Scalable Position-Based Multicast for Mobile Ad-hoc Networks

Matthias Transier*, Holger Füßler*, Jörg Widmer†, Martin Mauve‡, and Wolfgang Effelsberg*

*University of Mannheim, Institut für Informatik, 68131 Mannheim, Germany
{transier,fuessler,effelsberg}@informatik.uni-mannheim.de

†EPFL, Laboratory for Computer Communications and Applications (LCA), 1015 Lausanne, Switzerland
joerg.widmer@epfl.ch

‡University of Düsseldorf, Institut für Informatik, 40227 Düsseldorf, Germany
mauve@cs.uni-duesseldorf.de

Abstract

In this paper we present Scalable Position-Based Multicast (SPBM), a multicast routing protocol for ad-hoc networks. SPBM uses the geographic position of nodes to provide a highly scalable group membership scheme and to forward data packets with a very low overhead. SPBM bases its multicast forwarding decision on whether there are group members located in a given direction or not, allowing for a hierarchical aggregation of group members contained in geographic regions: the larger the distance between a region containing group members and an intermediate node, the larger can this region be without having a significant impact on the accuracy of the direction from the intermediate node to that region. Because of aggregation, the overhead for group membership management is bounded by a small constant while it is independent of the number of multicast senders for a given multicast group. We investigate the performance of SPBM by means of simulation, including a comparison with ODMRP.

1. Introduction

Many applications envisioned for mobile ad-hoc networks rely on group communication. Messaging during disaster relief, networked games, and emergency warnings in vehicular networks are common examples for these applications. As a consequence, multicast routing in mobile ad-hoc networks has received significant attention over the recent years.

In this paper we present Scalable Position-Based Multicast (SPBM), an ad-hoc multicast routing protocol comprising a multicast forwarding strategy and a group management scheme to determine where members of a multicast group are located. The forwarding strategy uses information about the geographic positions of group members to make forwarding decisions. In contrast to existing approaches it neither requires the maintenance of a distribution structure (i.e., a tree or a mesh) nor resorts to flooding. The group management scheme uses knowledge about geographic positions for a hierarchical aggregation of membership information.

The forwarding of packets by SPBM is a generalization of position-based unicast routing as proposed, e.g., in [2] and [6]. In these protocols, a forwarding node selects one of its neighbors as a next hop in a *greedy* fashion, such that the packet makes progress toward the geographic position of the destination. It is possible that a node has no neighbor with progress toward the destination although a valid route to the destination exists. The packet is then said to have reached a local optimum. In this case, a *recovery strategy* is used to escape the local optimum and to find a path toward the destination. The most important characteristic of position-based routing is that forwarding decisions are based only on local knowledge. It is not necessary to create and maintain a global route from the sender to the destination. Therefore, position-based routing is commonly regarded as highly scalable and very robust against frequent topological changes. In order to extend position-based routing to multicast, SPBM provides an algorithm for splitting multicast packets at intermediate nodes when destinations for that packet are no longer located in the same direction. This strategy includes both greedy forwarding and the recovery strategy.

The second important element of SPBM is its group management scheme. It relies on geographic information to achieve scalability: instead of maintaining a fixed distribution structure, an intermediate node just needs to know whether group members are located in a given direction or not. This allows a hierarchical aggregation of membership information: the further away a region is from an intermediate node the higher can be the level of aggregation for this region. Therefore, group membership management can be provided with an overhead that scales logarithmically with the number of nodes and that is independent of the number of multicast senders in a multicast group. A second observation is then used to reduce this overhead further: the higher the level of aggregation the lower the frequency of membership changes for the aggregate. In SPBM, we therefore propose to scale down the frequency of membership update messages exponentially with the level of aggregation. This results in a constant upper bound on the overhead as the size of the network increases.

The remainder of this paper is structured as follows: In the next section, we discuss related work. We describe the SPBM protocol in Section 3. Section 4 contains simulation results on the performance of SPBM as well as a comparison to ODMRP. Section 5 concludes the paper and gives an outlook on future work.

2. Related work

In the following we discuss existing position-based approaches to realize multicast in mobile ad-hoc networks. For a more detailed discussion please refer to [13].

