion (it fails if number of copies exceeds number of neighbors). In the *disjoint* method, each intermediate node, upon receiving *m*, will forward it to its best neighbor among those who never received the message (it fails if no such neighbor exists). In the *multi-path* method, the source node *S* forwards *m* to *c* best neighbors according to distance from *D*. Each of *c* created copies afterwards follows the original, alternate, or disjoint method (these copies may interact since copy numbers are not communicated). *Component* routing method follows original greedy method until a failure node *F*. Such node *F* forwards the message to one node (using distance criteria) in each connected component of its neighbors, and then withdraws from the network for that message *m* (that is, neighboring nodes will ignore *F* when forwarding further copies of *m*). Thus *F* creates *c* copies of the message, where *c* is the number of connected components in the subgraph of its neighbors. All proposed methods are loop-free, have improved delivery rate over greedy method and reduced flooding rates compared to other existing methods. Component routing method guarantees delivery of *m* in connected graphs (even if the location of *D* is inaccurate).

Keywords

Routing, wireless networks, localized algorithms, location.

1. INTRODUCTION

In this paper, we consider the routing task, in which a message *m* is to be sent from a source node to a destination node in a given wireless ad hoc (a recent survey of mobile ad hoc networks is given in [4]), sensor, or rooftop network. In a localized routing algorithm, each intermediate node makes forwarding decision based solely on the location of itself, its neighbors, and destination. Our goal in this paper is to improve the performance of greedy localized routing algorithms in terms of delivery and flooding rates (the flooding rate is the ratio of total number of message transmissions in a given method and the ideal shortest path method). In existing greedy routing method [3], when node *A* wants to send message *m* to destination node *D*, it forwards *m* to it's neighbor *C* whose geographic distance is closest to *D* among all neighbors of *A*. The same procedure is repeated until *D*, if possible, is eventually reached. The *original* [7] greedy method fails if the best neighbor is the one that message came from. The scheme does not perform well for sparse graphs and does not guarantee delivery. The full version of this paper is available at www.site.uottawa.ca/~ivan and contains literature review and performance evaluation (more detailed reviews are available in two survey papers, one on routing and other on location updates, also available at the same web page, and in recent review [6]). Because of space limitations, this paper contains only descriptions of newly proposed methods and the proof that component routing method guarantees delivery.

MobiHOC 2001, Long Beach, CA, USA
© ACM 2001 1-58113-390-1/01/10...$5.00

2. ALTERNATE, DISJOINT AND MULTIPATH ROUTING METHODS

In the *alternate GEDIR* algorithm, each intermediate node forwards *i*-th received copy of the same message to the *i*-th best (closest to destination) neighbor. Thus concave nodes of the original *GEDIR* method do not stop transmitting; however a node becomes concave if it has fewer neighbors than the number of received copies. Temporary loops may be created, but the exit is guaranteed with maximum node degree.

In the *disjoint GEDIR* method, each intermediate node *A*, upon receiving the message, will forward it to its best neighbor among those who never received and forwarded the same message before. After forwarding the message, node *A* becomes inactive with respect to that message, and rejects further copies of it. All neighbors of *A* receive message forwarded by *A* and may eliminate *A* from their list of candidate neighboring nodes for forwarding the same message. A node stops forwarding the message (e.g. becomes concave) if there is no neighbors left to forward the message. Loops cannot be created by the algorithm design.

Figure 1 illustrates why disjoint method is more successful than alternate one. This conclusion is also confirmed by experiments. In the alternate method, node *A* sends message to best neighbor node *B*, *B* back to *A*, *A* to the second best neighbor *C*, *C* to the best neighbor *B*, *B* to the second best neighbor *C*, and *C* to the second best neighbor *A* which is concave node for the method since it has no more neighbors. In the disjoint method, *A* sends to the best neighbor *B* and rejects further messages, *B* sends to the best remaining neighbor *C*, *C* sends to *E* since *A* and *B* are not available for the message, etc. until *D*.

