Buckshot Routing - A Robust Source Routing Protocol for Dense Ad-Hoc Networks

David Peters, Reinhardt Karnapke, and Jörg Nolte

Brandenburg University of Technology Cottbus
`{dpeters,karnapke,jon}@informatik.tu-cottbus.de`,
WWW home page: `http://www-bs.informatik.tu-cottbus.de/`

Abstract. Experiments with wireless sensor networks have shown that unidirectional communication links are quite common. What is even more, they have also shown that the range of a unidirectional link can exceed that of a bidirectional one by far. Still, most of todays routing protocols do not use them, they only eliminate their implications. Those protocols that do use unidirectional links introduce a lot of protocol overhead. In this paper we present *Buckshot Routing*, a robust, yet simple source routing protocol for dense wireless networks with lossy or unidirectional links, which reduces the overhead generated by route discovery and route maintenance significantly.

Key words: wireless sensor networks, source routing, robustness, dense networks

1 Introduction

Routing protocols for wireless ad-hoc networks try to optimize the packet delivery ratio while keeping the protocol overhead as low as possible. One of the problems that is extremely costly for routing protocols is the presence of unidirectional links. For routing protocols that can explicitly use them, they present an advantage due to shorter routes and more links leading to stronger connected communication graphs. Finding these links, informing the uplink node of the existence and keeping track of the changes is extremely expensive, however. Due to this fact, a lot of routing protocols work only on bidirectional links. In our opinion, this is the wrong way of addressing the problem. Unidirectional links can have a much longer range than bidirectional ones (see section 2), which could be used to reduce the number of hops needed and thus the overall energy consumption in the network. As the length of a route is a significant factor in the reliability of a whole path, this also increases the robustness of the routing protocol.

In this paper we present Buckshot Routing (BR), a robust source routing protocol that uses unidirectional links. BR is a modification of DSR [8], the dynamic source routing protocol. In contrast to DSR, BR does not need to flood the whole network a second time to find the reverse route (see section 5), but rather uses a limited directional flooding, which leads to multiple paths being used.

This paper is structured as follows: Section 2 describes some experiments which show the commonness of unidirectional links followed by section 3 which shows their impact on common routing protocols in detail. The Buckshot Routing protocol is shown in section 4 and evaluated in section 5. Related work is given in section 6. We finish with conclusion and future work in section 7.

2 The Nature of Unidirectional Links

In theory a unidirectional link is defined quite simple. A link from node A to node B is unidirectional, if Node B can receive messages from A, but not vise versa. In practice, it is fairly hard to establish such criteria. It is not possible to monitor the status of all links globally. You can only measure the status of a link at a certain time. Moreover, only one direction of the link can be measured because transceivers can not transmit and receive at the same time. Worse still, links change over time. A link that seems to be bidirectional at one moment can change at any time.

The authors of [16] describe an experiment they conducted in the Lüneburger Heide. The original aim was to evaluate a routing protocol, which is not characterized further in the paper. Rather, the observations they made concerning the properties of the wireless medium are described, focusing on the frequency of changes and the poor stability of links. These experiments were conducted using 24 Scatterweb ESB [15] sensor nodes, which were affixed to trees, poles etc, and left alone for two weeks after program start. One of the duties of the network was the documentation of the logical topology (radio neighborhood of nodes), which was evaluated by building a new routing tree every hour, e.g. for use in a sense-and-send application. The neighborhood was evaluated using the Wireless Neighborhood Exploration protocol (WNX) [16], which can detect unidirectional and bidirectional links. Once this was done, all unidirectional links were discarded and only the bidirectional ones were used to build the routing tree. Figure 1a shows one complete communication graph obtained by WNX, while figure 1b shows the same graph without unidirectional links, where a lot of redundant paths have been lost by the elimination. In fact, one quarter of the nodes are only connected to the rest of the network by a single link when unidirectional links are removed. If this single link breaks, the nodes become separated, even though there are still routes available. Thus, the removal of unidirectional links increases the probability of network separation severely.

