Beaconless Position Based Routing with Guaranteed Delivery for Wireless Ad-Hoc and Sensor Networks

Mohit Chawla[1], Nishith Goel[2], Kalai Kalaichelvan[3], Amiya Nayak[4], and Ivan Stojmenovic[4]

1 IIT Guwahati, B.Tech, Computer Science & Engineering, Guwahati, Assam, India mchawal@iitg.ernet.in

2 Cistel Technology Inc., Ottawa, ON K2E 7V7,Canada ngoel@cistel.com

3 EION Technology Inc., Ottawa, Ontario K1Y 2X5, Canada kalai@eion.com

4 SITE, University of Ottawa, 800 King Edward, Ottawa Ontario K1N 6N5, Canada {anayak,ivan}@site.uottawa.ca,

Abstract. Existing position-based routing algorithms, where packets are forwarded in the geographic direction of the destination, normally require that the forwarding node knows the positions of all neighbors in its transmission range. This information on direct neighbors is gained by observing beacon messages that each node sends out periodically. Several beaconless greedy routing schemes have been proposed recently. However, none of the existing beaconless schemes guarantee the delivery of packets. Moreover, they incur communication overhead by sending excessive control messages or by broadcasting data packets. In this paper, we describe how existing localized position based routing schemes that guarantee delivery can be made beaconless, while preserving the same routes. In our *guaranteed delivery beaconless routing scheme,* the next hop is selected through the use of control RTS/CTS messages and biased timeouts. In greedy mode, neighbor closest to destination responds first. In recovery mode, nodes closer to the source will select shorter timeouts, so that other neighbors, overhearing CTS packets, can eliminate their own CTS packets if they realize that their link to the source is not part of Gabriel graph. Nodes also cancel their packets after receiving data message sent by source to the selected neighbor. We analyze the behavior of our scheme on our simulation environment assuming ideal MAC, following GOAFR+ and GFG routing schemes. Our results demonstrate low communication overhead in addition to guaranteed delivery.

Please use the following format when citing this chapter:

Chawla, M., Goel, N., Kalaichelvan, K., Nayak, A., Stojmenovic, I., 2006, in International Federation for Information Processing (IFIP), Volume 212, Ad-Hoc Networking, ed. Al Agha, K., (Boston: Springer), pp. 61–70.

1 Introduction

Position-based routing was originally developed for packet radio networks in the 1980s [6]. It received renewed interest during the last few years as a method for routing in mobile wireless ad hoc and sensor networks [1, 2, 4]. The general idea of is to select the next hop based on position information such that the packet is forwarded in the geographical direction of the destination. Position-based routing can be divided into two main components: the location service and position-based forwarding. The *location service* [5, 13] is used for mapping the unique identifier (for example an IP address) of a node to its geographical position. In mobile ad hoc networks, providing accurate location service for position based routing, with low communication overhead, appears to be more difficult task than routing itself [13]. In case of sensor networks, however, destination is a sink or base station whose position is made available to source sensors by flooding. *Position-based forwarding* is performed by a node to select one of its neighbors as the next hop the packet should be forwarded to. Usually, the following information is required for the forwarding decision: the node's own geographical position, the position of all neighbors within transmission range and the position of the destination. Based on this information, the forwarding node selects one of its neighbors as the next hop such that the packet makes progress toward the geographical position of the destination. It is possible that there is no neighbor with positive progress towards the destination while a valid route exists. In this case, a recovery strategy [4] may be used to find a path to the destination. The most important characteristic of position-based routing is that forwarding decisions are based on local knowledge [5, 13]. It is not necessary to create and maintain a global route from the sender to the destination [11]. Therefore, position-based routing is commonly regarded as highly scalable and very robust against frequent topological changes.