Knowledge about the geographical position of nodes has been used for Dynamic Source Multicast (DSM) [1]. In DSM each node floods the network with information about its own position, thus each node knows the position of all other nodes in the ad-hoc network. The sender of a multicast packet then constructs a multicast tree from the position information of all receivers. This tree is encoded in the header of the packet. While DSM uses location information, the resulting distribution tree is completely determined by the sender. This eliminates the most important advantage of position-based routing: the exclusive use of local information. In addition, due to periodic flooding of the network, the scalability of this approach is limited.

In [3], the authors report on "Location-Guided Tree Construction Algorithms" using the position of nodes to build an application-level distribution tree. This approach enjoys the benefits of position-based routing but it is limited to receiver groups small enough so that the address of each destination can be included in each data packet.

A generalization of position-based unicast forwarding has been described in [8]. As for the "Location-Guided Tree Construction Algorithms" the sender includes the addresses of all destinations in the header of a multicast packet. In addition the location of all destinations is included as well. It remains open how the sender is able to obtain the position information and the scaling limitations seem to be similar to those discussed above.

In contrast to the existing position-based multicast protocols, SPBM retains the advantages of position-based routing while not being restricted to small receiver sets.

3. The protocol

We now introduce the two building blocks of our algorithm. The *group management scheme* is responsible for the dissemination of the membership information for multicast groups, so that forwarding nodes know in which direction receivers are located. The *multicast forwarding algorithm* is executed by a forwarding node to determine the neighbors that should receive a copy of a given multicast packet. This decision is based on the information provided by the group management scheme. In the following, we assume that each node in the network is able to determine its own position, e.g., through the use of GPS.

3.1. Group management

Position-based multicast requires that the forwarding nodes know the locations of the destinations. Including all of the destinations explicitly in the data packet header does not scale well as the size of the multicast group increases. To improve scalability, our proposal introduces hierarchical group membership management.

To this end, the network is subdivided into a quad-tree with a predefined maximum level of aggregation L. Figure 1 shows a quad-tree with four levels. Single squares are identified by their concatenated level-n to level-1 square numbers. In the example the identifier "442" identifies a level-0 square that is located in the level-3 square comprising the whole network, in the level-2 square "4" and in the level-1 square "44". In level-0 squares, all nodes are within radio range of each other (i.e., level-0 squares have at most a diameter of half the radio range).

3.1.1. Algorithm

The aim of the membership update mechanism is to provide each node in the ad-hoc network with an aggregated view of the position of group members. For this purpose, each node maintains a global member table containing entries for the three neighboring squares for each level from level 0 up to level $(L-1)$. In addition each node has a local member table for nodes located in the same level-0 square.

Each entry in the global member table consists of the square's identifier and the aggregated membership information of all nodes contained in that square. Each entry in the local membership table consists of a node ID and the membership information of that node. Membership information

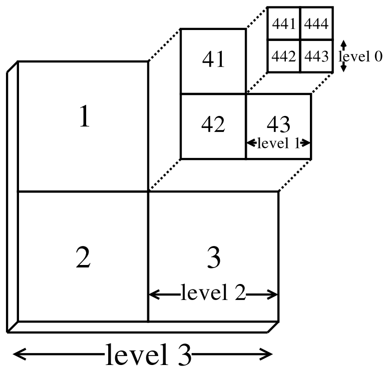


Figure 1. Network represented by a quad-tree ($L = 3$)

Table 1. Global and local member table of a node located in square "442"

Square	Groups
1	00011100
2	01000100
3	10100010
41	01010000
42	00010101
43	00100100
441	00000100
443	00010000
444	00100100

Node	Groups
14	00000001
23	01000100
51	00000100

is stored and transmitted as membership bit vectors. For simplicity, we assume that each bit represents one multicast group.[1] A bit set to 1 indicates group membership. Thus the amount of state maintained in a node scales logarithmically with the size of the network. Table 1 shows an example for a node located in square "442" with a membership vector length of 8. In this example the first entry of the global member table can be interpreted as follows: there is at least one multicast receiver for groups 3, 4 and 5 located in the level-2 square "1". The first entry of the local member table contains the information that node 14 is in the same level-0 square as the node maintaining the table and that 14 is member of group 7.