In experiments with XSM motes [14] 7 times 7 nodes were placed in a square, with a distance of about 1 meter between nodes. In four sets of experiments at different times of day each node sent 100 messages at three different power levels. Then the packet reception rate was recorded. It is defined for a node A as the number of packets A received from a node B divided by the number of messages sent (100). Then the packet reception rates of nodes A and B are compared. If the difference is less than 10%, the link is considered bidirectional. If it is more

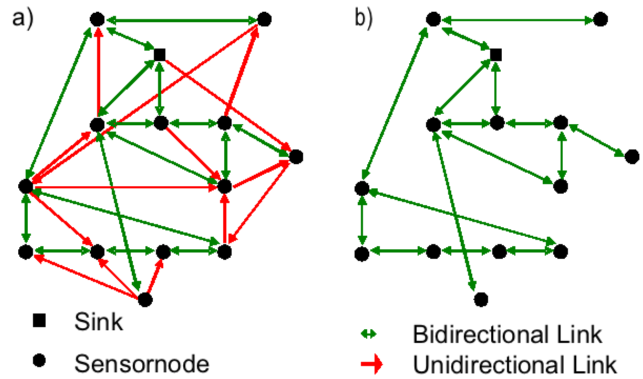

Fig. 1. A Communication Graph from [16] (presentation) [17]

then 90% the link is considered unidirectional. The XSM nodes offer 9 different transmission strengths, of which three were evaluated: the lowest, the highest and the third in between. Table 1 shows the results of the experiments.

Table 1. Link Quality versus Transmission Strength

PRR	less than 10%	10-90%	more than 90%	number of links
power level 1	50%	43%	7%	500
power level 3	65%	22%	13%	1038
power level 9	88%	6%	6%	1135

The results show that even when using the maximum transmission strength 12% of the links would have been discarded by ETX (Expected Transmission Count) [4] and similar link quality evaluation protocols that focus only on bidirectional links. As the lifetime is one of the major optimization goals in a sensor network and receiving/transmitting consumes a lot of energy, it is rather uncommon to have all nodes constantly transmit using the highest transmission strength. In fact, current research projects like e.g. [9] try to minimize power consumption by adjusting the transmission strength depending on the required reach and reliability.

The observations of [14] are concluded in three points:

1. Wireless links are often asymmetric, especially if transmission power is low
2. Dense networks produce more asymmetric links then sparse ones
3. Symmetric links only bridge short distances, while asymmetric and especially unidirectional ones have a much longer reach. A conclusion drawn from this is that the usage of unidirectional links in a routing protocol can increase the efficiency of a routing protocol considering energy and/or latency.

A sensor network which monitors water pumps within wells is described in [5]. The sensors were used to monitor the water level, the amount of water taken and the saltiness of the water in a number of wells which were widely distributed. The necessity for this sensor network arose because the pumps were close to shore and a rise in saltiness was endangering the quality of the water. The average distance between wells was 850 meters and the range of transmission was about 1500 meters. Communication was realized using 802.11 WLAN hardware both for the nodes as well as for the gateway. For data transmission between nodes Surge_Reliable [19] was used, which makes routing decisions based on the link quality between nodes.

During the experiments the authors observed, that the (logical) topology of the network changed dynamically, even though all nodes were stationary. The authors claim that these changes were probably due to antenna size and changes in temperature and air moisture. In this context it is important to remember that the distance of nodes was far below the range of the transmitters (about 50%). While about 70% of the routing trees observed followed the theory, there were a lot of strange exceptions. In one case the average distance between connected nodes even rose to 1135 meters, as nodes that should have been able to communicate directly with the gateway were connected to nodes on the far side instead. In one of these routing trees a single node had to take care of all communication with the gateway, even nodes that were on the other side were using it as next hop. The reason for this is that Surge_Reliable chooses the nodes with the best link quality, but only considers bidirectional links. If unidirectional links could have been used, the results could have been quite different.

VigilNet, a military sensor network for terrain surveillance is described in [6]. This projects aims at the detection of moving vehicles using magnetic sensors attached to Mica2 sensor nodes. The transport of messages from the nodes to the sink was realized using a diffusion based algorithm, similar to Directed Diffusion [7], which produced a routing tree with root at the sink. To eliminate unidirectional links, a protocol called Link Symmetry Detection was developed. Each node periodically transmitted the list of its neighbors. A node that received such a neighbor list checked the list to determine if it was mentioned. If it was not, the link was an incoming unidirectional one. When building the routing tree after deployment, the transmission power of all nodes was halved. Now all nodes determined their parent node from the neighbor lists received with this half strength. At the end of this setup phase, all nodes switched to full transmission power. The intention behind this scheme was to ensure that the connection to the father node would not break. During the experiments, the authors noted that asymmetric links were far more common than expected. They put this fact down to differences in hardware, as the transceivers were not calibrated before the experiment.