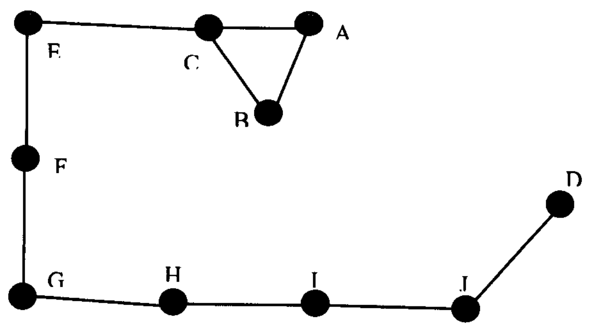

Figure 1. Alternate *ABACBCA* and disjoint *ABCEFGHIJD* paths from *A* to *D*

In the *multi-path* method, the source node forwards the message to *c* best neighbors, according to selection criterion. In the *original c-GEDIR* method, each node forwards the first received copy to the neighbor that is closest to destination (or stops forwarding if the node is concave) and does not forward other received copies. In the *alternate c-GEDIR* method, intermediate nodes on each path follow the described behavior whenever they receive any copy of the message (irrelevant from which of *c* initiated paths the message originates). Thus *i*-th received copy of the same packet is forwarded to *i*-th closest to destination neighbor, up to the number of neighbors. Nodes in the *disjoint c-GEDIR* algorithm do not receive the same packet twice; thus the scheme merely attempts to create *c* disjoint paths between source and destination nodes. The proposed multi-path methods provide robustness without much additional flooding. In all three methods, if destination *D* is one of neighbors of the current node *C*, the message is delivered to *D*, which might be an exception to the corresponding behavior at *C*.

Consider an example in Fig.2, where message is to be sent from *S* to *D* using $c=2$ paths. The radius of the unit graph is shown below the graph. The original methods produce the first successful path *SABCD*, while the second path *SFA* stops at *A*. Both paths are shown with bold lines. Two alternate paths are *SABCD* and *SFAIJCD*. New edges in alternate path are shown with dashed lines. Finally, the disjoint method produces the same first path *SABCD* while the second path, which fails, is *SFHIJ* (new edges on the path are shown with dotted lines).

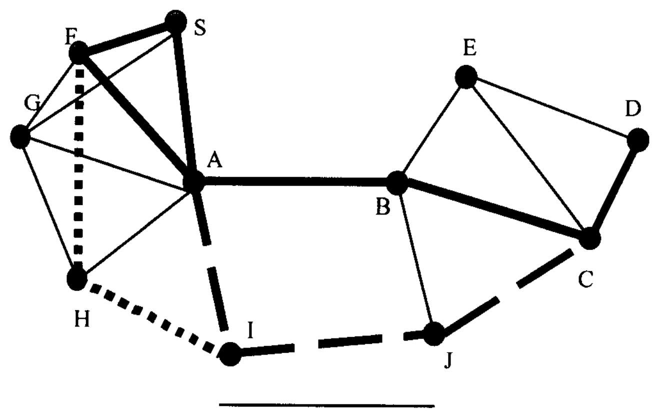

Figure 2. The first path *SABCD* and second original (*SFA*), alternate (*SFAIJCD*), and disjoint (*SFHIJ*) paths from *S* to *D*.

It was observed experimentally that $c<4$ are reasonable choices for *c*, while the additional success rate for $c>3$ does not compensate for additional flooding rate. We note that flooding effect, or the preferred choice of parameter *c*, may be related to the need to construct alternative QoS routes in the network.

3. COMPONENT ROUTING

We shall now describe the main contribution of this paper, the component routing algorithm. Consider an example in Fig. 3 as the motivation for the new algorithm. Let *A* be the source and *D* be the destination node. *A* initiates the first component in routing, by sending the packet to the best neighbor *P*. *P* is concave node in *GEDIR* algorithm, and the first component *AP* fails. As in the *f-GEDIR* algorithm [7], *P* will 'flood' its neighbors, which is *A* only, and will reject further copies of the same message. *GEDIR* algorithm initiated at *P* will stop at *Q*, completing the second component *PAQ* at concave node *Q*. Node *Q* has four neighbors: *A, E, F* and *B*. If *f-GEDIR* algorithm is now followed, each of these four nodes would initiate a separate routing task toward *D*. The results might have been routes *AE, EA, FEA* and *BEA*. However, since these routes are initiated simultaneously, node *E*, after sending its own message to *A* (provided there was no collision), receives messages simultaneously from *A, F* and *B*. The order of events is decided by MAC protocol, but clearly excessive number of messages was initiated locally while only 'flooding' message from *B*, arriving at *U, V* and *W*, is useful. Our newly proposed component routing method introduces some intelligence into the protocol at concave nodes. Node *Q* may observe that its four neighbors *A, E, F* and *B* are connected subgraph, and that therefore it suffices to 'activate' only one of these nodes. If destination *D* was connected to *Q*, after *Q* withdraws and its best neighbor, say *A* (following geographic distance as criterion) takes over (with all neighbors being connected), node *A* remains connected to *D* (a path from *Q* to *D* must go through one of *Q*'s neighbors in the first hop).