In most existing strategies for position-based unicast forwarding [8], the position of a node is made available to its direct neighbors by periodically transmitting beacons. Each node stores the information it receives about its neighbors in a table and thus maintains more or less accurate position information of all direct neighbors. The transmission of beacons and the storage of neighbor information consume resources. Due to mobility, collected neighbor information can quickly get outdated which in turn can lead to packet drops. Sending and receiving beacon messages consumes energy and disturbs sleeping cycles, which is not desirable for devices with strict limitations in energy consumption [9].

In this paper, we consider position-based forwarding without the help of beacons and without the maintenance of information about the immediate neighbors of a node. Instead, all suitable neighbors of the forwarding node participate in the next hop selection process and the forwarding decision is based on the actual topology at the time a packet is forwarded. The existing beaconless routing protocols [3, 7, 9, 14] do not guarantee the delivery of the packet and also either broadcast the data packet [9] or have too many messages involved or duplicate messages [7]. We describe here a beaconless position based routing protocol which guarantees delivery in connected networks, assuming an ideal MAC layer, without collisions, and unit disk graph model without obstacles. The proposed *Guaranteed Delivery Beaconless Forwarding* (GDBF) protocol involves selecting the appropriate next hop by means of RTS (Ready To Send) and CTS (Clear To

Send) packets. In greedy mode, similarly as in [14], only the forwarding neighbor sends CTS back to the node having the data packet. GDBF is a generic framework that can be applied to location based schemes. It guarantees delivery, when the underlying protocol is a guaranteed delivery protocol. The main contribution of this article is in protocol operation in recovery mode. GDBF reduces the number of messages (CTS's) sent by the neighbors of current node in recovery mode by using a special suppression scheme. We show that only neighbors of current node in Gabriel graph (of the set of all nodes) respond in our scheme, so that current node is enabled proper choice of forwarding node and preserving routes as if 1-hop knowledge was available.

We assume that nodes are placed in the Euclidean plane. In order to represent ad-hoc networks we adopt the widely used model, where every node has the same transmission range, without loss of generality normalized to 1. The resulting graph, having an edge between two nodes u and v if and only if the Euclidean distance $| uv | \leq 1$, is the unit disk graph (UDG).

We begin the paper by outlining the details of the algorithm that we simulate, namely GFG [4], in Section 2, which also describes some of the existing beaconless greedy routing schemes. Following this, GDBF has been fully explained in Section 3. We analyze GDBF on our simulation environment assuming the ideal MAC, with GFG and GOAFR+ as the underlying protocols, in Section 4. In Section 5 we conclude the paper and present some of the insightful issues that can be the subject of future research.

2 Related Work

2.1 Beaconless Greedy Routing

Heissenbuttel and Braun [9] proposed the *beacon-less routing (BLR)* algorithm. The *contention-based forwarding (CBF)* by Fussler *et al.* [7] and *implicit geo- graphic forwarding (IGF)* by Blum *et al.* [3] are also implementing similar ideas, focusing on the integration of beaconless routing with the IEEE 802.11 MAC layer. Node, currently holding the packet with known destination, is generally not aware of any of its neighboring nodes and simply broadcasts a data packet. The main idea of beacon-less routing is that each neighboring node, receiving the packet, calculates a small transmission timeout (before forwarding the packet) depending on its position relative to the last node and the destination. The node located at the "best" position introduces the shortest delay and will retransmit the packet first. The remaining nodes then cancel the scheduled packet after detecting this transmission. However, some neighbors with forward progress may not hear the message, and can also retransmit it. Hence, in [9], only nodes within a certain forwarding area are allowed as candidate nodes for the next forwarding step. The forwarding area has the property that each node is able to overhear the transmission of every other node within that area. However, because of this forwarding area this scheme fails to exploit all possible forwarding neighbors, and therefore has reduced success rate.