3 Impact on the Routing Layer

Most existing routing protocols are built for bidirectional links. A common way to detect a route from one node to another is to flood a message into the network. This can be called a Route Request Message (RREQ) in protocols like Dynamic Source Routing (DSR) [8] or Ad-Hoc On Demand Distance Vector Routing (AODV) [11]. In other protocols like Directed Diffusion[7] the flooded messages are called Interests. The basic mechanism is the same, though. Once the destination is reached, a message is sent back along the inverted path. In agent based protocols like Rumor Routing [2] the network is not flooded. Instead, an agent is sent which travels through the net using a random walk pattern. Still, the assumption that all links are bidirectional is the same. When an agent which was sent because of a certain event reaches a node, this node remembers the event and a route leading to this event, which is the inversion of the route the agent has traveled.

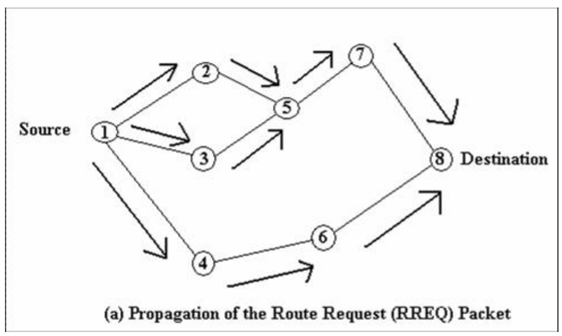

Fig. 2. Flooding of a Route Request Message

Figure 2 shows the propagation of a RREQ message as used in AODV. The source, node 1, wants to transmit to node 8, the destination. As it does not know a path to node 8, it floods the network with a RREQ message which reaches the destination through multiple paths. Once a RREQ message reaches the destination, it sends a RREP back the inverted route. In the example the message from node 6 arrives first, and gets answered with a RREP (figure 3). All other RREQ messages that arrive later are ignored.

In networks with stable bidirectional links these protocols provide good results. But when only one unidirectional link exists within the path the RREQ takes, the RREP will be lost. Even worse, due to the fact that only the first RREQ is answered, all further RREQ messages that arrive at the destination are ignored, any bidirectional path that may exist will be ignored. When the source does not receive a RREP message after a certain time, it will restart the route discovery with the same result, only increasing network traffic but never finding a path.

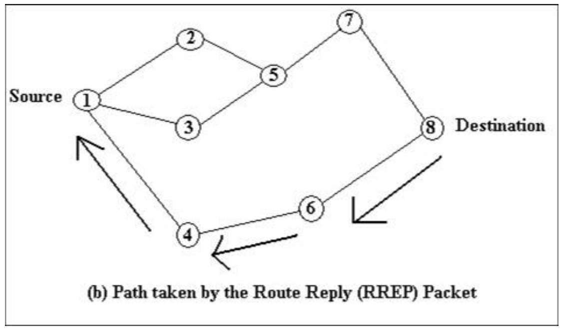

Fig. 3. Transmission of a Route Reply

This problem is increased by the characteristics of unidirectional links observed in section 2. As observed above, the range of unidirectional links is often far greater than that of bidirectional ones, leading to fewer hops needed to reach the destination. Thus, the messages forwarded over unidirectional links will often arrive earlier than those using the bidirectional ones.

A lot of routing protocols cope with this problem by eliminating the unidirectional links, e.g. [11, 10, 8]. This elimination can be done e.g. by blacklisting as in AODV, or by requesting explicit acknowledgments, which is possible in DSR. There are also protocols which enable the usage of unidirectional links. Some, by finding one way from source to destination and another one from destination to source as in DSR, others by providing an abstraction between MAC and routing. This abstraction can use multiple hops as return path from an unidirectional link, and present the routing protocol with a network consisting only of bidirectional links like the sub routing layer [13]. However, all these protocols introduce a significant overhead.

4 The Buckshot Routing Protocol

Buckshot routing is based on the source routing principle also used in DSR. When a node A wants to send a message to a node B it looks up the path in its routing table. If there is an entry, the whole route is attached to the message and the message is transmitted. If no entry is found, a route discovery is started. The big difference to DSR lies in the way messages are forwarded. In BR all nodes remember their neighboring nodes in a neighborhood table. This table is maintained without additional communication overhead, simply by listening to the medium and recording the last hop of all received messages. When a node receives a message that is not addressed to itself, it looks into the header to identify the node after the next. If that node can be found in this nodes neighborhood table, the message is forwarded.