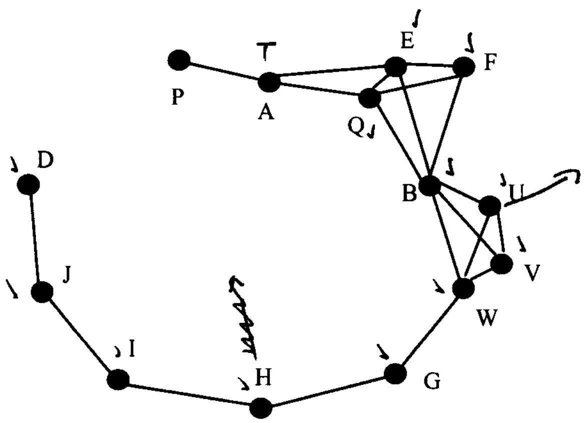

Figure 3. Components in routing from *A* to *D*: ***AP***, ***PAQ***, ***QAE***, ***EA***, ***EBWGHIJD***.

Node *A* now cannot count on two neighbors *P* and *Q* that reject further copies of the packet, and has *E* as the best option, and *E* is then a concave node, having *A* as best neighbor. However, neighbors of node *E* are now not connected (*A* is disconnected from *B* and *F*; again *Q* has to be ignored). Thus neighbors of *E* are divided into two connected components, and it is not clear, from *E*'s perspective, which of them is connected to *D*. Thus *E* must offer a chance to both components to deliver the packet. One of them, node *A*, remains lonely node and does not transmit anything. Thus in effect only one component is 'alive', and is activated by neighbor *B* that is closest to *D*. *B* forwards to *W* and path *BWGHIJD* to *D* is created. Note that, in this example, there were never two copies of the packet simultaneously running in the network. Thus, although the explanation leads to believe that there were multiple paths, in effect a single path was being created. That single path, in this example, is ***APAQAEBWGHIJD***. Note also that there are two types of messages being sent. 'Forward' messages are indicated in bold in capture of Fig. 3, while 'backward' messages, sent by concave nodes, are indicated in bold underlined letters.

However, the component routing method may indeed create multiple copies of the same packet running simultaneously in the network. Suppose that nodes *P*, *A* and *E* in Fig. 5 are not in the network, and that node *Q* wants to route a packet to *D*. Its best neighbor is then *B*, and *B* becomes concave node, having *Q* as best choice. The neighbors of *B* belong to two connected components, *Q* and *F* being one, and *U*, *V* and *W* being the other. *B* selects best options *Q* and *W* in each component, and two parallel paths, from *Q* and *W*, are simultaneously running. *Q* and *F* component finishes after two messages to each other, while the other component finds the path *BWGHIJD*. Although multi-paths can be created in the networks, experiments show that the overall flooding rate remains small, comparable or lower than in *DFS* algorithm [8] that maintains single path at all times (a variant of algorithm [8] where *DFS* is used for maintaining routing tables was independently described in [5]). It can also be observed, using simple geometric facts, that there are at most four connected components of neighbors of any concave node in unit graph model. Therefore the number of newly created components is at most three (note that one existing component terminates at concave node). However, while new component is sometimes created, the creation of two or three components is a rare event. On the other hand, the creation of multiple 'parallel' paths may lead to faster delivery.

Thus, in summary, component routing algorithm can be described as follows. The source node *S* sends message to its neighbor that is closest to the destination *D* among all neighbors of *S*. Each intermediate node *A*, upon receiving the packet from its neighboring node *B*, will determine which of its neighbors is closest to *D*. If the selected neighbor *C* is different from *B*, the message is forwarded to *C*. Otherwise, that is, if *B* is the best choice, *A* is concave node, and will determine connected components in the subgraph of its neighbors. It will select one node in each connected component (the one that is closest to *D* among nodes in that component), and will list *id*s of these nodes in the message that will also announce *A*'s resignation from the network for the purpose of delivery of that packet. The selected node *E* in each component will initiate a routing task to *D*, ignoring node *A*, if it has at least one neighboring node that did not resign from the packet (otherwise it terminates without success). Each node forwards only the first received message copy, unless repeated copy comes directly from a concave node.

The component routing resembles somewhat *DFS* based routing, but has also significant differences. Parent node is considered as an option for path continuation, and, if parent is selected, current node completes its role with possible creation of parallel search paths, whose number is limited (up to four). The overall size of search tree is therefore reduced, and the packet delivery is not delayed because of searching portion of network that is not connected to destination. Also, returned messages may take shortcut routes, as discussed in the following theorem which states the main property of the component routing scheme.

Theorem 1. The component routing algorithm guarantees delivery of packet in connected graphs even if the destination position is not accurate or consistent.