A node indicates its group membership status by broadcasting *announce* messages within its level-0 square (i.e., its direct neighbors). An announce message contains the ID of the node and a membership vector describing its subscribed groups. Announce messages are broadcast periodically, but need not be forwarded by any other node since all nodes within the same level-0 square are within radio range of each other.

A node stores the membership information of all nodes in its level-0 square. Update messages are then used to provide all nodes that are located in a level-1 square with the aggregated membership information of the four level-0 squares contained in the level-1 square. This is done by periodically selecting one node in each level-0 square. For now we assume that such a selection mechanism is in place. We will show later how it can be realized by means of random timers. The selected node floods the level-1 square with an update message including the ID of the selected node, a membership vector describing the aggregated group membership information, the identifier of the destination square that is to be flooded, and a sequence number for duplicate message detection. The aggregation is done by a bitwise or-operation on the membership vectors of the nodes located in the level-0 square. In order to perform flooding, each node in the level-1 square forwards this message once. In total, there will thus be four update messages flooded in each level-1 square per period, one for each level-0 square. In the example, one node in each square "441", square "442", square "443", and square "444" is selected. Those nodes aggregate their level-0 membership information and flood them in an update packet in the level-1 square "44".

The same mechanism is used to aggregate the membership information from an arbitrary level-λ square and flood it in the area of a level-$(\lambda + 1)$ square. In the example one node in each square "41", square "42", square "43", and square "44" would be selected to aggregate their level-1 membership information and flood an update message in square "4". If the node with the membership tables depicted in Table 1 would be selected for square "44", it would perform the aggregation by a bitwise or-operation on the membership vectors for the individual nodes 14, 23, 51 and on the aggregated information from the level-0 squares "441", "443", and "444".

Since the size of a square increases exponentially with each level, the likeliness that the aggregated group membership information changes in a given time-span decreases rapidly. We therefore propose to decrease the frequency of flooding membership information exponentially with the level of aggregation.

It remains to be shown how one node is selected to send an update message. The selection mechanism is performed by random timers. Every node maintains an update timer for each level. When the timer expires the node is selected, transmits the update message for the appropriate level and resets the timer. When a node receives an update message for a square that it belongs to, its timer is reset without sending the packet, suppressing the transmission of the update

[1] In practice, group membership information should be hashed to the bit vector to facilitate the assignment of group membership IDs.

message. The main component of each timer is determined by the update frequency of that level. In order to avoid that all nodes in a given square flood the same update information simultaneously, each timer has also a random exponential element. This behavior is adapted from [10]. Given a constant node density it can be shown that the amount of data transmitted per m^2 by this group management scheme is bounded by a small constant as the size of the network increases to infinity [13].

3.2. Multicast forwarding

To deliver multicast packets from a source to the subscribed group members, the nodes use the information stored in their member tables. By dividing the network into a quad-tree, geographic regions are build which can be used to aggregate multicast traffic to group members located geographically close to each other.

The forwarding decision is based on information about neighboring nodes. Each node maintains a table of nodes in its transmission range. This is accomplished by having each node periodically broadcast beacon messages containing the ID and position of the node. Beacon messages are not forwarded by the receiving nodes.

Algorithm 1 shows the forwarding algorithm. As an input, the algorithm requires the current node n, the packet p and the list of neighbors N of n. The packet includes a list-of-destinations field which is initially set to one entry that comprises the whole network and a group address field indicating the group the packet is sent to. Once the algorithm is invoked, it first checks whether the current node n is a member of the multicast group the packet is sent to. If this is the case, then the packet is delivered.

In the next step the algorithm looks at each entry in the list-of-destinations field of the packet: if the global or the local membership tables contain a de-aggregation of the entry, then the entry is subdivided into those squares of the next lower level that include members for the group the packet is transmitted to. At level-0 a de-aggregation is performed by replacing the square with the IDs of the nodes that are group members.