Figure 4 shows an example of the workings of what we call *pseudobidirectional links*. The node in the middle of the left side of the figure wants to send

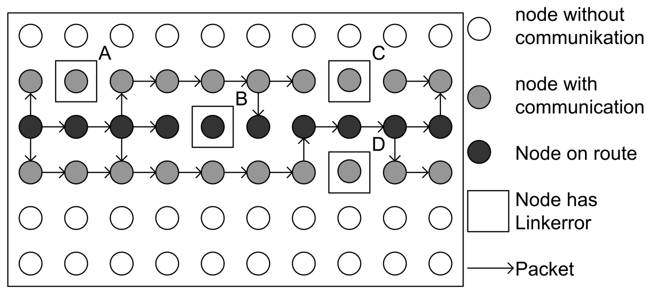

Fig. 4. Message Forwarding in Buckshot Routing

a message to the one in the middle of the right side. The route has been established previously, it leads through all black nodes. The route might be old or the network changes often, for whatever reason, the 4 nodes surrounded by a square (A-D) are not able to receive and/or transmit messages. As one of them is a black node (B) and lies on the path, sending of the message would normally have failed. But the nodes that are above and below node B in the picture forward the message, because they know the next but one hop, the node to the right of node B. This way, the message takes a detour and returns to the right path. Please note that it does not matter if the node failed permanently, had a momentarily interruption of its link or was only reached by a unidirectional link in the other direction.

4.1 Lossy or Unidirectional Links

DSR can work in the presence of unidirectional links. When specified, DSR does not use the inverted route of the RREQ packet as way from destination back to the source. Rather, a new route discovery is started. The new RREQ packet that is flooded into the network contains the route from source to destination and discovers the way back. This flooding of the enlarged RREQ packet into the whole net presents a significant overhead, which BR tries to reduce.

In BR the first step of the route discovery, the flooding of the RREQ, is identical to DSR. But when the destination received the message, it simply sends a route reply along the inverted path of the RREQ. Unidirectional links on this route, or lossy links are masked by the *pseudobidirectional links*. Due to the use of the forwarding mechanism described above, the neighbors of the nodes with the unidirectional link take over and forward the message to the next but one hop. This whole mechanism is similar to a limited directional flooding, which reduces the number of forwarding nodes compared to DSR. Also, no second RREQ message is needed.

5 Evaluation

For our evaluation we used the discrete event simulator OMNeT++ [18] as well as real sensornet hardware (TMote Sky sensor nodes). All evaluations compare BR with DSR.

5.1 OMNeT++

All simulations used the node layout shown in figure 5. Three different densities were used, with 4 (barely dense network), 8 (medium dense network) or 20 (dense network) neighbors for each node.

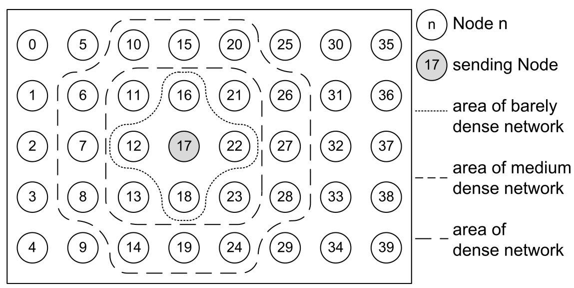

Fig. 5. Different Densities of Networks

Network Density In this row of simulations the basic differences between DSR and BR are shown, as well as the dependency of BR on the density of the network. To evaluate this, node 2 from picture 5 tried to find a route to node 37 and then transmit a data packet. For each experiment, a different probability of packet loss was used.

The results are shown on figure 6. The x-axis shows the probability of packet loss for each hop, while the y-axis shows the probability of successful delivery of the data packet from node 2 to node 37. For the barely dense network, where each node has only 4 neighbors, BR performs worst. This can be easily explained by the fact that it could only use the original route - there were never any nodes that received a message and had the next but one hop in their neighborlist. For the medium dense network BR was able to deliver all messages even at a probability of packet loss of 25%. In the dense network BR was able to make full use of its redundant forwarding scheme and managed to deliver all messages at a link failure probability of over 50%.

As these results show, it is clearly interesting to evaluate BR further, even though only for the medium dense and dense networks. For barely dense networks or sparse networks it is simply not suitable and another protocol should be used.