Fuessler et al [7] propose a technique called *active selection method.* A forwarding node sends to all its neighbors a control packet instead of the full message. Neighbors that provide forward progress respond after a timeout, which depends on their distance to the destination. If a neighbor hears another neighbor's response then it does not respond itself. The forwarding node then sends the full message indicating which of its neighbors shall forward the message. In a similar way, Zorzi [14] proposed a scheme to avoid duplicate forwarding in a beaconless routing scheme by applying the RTS/CTS MAC scheme. The current node sends an RTS signal instead of the message. Afterwards the node waits for a node to respond with a CTS signal. If several responses are received, the node selects one that looks best for forwarding and then sends the packet to that neighbor directly.

However, none of the existing beaconless routing protocols discussed how to guarantee the delivery of the packets. Moreover, they either retransmit the whole data packet immediately [9], potentially generating superfluous additional retransmissions, or have too many messages involved or duplicate messages [7]. The GDBF protocol proposed in this paper guarantees the delivery of the packets and entails less communication overhead by using a special suppression scheme.

2.2 GFG (Greedy-Face-Greedy)

The GFG algorithm [4] is a combination of greedy routing and face (or recovery/perimeter) mode routing. We first describe face routing. Gabriel graph (GG) property is used to preserve only edges that leave connected planar graph from the initial UDG. An edge PQ is included in GG if and only if there is no other node in the disk with diameter PQ. The construction of planar graph is possible without any messages, assuming each node is aware of positions of its neighbors. Planar graph created by GG divides the plane into faces. Imaginary line from source S to destination D passes through several faces between them. These faces are traversed, changing faces at intersections of imaginary line SD with the faces. Whenever a new face is entered, it can be traversed in one of two possible directions, clockwise (CL) or counterclockwise (CC), and direction could change for a reason, leading to several variants of the protocol (e.g. GOAFR+ variant [12]).

During face traversal, node U forwards message to neighboring node V together with the selected direction (CL or CC) to follow, which indicates also which of the two possible faces has been traversed. The next neighbor to forward is the one making minimal angle with respect to incoming edge and given direction (e.g. E for CL and H for CC in Fig. 2). When such rule (called also right/left hand rule for CL/CC direction respectively) is applied repeatedly, selected face can be traversed fully (e.g. face $VGIJKF$ in Fig. 2), which guarantees finding the next intersection with the imaginary line and making progress toward the destination. Note that local orientation for neighbor selection and global face orientations are opposite for closed faces (See Fig. 2).

The GFG algorithm begins by routing greedily; that is by forwarding the message at each intermediate node to the neighbor located closest to the destination D. Doing so, however, the algorithm can reach a local minimum with respect to the distance from D, that is a node G none of whose neighbors is located closer to D than G itself. From this position the algorithm recovers by routing around the *perimeter* of the region

by carefully selecting neighbors through the recovery mode protocol. Fig. 3 illustrates the two modes.

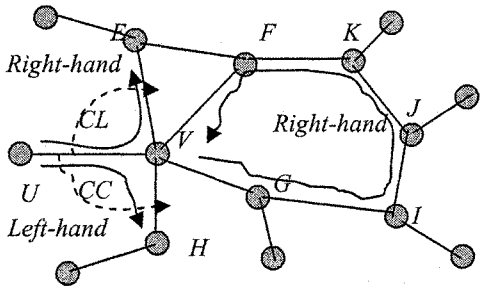

Figure 2. Continuing face traversal

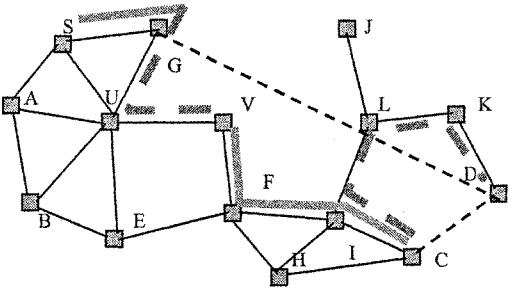

Figure 3. Greedy *SG*, face *GUV*, greedy *VFIC*, face *CILKD* modes in *GFG* algorithm

When a packet enters perimeter mode e.g. at *G* (Fig. 3), GFG records in the packet the location *G*, the site where greedy forwarding failed. This location is used at subsequent hops to determine whether the packet can be returned to greedy mode. GFG returns a packet to greedy mode if the distance from the forwarding node to *D* is less than that from *G* to *D*. Packet also contains source, destination, and position of last intersection *X* with the imaginary line (the next intersection is valid only if it is between *X* and *D*). When entering a new face, packet needs also to carry the first edge on it and the selected direction. If the packet discovers them again later on, a loop has been created which indicates that destination is disconnected from the source. Similar scenario may be created with node mobility, but it can be also then prevented by recording the time face was entered, and ignoring GG edge changes occurring afterwards.

