Sensors & Transducers
© 2015 by IFSA Publishing, S. L.
http://www.sensorsportal.com

Buckshot Routing with Distance Vectors in Three Application Scenarios for Wireless Sensor Networks with Unstable Network Topologies and Unidirectional Links

Reinhardt Karnapke, Jörg Nolte
Distributed Systems/Operating Systems Group
Brandenburg University of Technology Cottbus - Senftenberg
Cottbus, Germany
E-mail: Karnapke@informatik.tu-cottbus.de, Jon@informatik.tu-cottbus.de

Received: 14 November 2014 /Accepted: 15 January 2015 /Published: 28 February 2015

Abstract: Experiments have shown that the number of asymmetric and unidirectional links often exceeds the number of bidirectional ones, especially in the transitional area of the communication range of wireless sensor nodes. Still, most of today's routing protocols ignore their existence or try to remove their implications. Also, links are not stable over time, and routes become unusable often, resulting in a need for new routing protocols that can handle highly dynamic links and use unidirectional links to their advantage. At SENSORCOMM' 2014, we presented BuckshotDV, a routing protocol which is resilient against link fluctuations and uses the longer reach of unidirectional links to increase its performance. Furthermore, its distance vector nature makes it scalable for large sensor networks. This paper is an extended version which adds some implementation details and the evaluation of BuckshotDV in two more application scenarios. *Copyright © 2015 IFSA Publishing, S. L.*

Keywords: Wireless sensor networks, Routing, Unidirectional links, Buckshot routing, Distance vectors.

1. Introduction

This is an extended version of the paper called 'BuckshotDV - A Robust Routing Protocol for Wireless Sensor Networks with Unstable Network Topologies and Unidirectional Links' which was presented at SENSORCOMM' 2014 [1].

In recent years, asymmetric and unidirectional links have been shown to be common in wireless sensor networks. Depending on the used hardware and the distance between nodes, different regions (transitional region [2], grey area [3]) have been defined, in which unidirectional links are common and can even represent the majority of links. Also, most links are not stable over time [4].

In traditional routing protocols, unidirectional links and unstable links are ignored and not used for forwarding purposes. Bidirectional, stable links make routing decisions much easier. Unfortunately, this approach neglects a lot of potential optimizations, as unidirectional links often have a greater reach than bidirectional ones. Thus, unidirectional links reduce the number of hops needed to deliver a message to its destination. However, using unidirectional links is often considered to induce too much overhead [5]. An example for this overhead is the need to inform upstream nodes of their outgoing links.

In this paper, we present BuckshotDV, a routing protocol specifically designed to use unidirectional links implicitly. The overhead which results from the need to inform upstream neighbors of their outgoing unidirectional links in other protocols is eliminated. BuckshotDV is based on a multi path approach, enabling the usage of unidirectional links and making it resilient against link changes and node failures. Moreover, a node implicitly updates its routing table each time a message is received.

The remainder of this paper is structured as follows: the nature of unidirectional links and their commonness in wireless sensor networks are presented in Section II. Selected state of the art routing protocols that were used in the evaluation are presented in Section III, followed by the description of our protocol BuckshotDV in Section IV. In Section V the evaluation of BuckshotDV and selected state of the art protocols in simulations and real experiments is shown before concluding remarks are given in Section VI.

2. Unidirectional Links in Wireless Sensor Networks

Different classifications of link quality are used in literature. Examples are included in [2-4, 6], which all use different classifications (see below).

The most commonly used classifications divide links into bidirectional links, asymmetric links and unidirectional links. A bidirectional link is always defined as a link between two nodes which can be used to transmit a message from either of those two nodes to the other one. In contrast, the terms asymmetric link and unidirectional link are not always defined clearly, and sometimes used synonymously. Common definitions for asymmetric links focus on a variation of either Received Signal Strength Indication (RSSI) values or packet loss (delivery ratio). When the delivery ratio is used, unidirectional links can be seen as a subclass of asymmetric links where the delivery ratio in one direction is 0. However, this definition requires quite a lot of message transmissions in order to evaluate the delivery ratio. For this paper, a unidirectional link is defined as follows: a link from node A to node B is unidirectional, if node B can receive messages from A, but not vise versa.

Woo et. al. focus on link quality estimation in [2]. They measured link quality for a sensor network deployment consisting of 50 Mica Motes from Berkeley. All nodes within a distance of about 10 feet (about 3 meters) or less from the sender received more than 90 % of the transmitted packets (called the effective region). It is followed by the transitional region which reaches roughly from 10 feet to 40 feet (between 3 and 13 meters) distance. Nodes in this region cannot be uniformly characterized as some of them have a high reception rate while others received no packets at all. The last region is the clear region and contains only nodes that did not receive any transmissions.

Zhao and Govindan measured the properties of wireless sensor networks on the physical and medium access control layers [3]. These measurements were conducted using up to 60 Mica motes, which were placed in three different environments: an office building, a parking lot and a habitat. The experiments for the physical layer were realized with a single sender and multiple receiver nodes, and have shown the existence of a grey area in reception which can consist of up to one third of the network (similar to the transitional region described above). Another result described by the authors is that in the parking lot and indoor environments nearly 10 % of measured links were unidirectional (called asymmetric links in the paper).

The Medium Access Control (MAC) layer evaluation used a simple Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol, which is the default implementation for TinyOS. It was augmented with a retransmission scheme, to make use of the link-layer acknowledgments that were being transmitted anyway. The authors have defined the packet loss difference for two nodes as the difference between the packet delivery efficiency of both nodes. Unidirectional links are quite common: more than 10 % of the surveyed links have a difference of more than 50 %. Ortiz and Culler studied the feasibility of using multiple channels in wireless sensor networks [6]. They evaluated link quality in three different testbeds: a machine room, a computer room and an office building, using up to 60 sensor nodes. During the experiments, each node transmitted 100 messages and each other node recorded the number of received messages, enabling easy calculation of the packet reception rate. The authors found that unidirectional links were indeed common in their testbeds. In the machine room 32 - 36 % of links were unidirectional, 18 - 34 % in the computer room and 10 - 46 % in the office building. In previous work [4], we described connectivity measurements conducted using eZ430-Chronos sensor nodes from Texas Instruments. We evaluated different placements (desk, lawn, stones), different heights (ground or poles) and two radio channels. Connectivity graphs were gathered every minute, for 60 minutes in each experiment. The results show that unidirectional links were extremely common in those experiments, there were always more unidirectional than bidirectional links. Also, the increased communication range that resulted from the higher placement on the poles led to a stronger increase of unidirectional links than of bidirectional ones. On average, we measured about four to five times more unidirectional than bidirectional links. Furthermore, we found that all links were extremely unstable, with lots of link changes between measurements (minutes). All these experiments show that unidirectional links are normal in wireless sensor networks and should be taken into account when routing decisions are made. Using them can increase connectivity, which may prevent network separation and increase performance.