For example, consider the situation where a node in square "442" (see Figure 1) sends a multicast packet to the group number 1. It initializes the packet with the whole network as the single destination area and sets the multicast address to 1. Then the packet is handed to the forwarding algorithm. After checking whether the current node is a receiver of multicast group 1 the destinations are de-aggregated: based on the membership tables given in Table 1 for multicast group 1, the complete network can be de-aggregated into the level-2 square "2" (since bit 1 of the membership vector is set), the level-1 square "41", and the individual node 23 in the same level-0 square as the forwarding node.

```
Require: node n, packet p, list of neighbors N
  if n ∈ receivers(group(p)) then
    deliver(p)
  end if
  D ← ∅
  for all d ∈ destinations(p) do
    if mysquare ⊆ d then
      D ← D ∪ subdivide(d)
    else
      D ← D ∪ d
    end if
  end for
  F[N] ← ∅
  for all d ∈ D do
    ν ← ∅
    if recover(d) then
      ν ← rightHand(prevHop, d)
    else
      ν ← forwardGreedy(N, d)
    end if
    if ν = ∅ then
      ν ← rightHand(n, d)
      if ν = ∅ then
        drop(d)
      end if
    end if
    F[ν] ← F[ν] ∪ d
  end for
  for all ν ∈ N do
    if F[ν] ≠ ∅ then
      send(p, ν, F[ν])
    end if
  end for
```

Algorithm 1. The forwarding algorithm

After de-aggregation of the destinations it is checked which neighbor is best suited to forward the packet to each destination. This is done in a fashion similar to position-based unicast routing (see [9]): in order to determine the most suitable next hop for a packet and a given destination, the source compares the geographic progress for each of the neighbors in respect to the destination and picks the neighbor with the highest progress. In case that the destination is a square, the position of the nearest point in that square is used as the destination position.

After finding the next hop for each destination, the current node n makes a copy of the data packet for each of these next hops. In the list-of-destinations field, it enters a list of the destinations which shall be reached through this

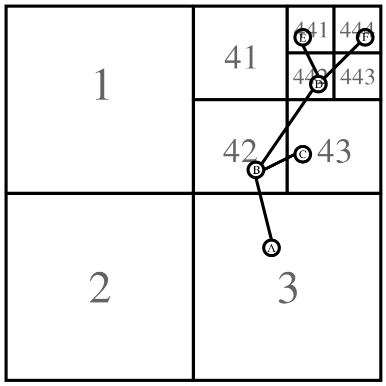


Figure 2. Forwarding on the quad-tree

specific next hop and sends the packet to the next hop by using unicast transmission. The use of unicast increases the reliability of data delivery at the expense of bandwidth utilization as each copy of the packet will be acknowledged on the MAC layer but has to be sent separately.[2]

Figure 2 shows an example of the forwarding procedure.[3] Node *A* wants to send a packet to the group in which nodes *C*, *E* and *F* are members. Thus *A*'s member table contains the information that there is at least one receiver in square "4". It sends the packet in this direction and node *B* is the first node located in the level-2 square "4". Consequently, it has the information that there are nodes subscribed to the group in the level-1 squares "43" and "44". It therefore updates the information in the packet header accordingly. Node *C* is the first forwarding node in square "43". Besides delivering the packet, it checks its member table and recognizes that it does not need to forward the packet to any additional receivers in square "43". In square "44", node *D* replaces square "44" in the packet header by the level-0 squares "441" and "444". After receiving the packet, nodes *E* and *F* replace their square by potential additional destination nodes in this square. If there were any, the packets would now directly be sent to the receivers since the radio ranges of *E* and *F* cover the complete squares "441" and "444", respectively.

If, for one or more destinations, a forwarding node does not find a next hop that yields geographic progress, a recovery strategy has to be employed. Similar to position-based unicast routing [6, 2], SPBM uses a distributed planarization of the network graph combined with the right-hand rule to route around void regions. When there is a destination with no suitable next hop, the algorithm first planarizes the surrounding network graph. Then, the node determines the angles counter-clockwise between the line from the node to the destination and the line from the node to each remaining neighbor. The neighbor with the smallest angle is chosen as the next hop. The packet is then marked to be in recovery mode and the position where the packet entered recovery mode is stored in the packet.