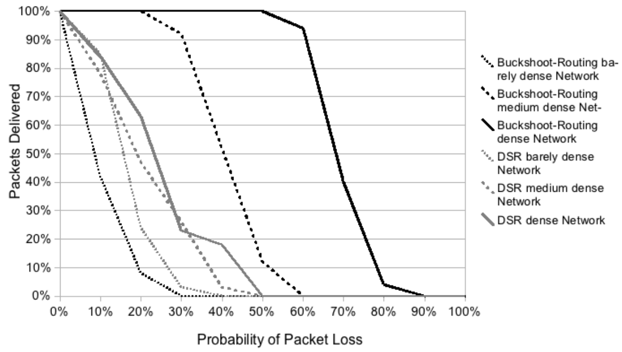

Fig. 6. Probability of a Successful Communication

Scenario with Temporary Changes In these experiments the robustness against temporary changes was evaluated for both protocols. Node 2 once again sent a data packet to node 37, but this time it expected an answer. In the time between the first and second data packet, the logical topology of the network (link connectivity) could change.

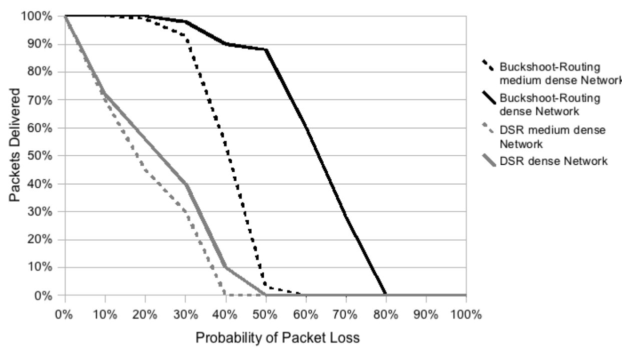

Fig. 7. Successful Transmissions for Temporary Changes

Figure 7 shows the probability of a reception of the answer by node 2 depending on the probability of link failure. All results depicted here are worse than the ones shown in the network density simulation, but this is to be expected as the total number of hops increased by far. Moreover, the figure also shows that BR still outperforms DSR.

The price that has to be paid for the increased number of delivered packets is shown in figure 8. For less than 10% probability of link failure, BR transmits less packets than DSR in the dense network, in the medium dense network the break even point is at roughly 20%. Once the probability of packet loss gets

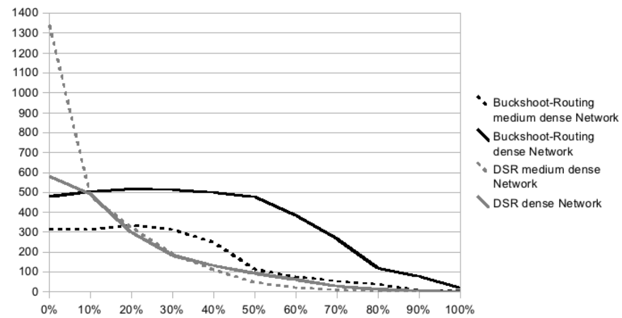

Fig. 8. Number of Packets transmitted

higher, BR transmits up to twice as many packets as DSR. When correlating this number of packets to the probability of a successful communication though, it can be concluded that when robustness is a must, the choice should always be BR.

Scenario with Permanent Changes In this scenario the robustness against permanent changes was evaluated. Of the 40 nodes in the simulation, one after another died and did not communicate anymore, the only ones that could not die were the source (node 2) and destination (node 37). The source transmitted 50 data packets and the destination tried to reply.

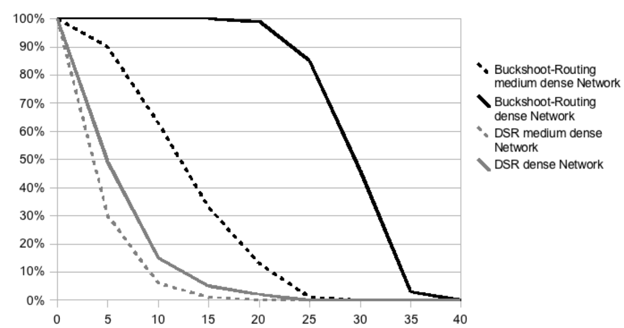

Fig. 9. Successful Transmission for Permanent Changes

Figure 9 shows the number of successful communications depending on the number of nodes that died. For DSR the difference between the medium dense and the dense network is minimal. BR on the other hand was once again able to use its advantage to its full extend, and even managed a successful communication after 35 nodes died.