3. Related Work

AODVBR [7] is an enhancement of Ad-Hoc On Demand Distance Vector Routing (AODV) [8, 9], that uses a mesh structure to supply multiple paths. The main achievement of the protocol is to build multiple routes without sending additional control messages. This is possible because of the broadcast character of the medium. Every node that overhears a route reply packet and is not the addressed next hop discards this packet in AODV. In AODVBR, these nodes enter the node from which the route reply was received as next hop to the destination into their routing cache. This way, a structure similar to a fish bone is constructed.

When a link breaks, the node that detected the break (re-)broadcasts the data packet with a flag indicating that this message should be sent using an alternative route. A neighboring node that receives this message and has overheard the route reply that created this route forwards the message to the next hop. This way, a detour of one hop is taken, which may enable the delivery of the data packet. Also, a route error packet is transmitted to the source, so that a new and possibly better route can be established. However, the message still has to traverse all nodes that are on the original route.

Dynamic Source Routing [10-12] is one of the first routing protocols that took unidirectional links into account. The authors specify two different modes of operation for DSR: one for the usage of only bidirectional links, and another which should be used when unidirectional links are common (used here). In this version, route request messages (RREQ) are flooded in the usual way. Route reply messages (RREP) however, are not sent back the inverted path of the RREQ message. Instead, the destination (D) inserts the path the RREQ has taken into a RREP message, which is also flooded. Once this message has arrived at the originator of the RREQ message (the source, S), S inserts the path taken by the RREQ into its routing table and transmits an additional routing message to node D, which contains the path taken by the RREP. Once the destination has received this message, the routes from S to D and from D to S, which can differ strongly, have been established.

Virtual coordinates are used by ABVCap Uni [13] to enable the usage of geographic routing in networks without location information. ABVCap Uni uses clusters and rings to enable the usage of unidirectional links. The overhead of maintaining clusters and rings is high, though. When links change often, the performance of ABVCap Uni decreases drastically.

In previous work, we introduced Buckshot Routing [14], a source routing protocol for dense ad-hoc networks. It uses a multi path approach to circumvent broken links, unidirectional links or dead nodes. These multiple paths are implemented by a limited directional flooding: when a node receives a message, the forwarding decision differs from that used in traditional source routing protocols. Normally, a node that receives a message only checks if it is the intended next hop. In Buckshot Routing, only the one after that is important, the next-but-one hop. All nodes that have this next-but-one hop in their neighbor table forward the message.

4. BuckshotDV

Buckshot Routing and BuckshotDV are both based on a limited directional flooding. When a node S wants to transmit a message to a node D and a path is already known, messages are not only sent along this path, but also within a certain tunnel around the original route.

An example of the forwarding mechanism is depicted in Fig. 1. The original path from node S to node D is a straight line in the middle of the figure. Where in traditional routing protocols a node only forwards the message if it is the intended next hop, nodes forward it if they have the hop after the next in their neighbor table in Buckshot Routing and BuckshotDV. This results in a higher message load, but also adds redundancy to the forwarding mechanism.

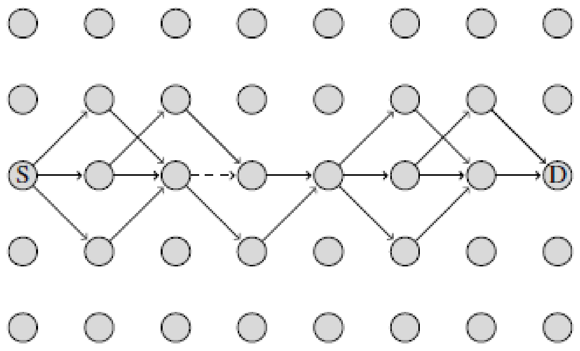

Fig. 1. Multiple paths taken by a message in Buckshot Routing and BuckshotDV.

The usefulness of the created redundancy can also be seen in Fig. 1. The dashed link between the second and the third node on the path is now broken, which would usually result in a delivery failure. In Buckshot and BuckshotDV, this broken link is implicitly circumvented, removing the need for a new route discovery.

Buckshot Routing and BuckshotDV are based on the same forwarding mechanism. However, while Buckshot Routing works quite well in networks with a small diameter, wireless sensor networks are assumed to consist of thousands of nodes in the future. The source routing character of Buckshot Routing means that the size of messages grows with the route length, which can become a problem in state of the art wireless sensor networks where the upper bounds for message size can be quite low (e.g., 64 Byte on the eZ430-Chronos from Texas Instruments [15]).

To make the forwarding principle of our Buckshot Routing usable in large scale networks, we developed its distance vector version called BuckshotDV, which reduces the message size while at the same time increasing the robustness of the routing protocol and increasing the delivery ratio.

In traditional distance vector routing algorithms like DSDV [16] or AODV, each node maintains a routing table, with entries consisting at least of the ID of the destination, the distance, and the next hop. Using the same entries in Buckshot Routing with Distance Vectors (BuckshotDV) is simply not possible. Buckshot Routing needs to know the next-but-one hop, which means that this value has to be kept in the routing table, too. Instead of a whole path with many node identities, a routing table for BuckshotDV contains a single ID: The next but one hop. But this has to be determined somehow, requiring changes to the way route request (RREQ) messages are built in distance vector protocols.

4.1. Message Types

In BuckshotDV a node enters its own ID along with the ID of the node from which it received a RREQ message before retransmitting it, previous entries are overwritten. A node that receives a RREQ message now knows its neighbor's neighbor, and thus the next-but-one hop on the reversed path, which it enters into its routing table in the form (Source of RREQ, next-but-one hop, distance). This entry is based on the fact that in Buckshot Routing the ”real” next hop is never important, only the next-but-one hop. The only exception to this is the source/destination, which does not have a next-but-one hop. To compensate this the source enters an illegal ID when creating a RREQ message.

The first value in the RREQ is the type of message, followed by the sequence number of the originating node and its identity, which are used for duplicate suppression and to build the reversed route. The destination ID is of course necessary to terminate the route discovery once the destination has been reached. All of these values are fixed throughout the lifetime of a RREQ message.

The first value being subject to change is the hop count which is incremented by one on each hop. Please note that of course any other weight function like, e.g. energy, would also be possible. The hop count is followed by the identities of the previous and the current hop, which of course change with each hop the message takes.