A node receiving a packet which is in recovery mode first checks whether itself is located closer to the destination than the position which is stored in the packet as the recovery starting point. In this case, regular forwarding is resumed. If this is not the case the node has to continue the recovery process. After performing planarization, it chooses the next hop using the right-hand rule.

[2]This is a design decision, depending on the application and the environment of the ad-hoc network one may choose to transmit the packet using broadcast.

[3]The figure only depicts nodes which are involved in the process of refining the destination square information.

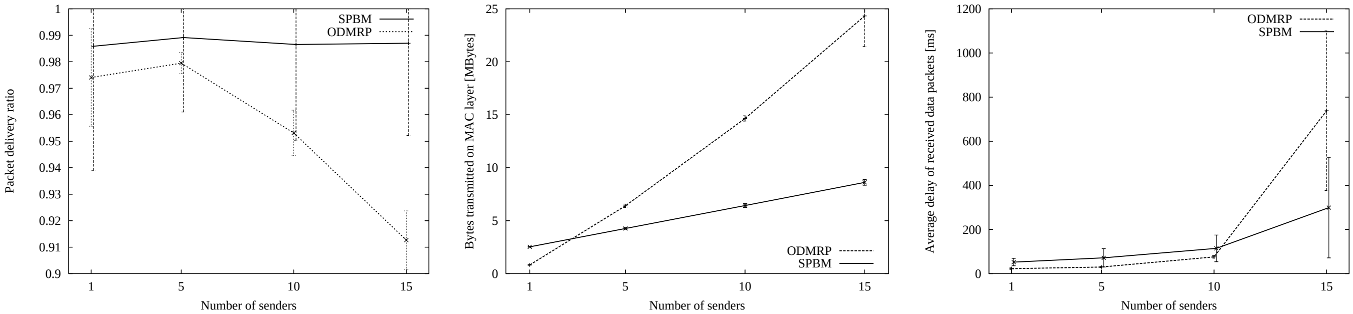
4. Simulations

4.1. Simulation setup

The simulations were performed using the network simulator *ns*-2 [11]. As a reference, the ODMRP implementation from [12] was chosen and ported to *ns*-2.27. Since we discovered some misbehaviors of this implementation, these were fixed: First, the calculation of the header sizes was corrected; second, the delay for join queries was changed to contain a constant part in addition to the random back-off. This reduces the overhead by about 10% without affecting the packet delivery ratio. Third, according to the ODMRP draft [7], duplicate join queries and join replies are suppressed, further reducing the overhead of the protocol.

The MAC layer in all simulations was IEEE 802.11 with a maximum bandwidth of 2 MBit/s and the transmission power resulted in a radio range of 250 meters. Since the transmitted packets were relatively small, the use of RTS/CTS was disabled. The modeled scenario was a square of 1400 meters by 1400 meters, where 196 randomly placed nodes (which corresponds to 100 nodes per square kilometer) moved according to the random way-point model [5] with a pause time of 10s and a minimum speed of 1 meter per second. The data payload had a size of 64 bytes per packet and each source transmitted one packet per second. All runs were simulated 20 times with different random seed values and movement scenarios, we report on the average of those runs. A run represented the simulated time of 180 seconds where nodes joined at the beginning of the simulation and the first data packet was sent after 60 seconds in order to give the group management enough time to initialize.

Some simulation parameters were varied to investigate their influence on the results. During each series of simulation runs, only one parameter was changed leaving the others constant. The number of senders ranged from 1 to 15, all senders and receivers did belong to one multicast group,



(a) Packet delivery ratio (b) Bytes transmitted on MAC layer (c) Delay

Figure 3. Performance w.r.t. number of senders (25 receivers, 1 Pkt/s, 5 m/s, 100 nodes/km^2)

but senders and receivers were disjoint. Mobility was varied from 0 to 15 meters per second.

The protocol specific parameters of SPBM were set as follows: the beacon interval was 2 seconds and a neighbor expired after 2.5 beacon intervals. The basic group-membership update frequency for an order-0 square f_0 was set to $\frac{1}{3\,s}$. The value for the timeout of entries in the member table was 2.5 times the corresponding update interval. The number of levels was set to 3 as in the example depicted in Figure 1 (i.e., $L = 3$).