Proof. Messages can be conveniently divided into forward and backward ones, and later ones are messages sent by concave nodes. Forward messages extend the path toward destination, while backward messages report failure in the current path, and request the alteration of path that 'mislead' to concave node. The message forwarded by a node *B* to neighboring node *C* can either be delivered to destination *D*, or will be returned back to *B* by *C* after path extended from *C* has failed and *C* became concave node, or the same message will be returned by *C* to a common neighbor *E* of *B* and *C* (if a common neighbor closer to destination than *B* exists). One of paths continued by each component remains connected to *D*. Thus the first copy of packet that arrives at *B* will eventually be connected to *D*, possibly with one or more attempts if the message is returned to *B* from concave neighbors. The subsequent copies of messages, forwarded to *B* from its non-concave neighbors, can be therefore ignored to reduce flooding rate. It can be observed that the packet delivery is guaranteed by the connectivity of D to the source node, and not by its specific position or accuracy or consistency of its position at other nodes. Of course, the position errors will make the path longer. A possible problem here is that inconsistency in destination location may create a loop of forwarded messages, which may lead to its cancellation if second copy is treated as being identical to the first one. However, this problem can be easily addressed by considering time as the third dimension, that is, routing is performed in three dimensions (or four, if the routing space was 3-D). A node will not consider received copy as being the same if the destination location was updated in the path and more recent information arrives. The most recent information on destination position on the route is taken in routing decisions. The time coordinate may also be added in various ways to distance calculation, preserving guaranteed delivery property. ♦

4. CONCLUSION

The component routing method, proposed in this paper, guarantees delivery, has comparable hop count, and has lower flooding rates than any other known method proposed in literature, including three others proposed in this paper. Delivery in connected graphs can be guaranteed by flooding all nodes. The first 'intelligent' routing algorithm that guarantees delivery is described by Stojmenovic and Lin [7]. Bose, Morin, Stojmenovic and Urrutia [2] described a stateless (memoryless) perimeter routing algorithm which guarantees delivery (Karp and Kung implemented the algorithm [2] in their paper presented at MOBICOM 2000 and added MAC layer in their experiments with moving nodes). Barriere, Fraigniaud, Narajanan, and Opatrny [1] extended algorithm [1] to achieve a robust routing algorithm. Two variants of depth first search based routing algorithms that guarantee delivery are described independently in [5,7]. Component routing, described in this paper, is another such method that aims at minimizing delay while preserving low flooding rate. It requires that nodes memorize past traffic. It is a good candidate for a QoS routing scheme, where delay, bandwidth, and jitter constraints need to be respected.

The routing schemes proposed in this paper can be combined with a number of location update schemes in order to produce a full routing protocol. The ultimate goal is to find the best combination of basic routing and location update schemes, for which an in-depth study is necessary because of possible interactions between ingredients. Note however, that the guaranteed delivery, small hop count, small flooding rate and stability of component routing scheme with respect to accuracy of destination information make this method a good candidate for combining with various other proactive or reactive location update schemes.

Our algorithms produce paths of length $O(\sqrt{n})$ and therefore provide scalability and better relative performance over competitive methods for larger values of n (we have confirmed that by doing experiments with various values of n). In multi-path methods, selecting proper value for c controls flooding rate. We concluded that values $c=1$, 2 or 3 may be justified, based on network density, desired success level, and desired trade-off with flooding rate as overhead. Component routing method seems to make compromise between single and multi-path methods. It may be best to start routing with the single path, hoping to reach destination in the first attempt, and create parallel path at nodes that may 'wonder' which route leads to destination out of several routes that might exist from the local perspective.

Further improvement of the performance of the method may be obtained by applying the connected dominating set concept, as in [8]. This will reduce the search space to about half nodes for all densities, therefore considerably reducing the failing portions of the routes. The search space on the network edges may be reduced by applying planar subgraph construction, such as ones applied in [2]. These improvements are planned for our future work and search for ultimate routing scheme.

ACKNOWLEDGMENTS

This research is partially supported by NSERC and CONACyT.

REFERENCES

[1] L. Barriere, P Fraigniaud, L. Narajanan, and J. Opatrny, Robust position based routing in wireless ad hoc networks with unstable transmission ranges, Proc. of 5th ACM Int. Workshop DIAL M01, 2001.

[2] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, DIAL M, Seattle, 1999, 48-55; Telecomm. Systems, to appear.

[3] G.G. Finn, Routing and addressing problems in large metropolitan-scale internetworks, ISI Research Report ISU/RR-87-180, March 1987.

[4] S. Giordano, Mobile ad hoc networks, to appear in: Handbook of Wireless Networks and Mobile Computing (I. Stojmenovic, ed.), John Wiley & Sons, 2001, to appear.

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

[6] S. Lindsey, K. Sivalingam and C.S. Raghavendra, Power optimization in routing protocols for wireless and mobile networks, in: Handbook of Wireless Networks and Mobile Computing (I. Stojmenovic, ed.), John Wiley & Sons, 2001.

[7] Ivan Stojmenovic and Xu Lin, Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks, IEEE TPDS, to appear.

[8] I. Stojmenovic, M. Russell, and B. Vukojevic, Depth first search and location based localized routing and QoS routing in wireless networks, IEEE Int. Conf. on Parallel Processing, Aug. 21-24, 2000, Toronto, 173-180.