Fig. 2 shows an example of a RREQ message that is transmitted from its source to node A and then to node B. The changing values are initialized with 0 for the hop count, an illegal value and the source's ID before the source transmits it RREQ message. Upon reception of this message, node A enters the source with a distance of 1 and next-but-one hop: the illegal value into its routing table. This is necessary to prevent all other neighbors from rebroadcasting a message from A to the destination over and over again. After creating the routing table entry, node A increments the hop count of the RREQ message and enters the last hop (the source) and its own ID before retransmission. On node B the procedure is the same. If some node C received the message from B it would create a routing entry consisting of the source, a distance of 3 and node A as the next-but-one hop.

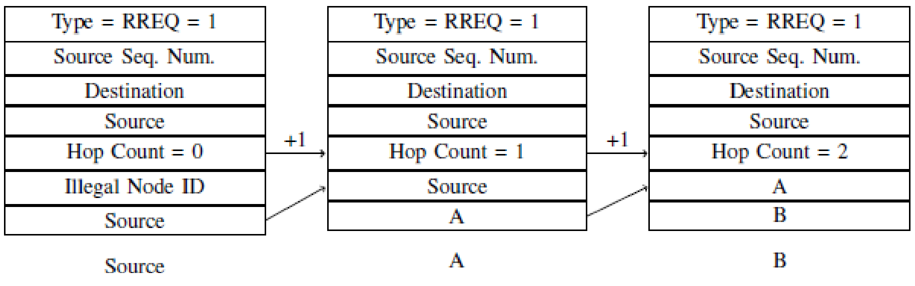

Fig. 2. Per Hop Changes in a Route Request Message in BuckshotDV.

When a node receives a RREQ and determines that it is the destination of this packet, it creates a routing entry for the source of the RREQ message and transmits a route reply (RREP). RREP messages contain the ID of the node from which the RREP was received and the identity of the next-butone hop in BuckshotDV. The next-but-one hop is needed to find the route to the source of the RREQ message, the identity of the previous node is needed to build the backward route. Thus, contrary to Buckshot Routing in its basic source routing variant, RREP messages are also used to build new routes. Nodes that receive a RREP message check their neighbor table for the next-but-one hop listed there, which is the next hop from their perspective. If and only if there is an entry, they look up the next-but-one hop from their perspective in their routing table, adjust the values in the RREP message and retransmit it.

In Fig. 3 an example of the way RREP messages are handled in BuckshotDV is given. The RREP message consists of four values that do not change and four that do. The type, sequence number, source ID and destination ID are used in exactly the same way as before. In the varying fields, the hop count has been reset by the destination of the RREQ before retransmission and now denotes the distance of each node that receives the RREP message from the destination of the RREQ. Following the hop count, the identity of the next-but-one hop is inserted, which is used on the receiving nodes to decide whether they should forward the message, following Buckshot Routing's forwarding mechanism. The other two varying fields are the same as in the RREQ message: The identity of the last and current hop. They are used to build routing table entries for the way to the destination (of the RREQ message). In the example, node A enters an illegal value into the next-but-one hop field, because it is a direct neighbor of the destination and no next-but-one hop exists. Please note that node A does not know that, i.e. it retrieves this value from its routing table. No special case handling is required, because the illegal value had been present as next-but-one hop in the RREQ due to which this routing entry was made.

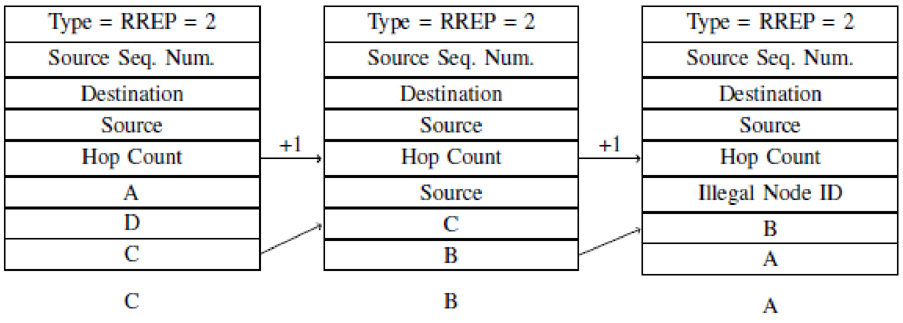

Fig. 3. Per Hop Changes in a Route Reply Message in BuckshotDV.

An example of the way routing table entries are created in BuckshotDV can be seen in Fig. 4. Node S, the source, searches for a route to node D, the destination. It transmits a route request message as described above. Upon reception of this message, node A enters node S into its routing table with no next-but-one hop (illegal value: NULL) and a distance of 1. When node B receives the (modified) RREQ, it enters node S with next-but-one hop node S and a distance of 2 into its routing table. Finally, node D receives the RREQ, creating an entry consisting of node A as next-but-one hop and a distance of 3 for node S. This concludes the building of the backward route. Now the forward route has to be established by the route reply message, which is transmitted by node D. Node B receives it and enters node D with no next-but-one hop and a distance of 1 into its neighbor table. For node A the entry consists of node D as next-but-one hop and a distance of 2. Node S enters node B as next-but-one hop and a distance of 3 into its routing table.

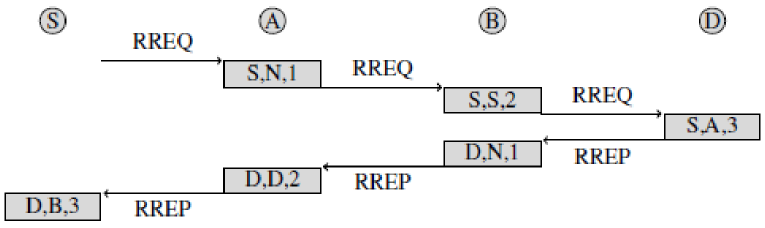

Fig. 4. Routing Table Entries Generated by BuckshotDV.

Once the RREP message has arrived at the source and the routing table entry has been created, the DATA packet can be transmitted. As no new route needs to be learned from a data packet, the identities of the previous and current hop are omitted in DATA packets.

The data packet format used in BuckshotDV is shown in Fig. 5. Just like when forwarding a RREP message, each node that receives a DATA message checks its neighbor table for the next-but-one hop listed in the message and replaces it with its own next-but-one hop for the listed destination if and only if it has found the neighbor in its neighbor table.

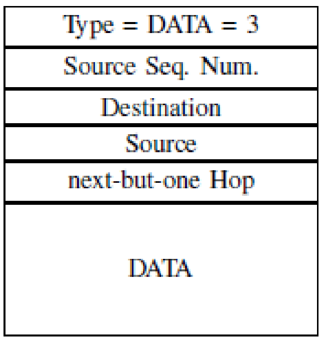

Fig. 5. DATA Message Format in BuckshotDV.

4.2. Variations

A possible variation of BuckshotDV concerns the DATA messages. In the basic version, they contain only one entry that changes with each hop: The next-but-one hop which is used for routing decisions. It would be possible to include the current and previous hop, to learn about the path that has been taken by the DATA message. Then, a node that receives a message could update its routing table entry for the source of the data message.