The number of transmitted packets is shown in figure 10. For the dense network, BR always transmits at least twice as many messages as DSR. In the

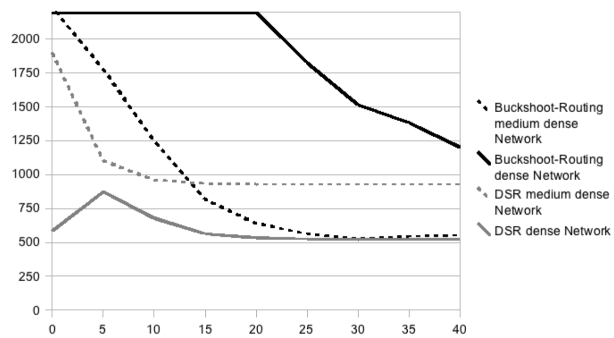

Fig. 10. Transmitted Packets

medium dense network the number of packets sent by BR is less than that of DSR after the death of 14 nodes. The reduction of the number of messages transmitted by DSR is correlating to the decrease in successful communications, though.

Another interesting figure is the amount of RREQ messages transmitted by each protocol. Figure 11 shows the number of RREQ messages divided by the number of total packets sent for each protocol, depending on the number of nodes that died. DSR reaches 90% RREQ messages pretty soon, while BR stays below 10%. This shows that the relative protocol overhead for BR is far lower than the one of DSR.

5.2 TMote Sky

In the real experiments TMote Sky sender nodes were used (see figure 12). There was only a simple MAC Protocol that delayed a message for a random time and then checked whether the medium was free before transmitting. To enable a

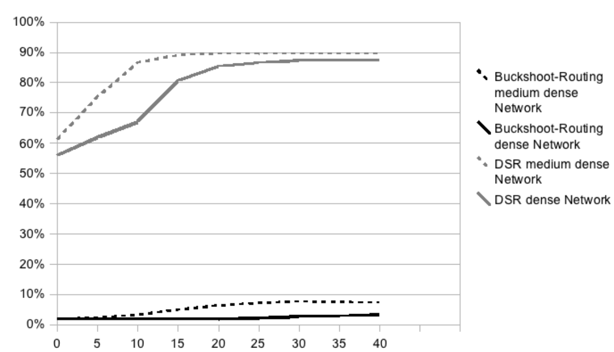

Fig. 11. Percentage of Route Request Messages

multihop scenario, the transmission power of the TMotes was reduced to its minimum.

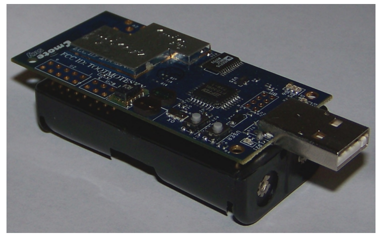

Fig. 12. A TMote Sky Sensor Node

Rectangle Scenario For this row of experiments 16 TMotes were placed on the floor of a building in a rectangle. The nodes were placed in a 4 times 4 grid, with a distance of 190 cm between nodes on one axis and 60 cm on the other. This distance was chosen to enable each node to communicate only with its direct neighbors. The differences in distance are due to the properties of the used antennae. The nodes were restarted after each experiment. For both protocols, DSR and BR, 50 experiments were conducted. In each experiment the source node (node Q) tried to first establish a route to the destination (node Z) and then transmit a data packet. The experiment was successful when the data packet was received by node Z. Figure 13 shows the number of received messages for each node, separated for both protocols.

Even in such a small network, the packet loss rate was high. BR managed a successful transmission in 90%, while DSR reached only 28%. For this reason, we made another row of experiments, with which the maximum number of hops each protocol can manage was to be determined.

Determining the Longest Possible Route To determine the longest possible distance each protocol could manage, 23 sensor nodes were used. They were

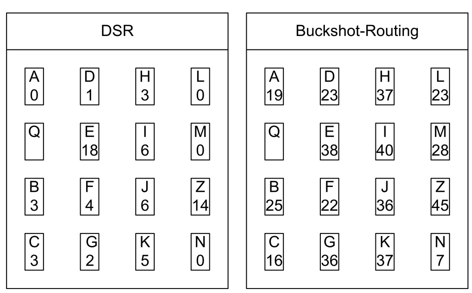

Fig. 13. Received Messages in the Rectangle Scenario

placed nearly in a rectangle of 3 times 8, only one space directly next to the source was left open due to previous hardware failure and the lack of replacement nodes (figure 14).