ODMRP's protocol specific parameters were: a join refresh interval of 3 seconds, an acknowledgment timeout for join table messages of 25 milliseconds, and a maximum number of join table transmissions of 3.

To improve comparability, all these protocol specific parameters were kept constant throughout all simulations.

4.2. Performance metrics

The metrics used to evaluate the protocol performance are packet delivery ratio, overhead and delay. The *packet delivery ratio* (PDR) is defined as the sum of all unique data packets received divided by the sum of all data packets that should have been delivered (sum of sent packets multiplied by the number of receivers).

The *overhead* is the total number of bytes transmitted at the MAC layer, including acknowledgments in case of unicast transmissions. To measure the overhead on the MAC layer it is necessary to capture MAC layer retries induced by mobility or packet collisions. These effects would be invisible if the overhead was counted on the network layer.

The *delay* is defined as the time between sending a packet and the time the packet is successfully delivered. This value is averaged over all packets and all receivers.

4.3. Results

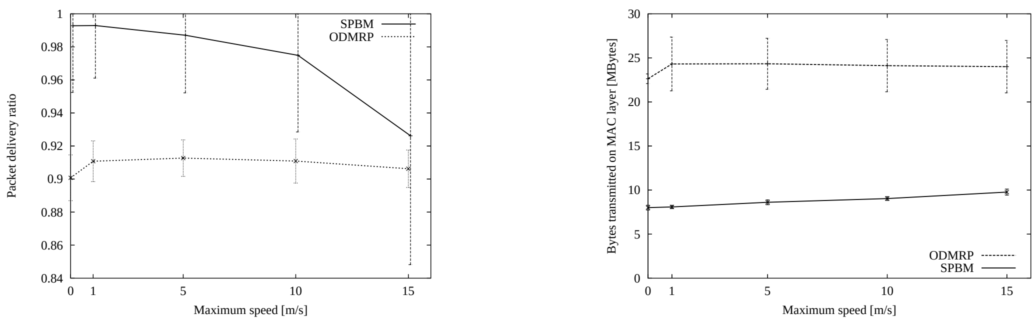
4.3.1. Number of senders

Figure 3 shows the respective PDR, overhead and delay when the number of senders increases. The other parameters were kept constant in this setup. While the PDR of SPBM is quite stable for different numbers of senders (up to 15 in these experiments), ODMRP suffers from the load generated by the additional senders. This is caused by the fact that each sender floods the whole network with data and control packets at regular intervals in order to build its forwarding group. The group management of SPBM is independent of the existing multicast sources. If only one sender is active, the network load induced by ODMRP is lower than in SPBM. This is caused by the fact that the proactive group management of SPBM is responsible for a certain constant overhead. For ODMRP, the high increase in load is accompanied by a high decrease in the ratio of delivered packets.

SPBM, in contrast, sustains a satisfactory packet delivery ratio. The increase in overhead is mainly due to the increased number of data forwarding operations for the data packets of the additional senders. The proactive group management overhead of SPBM remains constant, while the number of neighborhood beacons decreases. This is caused by the use of implicit beaconing where beacon information is prepended to data packets whenever possible.

A similar result was achieved when varying the number of receivers while keeping the number of senders constant. In this case, ODMRP quickly saturates the network resulting in a constantly high network load, while SPBM still operates with a satisfactory packet delivery ratio with a load increase mainly caused by the higher number of forwarding operations.

Regarding the end-to-end-delay (Figure 3(c)), the results show that ODMRP performs slightly better than SPBM for a small number of senders. Since ODMRP's forwarding al-



(a) Packet delivery ratio (b) Bytes transmitted on MAC layer

Figure 4. Performance w.r.t. maximum movement speed (15 senders, 1 Pkt/s, 25 receivers, 100 nodes/km^2)

gorithm is a form of scoped flooding and the delay is measured as the first copy of a certain packet arrives, ODMRP is able to use the direct route from source to each destination. At the same time the overhead introduced through the scoped flooding leads to a steep increase in the delay once the network becomes saturated due to the increase in senders.