4.3. Advantages and Disadvantages

When compared to pure Buckshot Routing, BuckshotDV is complicated and requires more computation and copying on each node. Still, when comparing it to protocols like AODV, it remains simple. Its main advantage compared to Buckshot Routing is its scalability. In Buckshot Routing, as in all source routing protocols, the message headers grow with increasing network diameter. In BuckshotDV the header size is constant for each type of packet, making it usable in large networks. However, where Buckshot Routing is able to use route shortening if some of the intermediate nodes move closer to the source, BuckshotDV is not. The only point where BuckshotDV could use route shortening is when the destination becomes a direct neighbor of one of the intermediate nodes, which could then find it in its neighbor table.

5. Evaluation

The evaluation includes simulations using the OMNeT++ framework [17], as well as outdoor experiments on 36 eZ430- Chronos sensor nodes from Texas Instruments [15].

OMNeT++ [17] is a discrete event simulator that can be used to simulate different kinds of networks. OMNeT++ supplies a framework of modules which can be combined to form compound modules. Both types of modules contain gates, which can be connected using channels, to allow the modules to communicate with each other. This is realized by passing messages from one module to the other. OMNeT++ is implemented in C++ and enables the usage of the same code in simulations as well as on the sensor nodes. To enable simulations of a sensor network, the MiXiM framework [18] has been used. It provides an abstraction for communication layers. Explicit simulation of unidirectional links using a connectivity matrix has been added to MiXiM for this evaluation.

For all experiments, eZ430-Chronos Sensor nodes from Texas Instruments [15] were used. The eZ430-Chronos is an inexpensive evaluation platform for the CC430. These feature an MSP430 micro controller with an integrated CC1101 sub-gigahertz (868 MHz) communication module [19]. The evaluation board is delivered as a compact sports watch containing several sensors, e.g. a three-axis accelerometer, and five buttons which are connected through general purpose I/O pins.

Fig. 6 shows the used eZ430-Chronos sensor nodes in three different placements which were used in the experiments. An external battery pack has been soldered to the nodes, which replaces the internal coin cells. This enables the usage of freshly charged batteries for each protocol.

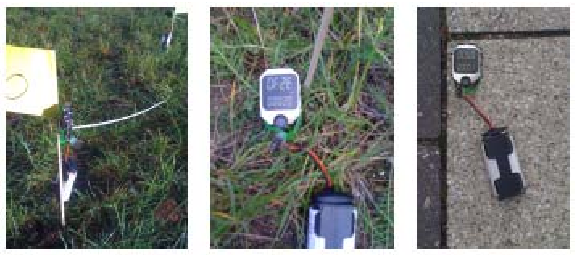

Fig. 6. A modified eZ430-Chronos sensor node from Texas Instruments, (a) affixed to poles; (b) placed on the lawn; (c) on a stone pavement.

Apart from the modification for the batteries, the sensor nodes were used as they were delivered, no calibration was made. The transmission power was also left at the preset level of 0 dBm, which lead to a small transmission range. This small transmission range is also due to the absence of a real antenna on the eZ430-Chronos: the metal surrounding the display acts as antenna.

Five different routing protocols were chosen as competitors for BuckshotDV in the evaluation: Flooding, Tree Routing, AODVBR, DSR and Buckshot Routing.

5.1. Simulation Method

The simulated networks consisted of four different sizes of grids: 100 nodes (10x10), 400 nodes (20x20), 900 nodes (30x30) and 1600 nodes (40x40). A grid alignment was chosen to represent applications that need area coverage, where each node is equipped with sensors that have a range of one distance unit. To simulate a certain connectivity between nodes, we used the matrix-based simulation approach presented in [20]. As the largest networks, consisting of 1600 nodes, needed to be simulated for the longest time, they also needed the highest number of connectivity matrices: for a single simulation 17761 connectivity matrices were needed. In each of these matrices, a (directed) link from node A to node B exists with a probability of α / d^6 where d is the distance between node A and node B. The inverse link, from node B to node A, exists with the same probability. Therefore, the link is bidirectional with a probability of $\alpha / d^6 \times \alpha / d^6$, unidirectional (in any one direction) with $\alpha / d^6 \times (1 - (\alpha / d^6))$ and non existing with $(1 - (\alpha / d^6))^2$. The quotient (d^6) reflects the dampening induced by the distance between nodes while α represents the probability that a link between geographically adjacent nodes exists. Nodes that are directly above, below, right or left of a node are called direct neighbors and their distance was defined as 1. α was varied between 0.9, 0.95 and 1, and for each value of α ten sets of matrices with different seeds for the random number generator were generated, leading to 30 sets of matrices per network size, and a total of 996120 connectivity matrices containing between 10.000 and 2.560.000 entries. Please note that due to the fact that connectivity matrices were generated randomly, there is no guarantee that there always was a path from sender to destination. Therefore, no upper limit can be calculated, but Flooding is used as reference protocol: the number of application messages delivered by Flooding is taken as 100 % and the delivery ratio of all other protocols calculated accordingly.

5.2. Experimental Methodology

In the experiments, four different placements were used: a desk, a lawn, poles, and stones. The desk placement is a one hop environment with all 36 nodes lying directly next to each other. In the other experiments, nodes were placed one meter from each other, on the grass of a lawn, on a stone pavement or affixed to poles at a height of 20 cm above ground. Each placement has different radio characteristics. In the experiments the delivery ratio was defined as the number of received application messages divided by the number of application messages handed to the routing protocol.

5.3. Application Scenario 1: Sense and Send

The application implemented for scenario 1 represents a sense-and-send behavior that is often found in sensor networks: All nodes within the network wanted to transmit all their messages to the same destination.

The delivery ratio of Buckshot Routing, BuckshotDV, DSR, AODVBR, Flooding and Tree Routing is shown in Fig. 7. For a small network containing only 100 nodes, the delivery ratio of Buckshot Routing in its source routing version and that of BuckshotDV are still close to each other. However, when the number of nodes and thus the network diameter and route length increase, the performance of Buckshot Routing declines while that of BuckshotDV improves.

Indeed, the performance of all protocols declines, except for BuckshotDV. This is due to the forwarding mechanism of BuckshotDV, which always uses the next-but-one hop from the perspective of the node which is currently handling a message. As this next-but-one hop might be different for different nodes, the limited directional flooding gets broader with increasing network diameter, increasing redundancy and robustness against unidirectional links and link breaks.