3 GDBF (Guaranteed Delivery Beaconless Forwarding)

The GDBF algorithm proposed in this paper does not require beacons and thus completely eliminates the proactive part of position-based routing. Routes in GDBF algorithm are the same as in the underlying position based routing protocol to which it is applied. More precisely, GDBF is a general framework, with particular instances such as BGFG and BGOAFR+ which are built by applying GDBF on GFG and GOAFR+, respectively. BGFG and BGOAFR+ are beaconless forms of GFG and GOAFR+. Assuming the same neighbors at the same positions, GDBF always selects the same neighbor as the underlying protocol, in both recovery and greedy modes. That is, routes in BGFG (BGOAFR+) are the same as routes in GFG (GOAFR+, respectively) as if 1-hop knowledge was available. GDBF algorithm uses control messages to select the appropriate next neighbor rather than periodic beacons, hence it always selects the neighbor dynamically. Frequent topological changes impact the accuracy of neighborhood information in beacon based protocols, while the proposed protocol resolves this problem with the assumptions stated earlier in section 1. The beaconless protocol GDBF works in three steps. First, the forwarding node transmits the RTS packet

as a single-hop broadcast to all neighbors. The packet contains request to forward the message but not the message itself. It also contains one bit indicating whether the request is for greedy or for recovery mode assistance. Next, the neighbors compete with each other for the "privilege" to forward the packet. During this period, a node determines how well it is suited as a next hop for the packet and sets a timeout depending on its suitability. While waiting, nodes may receive more responses from other neighbors and decide whether or not to respond at the end of their timeouts. Response, if sent, is in the form of a CTS. Finally, the sender node decides which neighbor is most appropriate among those that responded, and forwards the message to that node. Details differ in greedy and recovery modes.

In case of the greedy mode, neighbor's timeout is inversely dependent on distance of that neighbor to the destination. Therefore, the node that is selected is the closest to the destination from among all neighbors. Only nodes that are closer to destination than S set such timeout and potentially respond. Once a CTS is received, S transmits the message (data packet) to the neighbor that has just sent the CTS. That message also cancels the transmission by other neighbors. Thus duplicate messages or routes in the network are avoided because a unique CTS is allowed. In Fig. 4, A and B are the greedy nodes because they are closer to the destination D than the current node S. However B is closer to D than A and hence B will have the smallest timeout and will be the first one to send the CTS to S. Nodes overhearing that CTS, like A, will automatically cancel their message. Node S forwards full message to B. This message cancels CTS from other neighbors which did not hear the CTS from B. Therefore, in greedy mode, exactly one neighbor sends CTS message. Note that our GDBF protocol in the greedy mode is similar to the one described in [14].

In a specific MAC implementation, discrete values of timeout are required. It is then possible that two or more of neighbors nearest to D will then respond simultaneously, causing collisions. In such case, node S will issue request for retransmissions to these nodes. They will in turn select new random timeouts, ignoring their positions, to avoid another collision. Similar resolution can be applied in the case of recovery mode.

In case of the recovery mode, the neighbor's timeout is based on the closeness to the current node S having the packet rather than the destination. Closer nodes to S have shorter timeouts. Once its timeout expires, node A responds with a CTS to S. That response subsequently cancels all the transmissions from neighbors X that find A to be located inside in the circle with diameter SX. We now show the basic property of this protocol.