4.3.2. Node mobility

Figure 4 shows the impact of node mobility on the packet delivery ratio and the bytes transmitted on the MAC layer.

While SPBM performs very well for low to medium node mobility the packed delivery ratio drops significantly for high node speeds. This was surprising for us, given that position-based routing is commonly considered to be very robust to topological changes. Further investigation revealed two reasons for this behavior: (1) When group members cross square "boundaries" into a square that did not previously contain a group member, they will not receive packets until the group management scheme has spread the new information. (2) When node mobility increases, forwarding failures appear that are induced by discrepancies in the neighbor table used for the next-hop selection. If a node is selected as a forwarder but moved out of radio range, the current forwarder has to wait for four unsuccessful retransmissions followed by a link layer notification before it is able to select a different node[4]. This reduces the packet delivery ratio and increases the amount of MAC packets transmitted. To avoid this problem we have conducted first experiments to adapt the ideas of contention-based forwarding, as described in [4], to SPBM. Furthermore we specifically managed situations where nodes crossed square boundaries. These experiments indicated that the modifications will make both the delivery rate and the number of transmitted packets largely independent of node mobility.

[4]This effect has been extensively described in [4].

5. Conclusions and future work

In this paper we described a novel position-based ad-hoc multicast routing protocol. It differs significantly from previous work in that it introduces a hierarchical organization of nodes for membership management as well as packet forwarding. By means of simulation we demonstrated that SPBM performs very well in particular as the number of multicast senders and receivers increases.

Our main priority for future work is the integration of contention-based forwarding and a management scheme for nodes crossing square boundaries into SPBM. First results show that these mechanisms are able to eliminate the impact of very high node mobility on the performance of SPBM.

To summarize, we believe that a hierarchical approach to position-based multicast is a very promising solution if the protocol is intended to scale to a reasonable number of nodes.

References

[1] S. Basagni, I. Chlamtac, and V. R. Syrotiuk. Location aware, dependable multicast for mobile ad hoc networks. *Computer Networks*, 36(5–6):659–670, August 2001.

[2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc Wireless Networks. In *Proc. of ACM DIAL-M '99*, pages 48–55, Seattle, WS, August 1999.

[3] K. Chen and K. Nahrstedt. Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET. In *Proc. of IEEE Infocom '02*, pages 1192–1201, New York City, New York, June 2002.

[4] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein. Contention-Based Forwarding for Mobile Ad-Hoc Networks. *Elsevier's Ad Hoc Networks*, 1(4):351–369, 2003.

[5] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, volume 353, pages 153–181. Kluwer Academic Publishers, 1996.

[6] B. N. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. of ACM MobiCom '00*, pages 243–254, Boston, Massachusetts, August 2000.

[7] S.-J. Lee, W. Su, and M. Gerla. On-Demand Multicast routing Protocol (ODMRP) for Ad Hoc Networks. Internet Draft, draft-ietf-manet-odmrp-02.txt, work in progress, 2000.

[8] M. Mauve, H. Füßler, J. Widmer, and T. Lang. Position-Based Multicast Routing for Mobile Ad-Hoc Networks. Technical Report TR-03-004, Department of Computer Science, University of Mannheim, 2003.

[9] M. Mauve, J. Widmer, and H. Hartenstein. A Survey on Position-Based Routing in Mobile Ad-Hoc Networks. *IEEE Network*, 15(6):30–39, November/December 2001.

[10] J. Nonnenmacher and E. W. Biersack. Scalable Feedback for Large Groups. *IEEE/ACM Transactions on Networking (TON)*, 7(3):375–386, 1999.

[11] The ns-2 network simulator. http://www.isi.edu/nsnam/ns/.

[12] Wireless multicast extensions for ns-2.1b8. http://www.monarch.cs.rice.edu/ multicast_extensions.html.

[13] M. Transier, H. Füßler, J. Widmer, M. Mauve, and W. Effelsberg. A Hierarchical Approach to Position-Based Multicast for Mobile Ad-hoc Networks. Technical Report TR-04-002, Department for Mathematics and Computer Science, University of Mannheim, 2004.