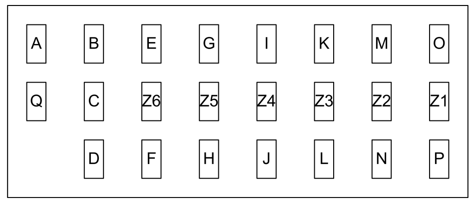

Fig. 14. The Layout of the Sensor Nodes in one of our Experiments

The nodes were once again placed on the floor of a building, set to the lowest transmission power and the distance between nodes was again 190 cm and 60 cm respectively. For each routing protocol and each destination (nodes Z1 to Z6) the source (node Q) tried 10 times to establish a route and transmit a data packet. The number of received data packets for each node is shown in figure 15. The

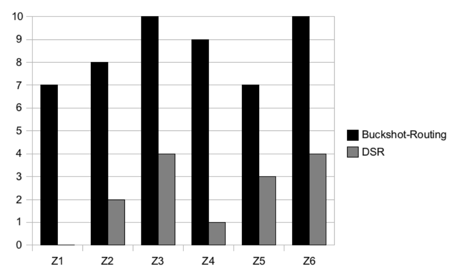

Fig. 15. Successful Transmissions

figure shows that BR outperforms DSR for every destination, even the one that is only 2 hops distant. BR delivered between 7 and 10 packets for each node, while DSR delivered between 1 and 4. For the farthest node, Z1, DSR was not able to deliver at all, while BR still managed 7 successful transmissions. This is of course not only due to the routing protocols. A part of the blame for the bad results of DSR has to be assigned to the MAC protocol. Sometimes, DSR tried to repair routes and the protocol packets have probably collided with the data packets. But this also shows that BR is less prone to problems from the MAC layer than DSR.

6 Related Work

The authors of [3] analyze the performance of the full and the partial link reversal algorithm. These algorithms build a directed acyclic graph (DAG) that leads from any node to a fixed destination. If there is more than one destination, one graph has to be built for each of them. When a node has to transmit a message, it simply sends it on any of its outgoing links for that destination. This way, the used route is not always the shortest, but the message will reach the destination eventually. When links break, nodes that have no more outgoing links reverse all of their incoming ones (full reversal) or only a certain subset of them (partial reversal). This may of course result in a reversal in neighboring nodes, too. The results of the analysis show that the full reversal algorithm is asymptotically optimal, while the partial one is not. While link reversal routing may be able to work with unidirectional links and is efficient for n to 1 communication, the fact that a separate DAG is needed for each destination disqualifies it for any to any communication. Realizing any-to-any routing with link reversal algorithms would result in as many DAGs in each node as there are nodes in the network. This represents to much data storage, computational overhead and communication cost.

The Loop Based Source Routing protocol (LBSR) [1] is based on DSR [8] and designed to use unidirectional links in the routing process. LBSR eliminates the need for the flooding of the net by the destination as used in DSR. Instead of building a route from the source S to the destination D and then vise versa, routing loops are created. A so called Lreq message is flooded into the net by S. Every node that receives this Lreq for the first time rebroadcasts it after attaching its ID. If an Lreq is received by S, a loop has been found. S now knows a route to each node whose ID is enclosed in the Lreq. It then unicasts a packet along that route which enables all nodes along this route to send packets to each other and to S by following the enclosed route. If a node that is already part of a loop receives a message from a node that is not its predecessor in the loop, it adds its ID to the Lreq and forwards it along its loop. The evaluation of LBSR focuses on the number of floodings and messages transmitted. The number of floodings is naturally only half as high, but the total number of sent messages is higher. The fact that a loop has to be passed by two messages costs time, energy and bandwidth. The broadcast character of wireless networks, i.e. that every node receives unicast messages even if they are not addressed to it, has been ignored completely. The main advantage pointed out by the authors is that the number of entries in the route cache is many times as large as in DSR. The main disadvantage is that the messages become arbitrarily large and have to pass each loop at least two times, thus creating a huge network load.