Theorem 1. In described protocol, a neighbor A responds to S with a CTS if and only if SA belongs to the Gabriel graph (GG) of the set of all nodes.

Proof. Suppose that a neighbor X is such that SX is not in GG. Then there exists another neighbor A so that A is inside circle with diameter SX. But then $|AS|<|XS|$ and the timeout of A is shorter than the timeout of X. Therefore A transmits CTS before the timeout at X expires, and CTS transmission of X is therefore canceled. Suppose now that SX belongs to GG. Then there is no other node inside the circle with diameter SX, and transmission from X, at the end of its timeout, is not cancelled, according to the protocol. Note that X may still hear CTS from some other neighbor A which is not inside the considered circle.♦

Therefore the number of CTS responses to the current node *S* is exactly equal to the number of edges in Gabriel graph, with an endpoint at *S*. Knowing all edges in GG, sender *S* (in recovery mode) then follows the corresponding algorithm and makes the decision which of them should receive the message. The message is sent to that node, and routing continues along the same path as if position of all neighbors was known to *S*. The details depend on particular protocol being followed. Normally, *S* selects forwarding node that creates the smallest angle toward the incoming packet direction, in decided direction, either clockwise or counterclockwise, following the selected protocol (e.g. GFG or GOAFR+). Initially (in our implementation) it chooses to route the packet to the first edge counterclockwise about *S* from the line *SD* (different decision criterion is possible, such as smaller of the two angles) and it continues perimeter traversal till it reaches a point where it shifts back to greedy mode or it reaches the boundary of the enclosing circle/ellipse. In the latter case it reverses its direction of traversal and starts traversing the network in the opposite direction to the present one. Note that, in this case, node receiving packet may simply decide to return it to the previous sender node, in which case no 'competition' is announced, that is, no RTS or CTS messages are sent.

The protocol has details about transition from greedy to recovery and from recovery to greedy modes. The running mode is indicated in the packet from *S*, therefore each neighbor starts proper timeout.

When the Greedy algorithm reaches local minima, there is no CTS response to the RTS, before the timeout in the source node itself expires. That is, there is no neighbor that is closer to the destination than the current node *S*, and the algorithm shifts to the recovery mode. *S* then again transmits a RTS but this time the bit for the selected mode indicates recovery mode. Neighbors then start another timeout, as described for the recovery mode, and the protocol proceeds accordingly.

The shift from the recovery mode to greedy mode in GDBF is dependent on the underlying routing protocol to which the GDBF framework is applied (GOAFR+ & GFG in our case). Beaconless GFG (BGFG) is GDBF framework applied to the GFG routing protocol. BGOAFR+ follows similarly. Sender node S includes, in the RTS packet, the distance of the node that switched to recovery mode from the destinations. If some neighbors detect that they are closer to destination than that distance, they are eligible to respond and convert to greedy mode. Since only one of them is 'allowed' to do so, they start a timeout based on their distance to destination, so that again the closest one wins, that is, responds first with a CTS indicating also transfer to greedy mode. This also means that *S* needs to wait for all such possible timeouts to expire before proceeding with the recovery mode. Therefore several neighbors may already respond offering 'services' for the recovery mode, before the best neighbor for converting to the greedy mode responded. In case of BGFG, there is immediate fallback to greedy mode, because as soon as a node is encountered that is closer to the destination than the source node where we started the recovery mode, the protocol will shift back to the greedy mode. This is in line with the GFG [4] protocol described earlier, because the paths have to be same to guarantee the delivery of packets.

In case of BGOAFR+, response from a neighbor *G* offering 'greedy' service does not necessarily imply the change in the mode. If the node that has the packet detects that it is closer to destination than that distance, it increments its counter *p* (otherwise it increments counter *q*) and checks for the condition for fallback to the greedy mode. If the

condition is satisfied, the protocol shifts back to the greedy mode and the packet is forwarded to G.