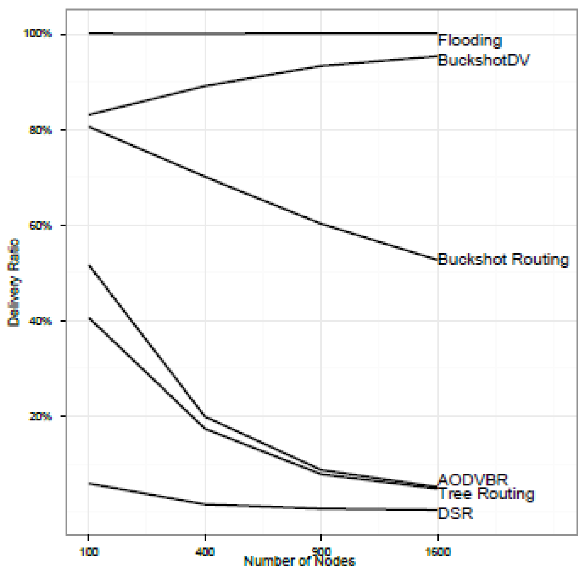

Fig. 7. Scenario 1: Delivery Ratio.

However, this increase in robustness comes at a price: the increased redundancy means that a higher number of messages are transmitted. Fig. 8 shows the number of messages transmitted by each protocol. While Flooding naturally transmitted the most messages, BuckshotDV nonetheless transmitted about twice as many messages as Buckshot Routing. The least number of transmitted messages can be seen on Tree Routing as expected: when a link breaks, two retransmissions are tried before the message is discarded, keeping the cost of delivery failure low. In the case of DSR, a failure to deliver a message to the next hop results in a route error message being transmitted to the originator of the message, and a subsequent new route discovery, which includes two floodings of the whole network. AODVBR should in theory be robust against message losses due to the fish bone structure it uses to reclaim lost data messages. However, this reclaiming mechanism is only used for data messages, meaning that AODVBR needs a completely bidirectional path during route discovery.

The cost of delivering a single application message to the destination measured in transmitted messages is shown in Fig. 9. With a delivery ratio of 40 % and a low number of overall transmissions, Tree Routing can be a good choice for small networks if network load is more important than delivery ratio. DSR represents the other end of the spectrum - the low number of delivered application messages compared to the fairly high number of transmitted messages results in a very bad cost ratio.

Therefore, DSR should not be used in such dynamic environments. The ratio of AODVBR is also worse than that of Flooding, therefore it should not be used for the resource constrained sensor networks. The ratios of Buckshot and BuckshotDV are close to each other, with BuckshotDV a little worse. The decision which of these two should be used in a certain scenario depends on the importance of data and network load: if the delivery ratio is more important, BuckshotDV should be chosen, and the increased network load tolerated. If network load needs to be reduced, Buckshot Routing should be used.

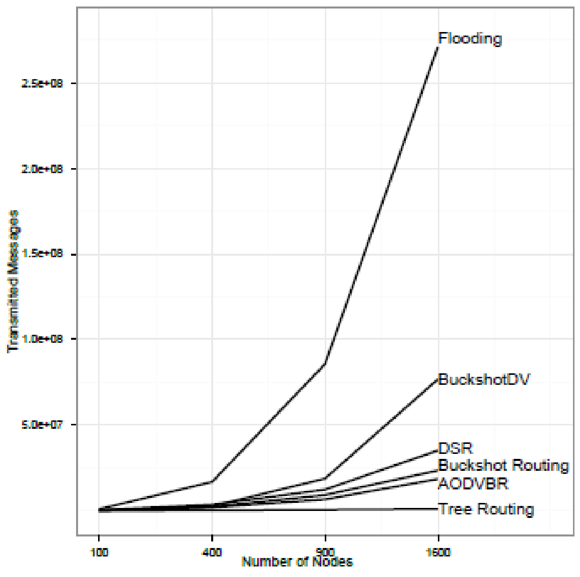

Fig. 8. Scenario 1: Transmitted messages.

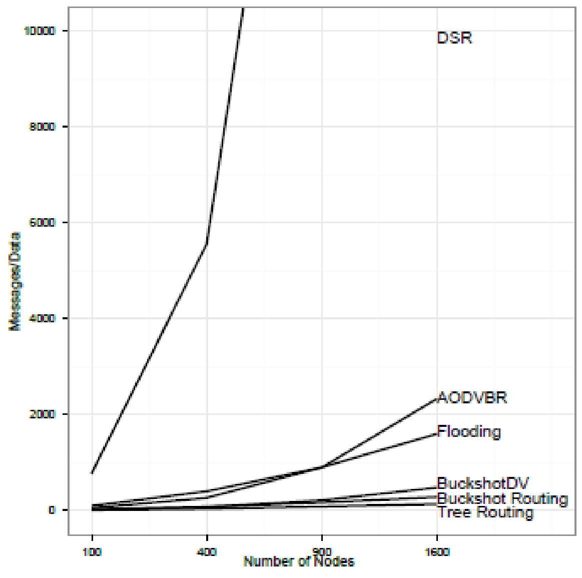

Fig. 9. Scenario 1: Transmissions per application message.

The delivery ratio of each protocol, divided by different placements, is shown in Fig. 10. The figure shows that all protocols work well on the desk, and fairly well in the pole experiments. But when the sensor nodes are placed on the ground, AODVBR and DSR show a steep decline in delivery ratio. Tree Routing works much better, but still not as well as Buckshot Routing or BuckshotDV. Even Flooding shows a strong decline, which is due to problems with the MAC layer. However, BuckshotDV outperforms all protocols chosen for comparison.

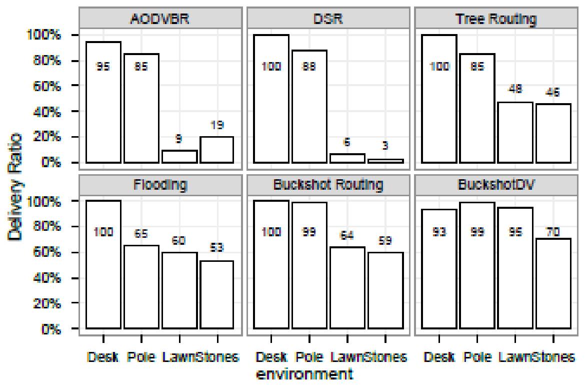

Fig. 10. Delivery ratio of each protocol achieved in the experiments.

The total number of messages transmitted by each protocol is shown in Fig. 11. It can be seen that Flooding transmits the most messages in all placements, with Buckshot Routing and DSR following for the placements on the ground (Lawn, Stones). Tree Routing transmitted the lowest number of messages in the placements on the ground. However, the number of transmitted messages needs to be correlated to the delivery ratio.