The author of [12] proposes a routing protocol based on distance vector routing as used in AODV [11] or DSDV [10] which is able to work in the presence of unidirectional links. In this protocol, every node in the network exchanges beacons with its neighbors periodically. Each node contains a Neighborlist called *Nodesheard* generated by the beacons and a matrix of dimension n^2 called D, where n is the number of nodes in the network and every entry consists of a tuple (sequence number, distance). The node also maintains two vectors called *To* and *From* in which the sequence number, distance and next hop are saved for each destination to which this node can send(*To*) and source, which sends to this node (*From*). The beacons transmitted periodically include *Nodesheard*, which is used to distinguish between unidirectional and bidirectional nodes. The matrix D is also transmitted periodically to build routes. This is not done as often as sending *Nodesheard* for performance reasons, as its size is much larger. This periodic, proactive transmission is also the main disadvantage of the protocol. The Matrix D stored on each node and transmitted periodically means that $O(n^2)$ space is needed on each node and, even worse, messages of the same size have to be transmitted. This is a major drain on energy and bandwidth.

7 Conclusion and Future Work

In this paper we presented a new source routing protocol. BR is a robust source routing protocol for dense, lossy wireless sensor networks. We have shown its advantages both in simulations and in real experiments. The evaluation has shown that even though it theoretically produces a large overhead because of the thickening of routes, in reality it transmits even less messages than DSR. This is due to the fact that DSR needs to find new routes much more often then BR.

In the future we would like to apply the concept used in BR to distance vector protocols. This would make some modifications of the distance vector routing protocol necessary, because the knowledge of the next but one hop would no longer be carried in the message. Rather, each node would have to keep track of that.

References

1. T. Asano, H. Unoki, and H. Higaki. Lbsr: Routing protocol for manets with unidirectional links. In *18th International Conference on Advanced Information Networking and Applications (AINA'04) Volume 1*, page 219, 2004.
2. D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 22–31, New York, NY, USA, 2002. ACM Press.
3. C. Busch, S. Surapaneni, and S. Tirthapura. Analysis of link reversal routing algorithms for mobile ad hoc networks. In *SPAA '03: Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*, pages 210–219, New York, NY, USA, 2003. ACM Press.
4. D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146, New York, NY, USA, 2003. ACM.
5. T. L. Dinh, W. Hu, P. Sikka, P. Corke, L. Overs, and S. Brosnan. Design and deployment of a remote robust sensor network: Experiences from an outdoor water

quality monitoring network. In *LCN '07: Proceedings of the 32nd IEEE Conference on Local Computer Networks*, pages 799–806, Washington, DC, USA, 2007. IEEE Computer Society.

6. T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, and B. Krogh. Vigilnet: An integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sen. Netw.*, 2(1):1–38, 2006.
7. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, 2003.
8. D. Johnson, D. Maltz, and J. Broch. *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*, chapter 5, pages 139–172. Addison-Wesley, 2001.
9. R. Min and A. Chandrakasan. A framework for energy-scalable communication in high-density wireless networks. In *ISLPED '02: Proceedings of the 2002 international symposium on Low power electronics and design*, pages 36–41, New York, NY, USA, 2002. ACM.
10. C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244, New York, NY, USA, 1994. ACM Press.
11. C. E. Perkins and E. M. Royer. "ad hoc on-demand distance vector routing.". In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA*, pages 90–100, FEB 1999.
12. R. Prakash. A routing algorithm for wireless ad hoc networks with unidirectional links. *Wirel. Netw.*, 7(6):617–625, 2001.
13. V. Ramasubramanian, R. Chandra, and D. Mosse. Providing a bidirectional abstraction for unidirectional ad hoc networks, 2002.
14. L. Sang, A. Arora, and H. Zhang. On exploiting asymmetric wireless links via one-way estimation. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 11–21, New York, NY, USA, 2007. ACM Press.
15. J. Schiller, A. Liers, H. Ritter, R. Winter, and T. Voigt. Scatterweb - low power sensor nodes and energy aware routing. In *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005.
16. Turau, Renner, and Venzke. The heathland experiment: Results and experiences. In *Proceedings of the REALWSN'05 Workshop on Real-World Wireless Sensor Networks.*, Jun 2005.
17. V. Turau. The heathland experiment: Results and experiences, presentation. www.tm.uka.de/forschung/spp1140/events/kolloquium/2005-11/turau.pdf.
18. A. Varga. The omnet++ discrete event simulation system. In *Proceedings of the European Simulation Multiconference (ESM'2001)*, Prague, Czech Republic, June 2001.
19. A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27, New York, NY, USA, 2003. ACM Press.