The value of the timeout is in the interval [0, *Max_Timeout*] and depends on the relative position of current node (node sending the RTS), and destination node. Eventually, the node which computes the shortest timeout as per the formula for greedy and recovery modes, is the first one to send back the CTS. The timeout at a particular node in case of the recovery mode is given by $Timeout = Max\_Timeout*(u/r)$ while in the case of greedy mode it is given by

$$Timeout = Max\_Timeout*((r+d-|SD\backslash)/|SD|)$$

The above timeouts are only defined for neighbors within the transmission region with a radius r. Here, u is the distance from the node S currently having the packet, and d is distance from neighboring node to destination D.

In case of recovery mode, the current node has to wait for all the neighbors to reply, and hence it has to wait for the complete *Max_Timeout* to elapse, following which it selects the best neighbor out of the CTS's that it has received. In greedy mode, current node will act as soon as the first neighbor responded.

4 Simulations

In this section we present a detailed analysis of GDBF framework applied to GOAFR+ and GFG (before crossing variant of GFG is followed) as the underlying protocols. Our results show performance characteristics in terms of control messages. We measure the number of control messages sent on average by nodes for the whole routing cycle and also the average number of neighbors that send the CTS in the recovery mode. We assume an ideal MAC layer without collisions. Timeouts are real numbers and they are assumed to be distinct.

Table 1 shows the average number of messages that are sent by the neighbors in the recovery mode and in the complete routing cycle. We simply calculate the average on the number of the CTS's sent by the neighbors when the routing is in recovery mode. The count does not include messages offering conversion to greedy mode. As we can observe from the first two columns of Table 1, the number of messages on average from neighbors in Recovery mode slowly increases as the density increases. This is because the number of neighbors eligible as possible next hop nodes will keep on increasing as the density increases. The probability of greedy algorithm getting stuck is very low at high density and hence the protocol rarely invokes recovery mode.

The last two columns of Table 1 present the average number of CTS messages (including both greedy and recovery types of responses) per hop that are sent by the neighbors during the whole routing cycle. This average value increases as we approach the critical density (= 4.71) [12] and then decreases afterwards. For higher network densities, local minimum for greedy routing is occurring with lower probabilities, and greedy routing can be expected to reach the destination with increasing probability. Thus the decrease in average number of CTS messages is observed. For the network of high densities, e.g. over 12 nodes per unit disk, the value approaches 1, since only one CTS from the best host is expected.

Table 1. Average Number of CTS messages from neighbors (RM: Recovery Mode; CRC : Complete Routing Cycle)

Network Density	BGOAFR+ (RM)	BGFG (RM)	BGOAFR+ (CRC)	BGFG (CRC)
3.50	2.423	2.410	1.587	1.599
4.00	2.680	2.645	1.889	1.879
4.50	2.736	2.693	2.050	2.042
4.75	2.742	2.720	2.115	2.099
5.00	2.771	2.745	2.084	2.088
5.25	2.789	2.761	2.161	2.159
5.50	2.810	2.857	2.121	2.106
6.00	2.874	2.851	2.005	1.970
6.50	2.892	2.884	1.892	1.885
7.00	2.934	2.890	1.782	1.820
7.50	2.962	2.926	1.732	1.669
8.00	3.033	2.950	1.570	1.541
8.50	3.065	2.969	1.499	1.455
9.00	3.071	2.986	1.432	1.421
9.50	3.087	2.990	1.350	1.320
10.0	3.103	3.055	1.274	1.260
10.5	3.130	3.194	1.253	1.268
11.0	3.191	3.042	1.224	1.208
11.5	3.166	3.058	1.192	1.176
12.0	3.170	3.079	1.167	1.149
12.5	3.179	3.189	1.139	1.119
13.0	3.269	3.219	1.120	1.108
13.5	3.339	3.243	1.102	1.088
14.0	3.378	3.155	1.083	1.069