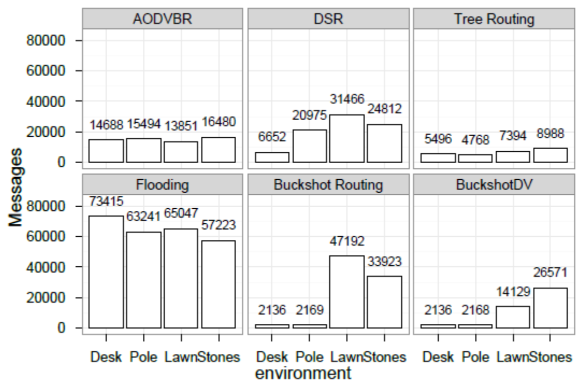

Fig. 11. Number of messages transmitted by each protocol in the experiments.

The cost of delivering a single application message to its destination measured in transmitted messages is shown in Fig. 12. DSR performs worst due to the high number of transmissions and low number of delivered application messages. However, BuckshotDV performs at least as well as all other protocols, often outperforming its competitors. Even

TreeRouting which has a better cost function in the stones placement has a lower delivery ratio for that same scenario. In the desk and pole placements Buckshot Routing has the same ratio of 1 as BuckshotDV and in the lawn placement Tree Routing and BuckshotDV share a value of 7. When the delivery ratio is taken into consideration, this means that BuckshotDV always outperforms its competitors, even for our relatively small testbed consisting only of 36 nodes.

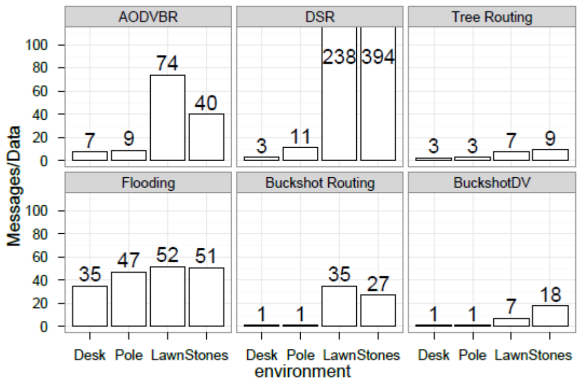

Fig. 12. Number of messages transmitted to deliver a single application message in the experiments.

5.4. Application Scenario 2: Single Pairing

In this scenario all settings, including the number of messages a node wants to transmit, are the same as in the sense-and-send scenario. However, instead of a single sink as destination for all messages from all nodes, each node has a randomly chosen partner node it wants to communicate with. This pairing of nodes was generated before the simulations and experiments, and differs only between different network sizes: If e.g. node 15 is the partner of node 21 for the network consisting of 36 nodes, this pairing remains fixed for all protocols as well as for simulations and real world experiments.

This pairing of nodes represents a communication pattern for MANETs and was chosen because two of the protocols used for comparison (AODVBR and DSR) are MANET protocols.

In the simulations for the single pairing scenario, the same connectivity change lists were used that have already been used in the sense-and-send scenario. However, as the destination was not a single fixed one for all nodes, the simulations were not varied according to the destination. Instead, the generated pairings were used as stated above.

Flooding was once again used to measure the upper limit for delivered messages and the delivery ratio was defined as the number of messages delivered by a protocol divided by the number of message delivered by Flooding.

The delivery ratio of AODVBR, Buckshot Routing, BuckshotDV, DSR, Flooding and Tree Routing is shown in Fig. 13. For all protocols except Flooding the delivery ratio declines with increasing number of nodes. It can be seen that AODVBR and Tree Routing suffer the most from the increased route length in the larger networks, as the decline of their delivery ratio is steep. For AODVBR, building the initial route is the crucial part. When a route has been successfully established, the fish bone structure can be used to salvage data packets. But since building the initial route requires a bidirectional path and the probability of a complete path being bidirectional decreases with route length, AODVBR only works in small networks. For Tree Routing, building the initial route is no problem. However, due to the dynamic nature of links between nodes, the initial path is obsolete soon and the two retransmissions used as reaction to message loss are not sufficient in larger networks.

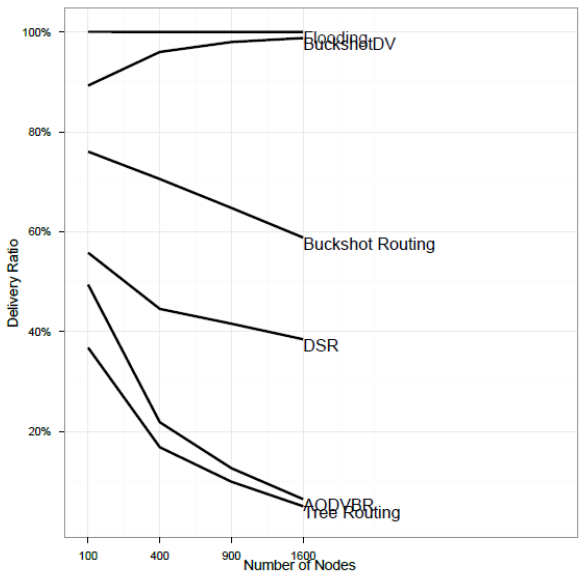

Fig. 13. Delivery ratio of AODV-BR, Buckshot Routing, DSR, Flooding and Tree Routing, Scenario 2.

The delivery ratio of DSR and Buckshot Routing also declines due to their source routing nature. However, finding the initial route is not a problem for either of them, as DSR uses one flooding for each direction and Buckshot Routing uses multiple paths implicitly. The main difference between both protocols is their route maintenance mechanisms. When DSR detects a route break it tries to inform the originator of the message that caused the detection of the break. Following this, a new route discovery with all its costs takes place. In Buckshot Routing this route maintenance is done implicitly with each received message, resulting in fewer stale routes and a better delivery ratio. Also, a maximum route length of 40 and caching of overheard routes were used for Buckshot Routing and DSR in this scenario. When the delivery ratio of Buckshot Routing for this scenario is compared to that achieved by the same variant in the sense-and-send scenario it can be seen that the delivery ratio has increased from less than

20 % to nearly 60% in the networks containing 1600 nodes. This is due to the fact that the implicit route maintenance did not work in the sense-and-send scenario as all messages were transmitted from the nodes to the sink. As the nodes never received replies from the sink, they could never use the implicit route maintenance mechanism. In this scenario however, the pairing of nodes results in a constant message exchange between a node and its partner, leading to routes that are up to date most of the time. The figure once more confirms that the performance of BuckshotDV increases with network size as the number of available redundant paths increases.

The number of transmitted messages for each protocol is shown in Fig. 14.

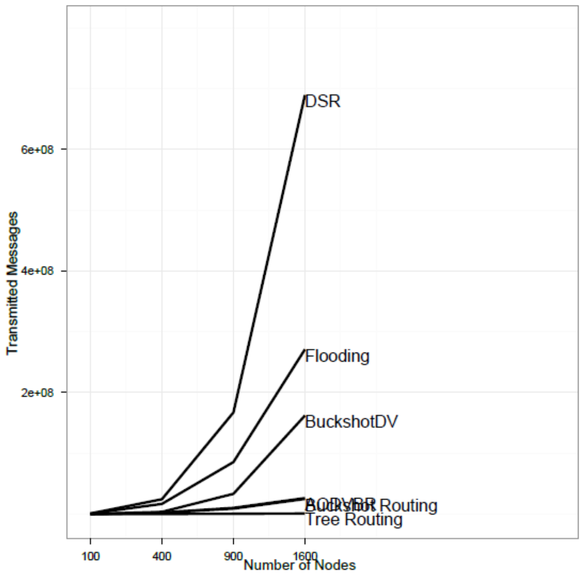

Fig. 14. Number of transmitted Messages, AODV-BR, Buckshot Routing, DSR, Flooding and Tree Routing, Scenario 2.

Here, the impact of the route maintenance mechanism of DSR can be seen: It transmits more than twice as many messages as Flooding as it tries to repair broken routes. Tree Routing presents the other extreme, it transmits nearly no messages at all, while Buckshot Routing and AODVBR need slightly more messages. When these results are compared to those of the sense-and-send scenario, it can be seen that the number of messages transmitted by BuckshotDV has increased much more than that of the other protocols. This is the price that BuckshotDV pays for a high delivery ratio: As the routing tables are continuously refreshed by messages from the partner node, the number of nodes that receive a message and also know the next but one hop increases. As all of these forward the messages in BuckshotDV, the number of redundant paths that are used is increased. This leads to an increase of delivery ratio of roughly 5 % for all network sizes. But to achieve this raise in delivery ratio the number of transmitted messages is nearly doubled.

When the network load is considered (Fig. 15), the impact of the low number of messages transmitted by Tree Routing can be seen even better: The number of messages transmitted to deliver a single application message would suggest that Tree Routing is an excellent choice. However, this fact needs to be correlated with the delivery ratio in most cases, and the delivery ratio of Tree Routing is the lowest of all protocols. This is once again due to the length of routes. Tree Routing delivers a nearly constant number of data messages to the destination (roundabout 8000) for the networks with 400, 900 and 1600 nodes, even though the total number of application messages that is handed to the routing protocol increases proportionally to the number of nodes in the network. The increased number of messages transmitted by BuckshotDV naturally also increases the number of messages transmitted to deliver a single application message. BuckshotDV still performs best, but only marginally. When it is compared to the performance of Buckshot Routing in its source routing variant, it can be seen that the source routing variant transmits much fewer messages per delivered data message but only has a delivery ratio of 59 % for the network consisting of 1600 nodes, whereas BuckshotDV delivers 99 %. This is a good example for a choice to be made by the application programmer: If high network load poses a problem but message delivery might fail every once in while, source routing Buckshot Routing can be used. But if the delivery ratio takes prominence over all else, BuckshotDV is the protocol of choice.

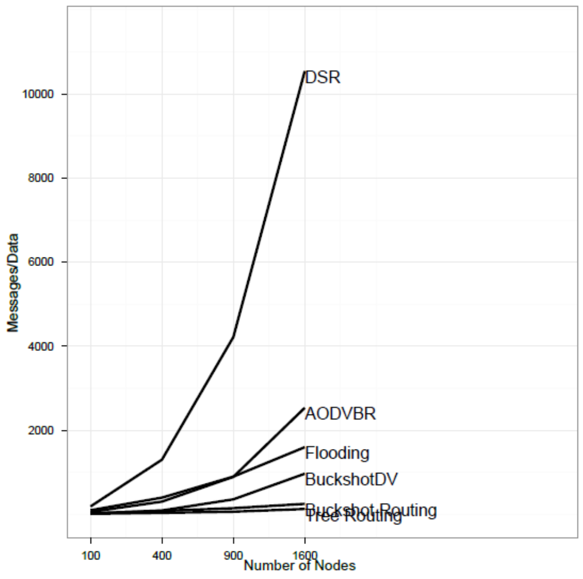

Fig. 15. Number of Messages transmitted to deliver a single application message, AODV-BR, Buckshot Routing, DSR, Flooding and Tree Routing, Scenario 2.

In the experiments for the single paring scenario, only two locations were used: The desk and the stone pavement. No experiments were made on the poles, because of the similarity between pole and desk scenario. On the desk, all nodes can communicate directly while on the poles the logical distance between nodes was only 1-2 hops even in the sense-and send scenario where the destination was on the corner of the deployed grid. The pairings used in this scenario reduce the average route length and would result in even more single hop routes for the pole scenario, making the experiments redundant. The lawn placement has been neglected due to its similarity with the stone pavement placement.

Fig. 16 shows the delivery ratios of all protocols that were achieved in the real world experiments on the desk and stone pavement. All protocols delivered 100 % of messages in the desk scenario.

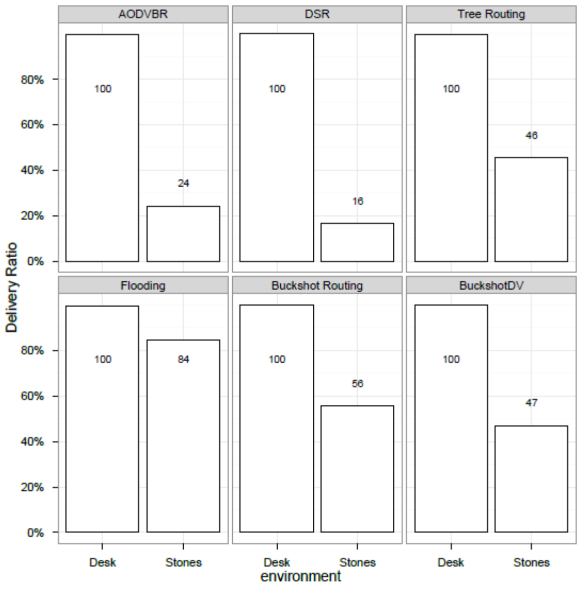

Fig. 16. Delivery Ratio of each Protocol achieved in the real experiments, Scenario 2.

In the stone pavement experiments, even Flooding did not deliver all messages, which gives an insight into the MAClayer problematic experienced more or less by all protocols. Flooding has the best delivery ratio in this scenario, Buckshot Routing and BuckshotDV which are next in line. Tree Routing also has a good delivery ratio in this scenario as it does not produce too much network load and the average path length was fairly small, making its two retransmissions a good reaction to message loss. DSR is continuously trying to repair routes, and thereby increases the network load very much, which can be seen in the next figure.