5 Conclusions

There are several directions for extending and improving the results presented here. The impact of more realistic medium access layer needs to be studied, and protocol adjusted accordingly. The timeout needs to be discretized, following, for example, IEEE 802.11 standard. That means that obtained timeouts need to be rounded to nearest integer in interval [1, 32], for example. What would be the impact of collisions encountered by CTS messages in the protocol? In greedy mode, it is possible that two or more neighbors are roughly at the same distance from destination, and could select the same slot for reporting. In such case, current node, after detecting collision, should issue request for retransmitting their offer to help. Neighbors whose CTS message collided then need to select new random timeouts and respond again, this time hopefully without collisions. The distance from destination is not needed for setting the second timeout since only few 'winners' that just collided will try again. After the first response is received, others can be suppressed similarly. In recovery mode, neighbors along GG tend to be close to given node, and therefore could frequently be at approximately the same distance. Therefore

collisions of messages from them can be expected. Similar retransmission requests from them could be required from current node, which will 'suspend' other timeouts until the collision is resolved. GDBF, in greedy mode, provides clear winner since only one neighbor will respond. On the other hand, about three neighbors respond in recovery mode. It is an interesting open problem to improve this response amount and design a technique that will select proper neighbor with reduced number of CTSs, on average. As an intuition, a different timeout formula could incorporate the angle that neighbors form with last edge on the route (measured in the desired direction), in addition to distance from sender node, so that the desired neighbor on Gabriel graph responds before neighbors on other GG edges. However, the problem is not trivial since canceling the CTS from any GG neighbor may allow other nodes, not on GG, to offer services. This could potentially invalidate the method or cause even more responses than in current method. Therefore the problem is nontrivial and requires careful investigation.

6 References

[1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A Distance Routing Effect Algorithm for Mobility (DREAM), MobiCom'98, pages 76–84, Dallas, Texas, October 1998.
[2] L. Blazevic, S. Giordano, and J.-Y. LeBoudec. Self Organizing Wide-Area Routing, Proceedings SCI 2000/ISAS 2000, Orlando, July 2000.
[3] B. M. Blum, T. He, S. Son, and J. A. Stankovic. IGF: A state-free robust communication protocol for wireless sensor networks. TR CS-2003-11, University of Virginia, April 2003.
[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc Wireless Networks. ACM DIALM, 1999.
[5] T. Camp, J. Boleng, and L. Wilcox. Location Information Services in Mobile Ad Hoc Networks. Proc. IEEE ICC, 3318–3324, April 2002.
[6] G. G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, March 1987.
[7] H. Fuessler, J. Widmer, M. Kasemann, M. Mauve, Beaconless Position Based Routing For Mobile Ad-Hoc Networks. Ad Hoc Networks 1 (2003) 351–369.
[8] S. Giordano, I. Stojmenovic, Position based routing algorithms for ad hoc networks: A taxonomy, in Ad Hoc Wireless Networking; eds. X. Cheng et al Kluwer,2003,103-136.
[9] M.Heissenbuttel and T. Braun, A novel position-based and beacon-less routing algorithm for mobile ad-hoc networks, ASWN' 03, Bern, 2003, 197-210.
[10] Q. Huang, C. Lu, G.C. Roman, Reliable mobicast via face-aware routing, IEEE INFOCOM 2004.
[11] J. Hou, N. Li, I. Stojmenovic, Topology construction and maintenance in wireless sensor networks, in: Handbook of Sensor Networks: Algorithms and Architectures (I. Stojmenovic, ed.), Wiley, 2005, 311-341.
[12] F. Kuhn, R. Wattenhofer, Y. Zhang and A. Zollinger, "Geometric AdHoc Routing: Of Theory and Practice," PODC, 2003.
[13] I. Stojmenovic, Location updates for efficient routing in wireless networks, in: Handbook of Wireless Networks and Mobile Computing, Wiley, 2002, 451-471.
[14] M. Zorzi. A new contention-based mac protocol for geographic forwarding in ad hoc and sensor networks. IEEE Conf. Communications (ICC 2004), Paris, 2004.