The total number of messages transmitted by each protocol is shown in Fig. 17 In the stone pavement placement, DSR transmits more than 57.000 messages and thus nearly as many as Flooding. Buckshot Routing and BuckshotDV transmit between 17.111 and 23.301 messages, while AODVBR and Tree Routing transmit about 15.000 and 8.000 messages respectively. These numbers already hint at the fact that Tree Routing profits quite a lot from the application setting and the small network diameter.

The number of messages transmitted to deliver a single application message is shown in Fig. 18. As there were 36 nodes in the network, Flooding transmitted 36 messages for each data message delivered to the destination. The overhearing of route reply messages described above leads to a good performance on the desk for Buckshot Routing and BuckshotDV, with AODVBR, DSR and Tree Routing following close.

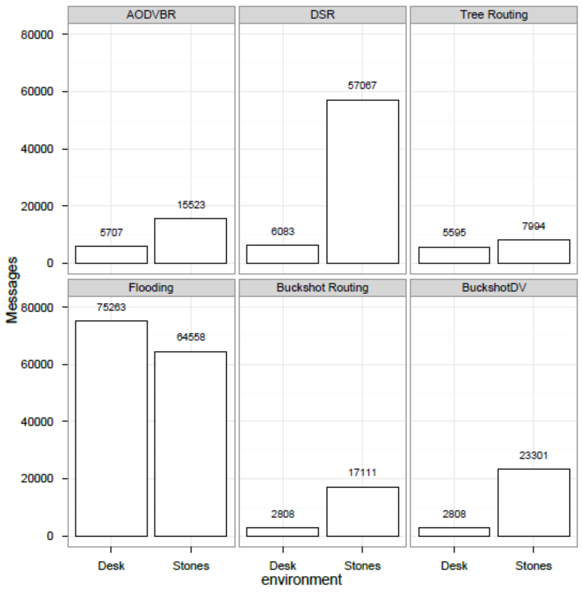

Fig. 17. Total Number of Messages transmitted by each Protocol, Scenario 2.

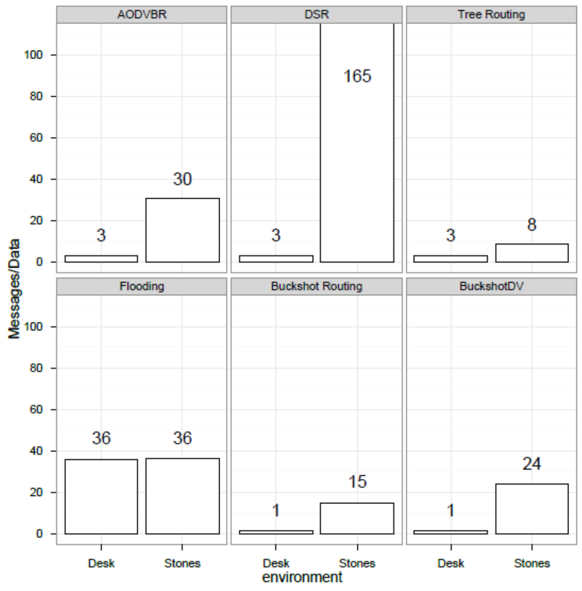

Fig. 18. Total Number of Messages transmitted by each Protocol divided by the number of delivered data messages, Scenario 2.

On the stone pavement, Tree Routing performed best, with Buckshot and Buckshot following. When the delivery ratio (Fig. 16) is also taken into account it can be said that for this application scenario, network size an placement, the choice of routing protocol should be made between Tree Routing and Buckshot Routing. Tree Routing produced the least network load per application message delivered and should be chosen if some message losses could be tolerated but the network load is the most important factor. Buckshot Routing should be chosen if network load is not a major concern.

5.5. Application Scenario 3: Multiple Pairings

The third application scenario, multiple pairings, once again uses the same settings as the two previous ones, only the application was changed. Instead of all nodes transmitting to a single sink or one communication partner for each node, there are multiple partners. Each node has one communication partner at the start of the simulations/experiments and transmits the first five messages to this node. Once five messages have been transmitted, the communication partner is changed. This is repeated every time five messages have been transmitted, until the total number of messages specified (110 for simulations, 60 for experiments) has been reached. The pairings of nodes were once again generated randomly before the start, and the same pairings were used for all protocols.

This represents a MANET scenario where all nodes only want to exchange a few messages with a chosen partner before communicating with a different node. The fact that each pairing is only used for five messages results in a reduction of the importance of route maintenance. It is much more likely that a route is stable for five minutes than for a whole simulation/experiment, resulting in less route errors. Instead, route discovery rises in importance, as it is carried out after every five application messages.

The simulations once again used the connectivity change lists that were generated before the start, to keep network connectivity equal for all protocols. As in the single pairing scenario, the pairings define a different destination for each node, making the additional simulation parameter destination used in the sense-and-send scenario unnecessary.

The delivery ratio remains defined as the number of application messages delivered by a protocol divided by the number of messages delivered by Flooding in the simulations.

The delivery ratio of Buckshot Routing and BuckshotDV is compared to that of the related work protocols in Fig. 19. It can be seen that Buckshot Routing still outperforms all related work protocols, even though the application scenario has been switched to one that should be better for the related work protocols. As the importance of route maintenance is reduced, one of the advantages of Buckshot Routing, the implicit route maintenance, has only a small impact.

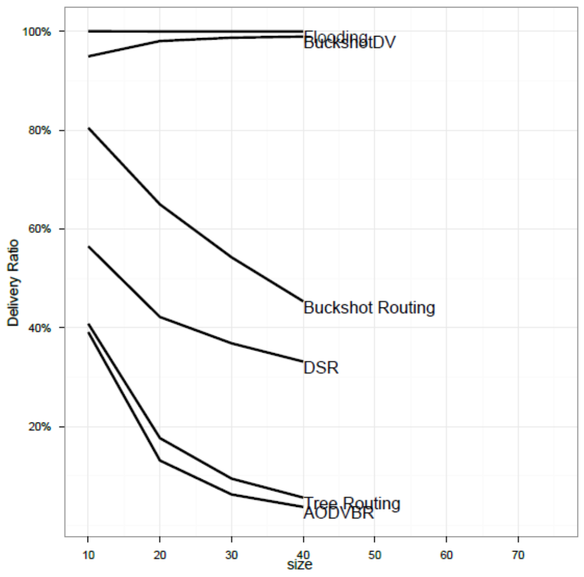

Fig. 19. Delivery ratio of AODV-BR, Buckshot Routing, DSR, Flooding and Tree Routing, Scenario 3.

For AODVBR and Tree Routing, the number of nodes and therefore the route length is much more important than the communication pattern of the application: The changes between single pairing and multiple pairings are marginal. The performance of Tree Routing increased by one percent for the largest network while that of AODVBR decreased by two percent. A bigger difference can be seen for the smaller networks, where AODVBR has lost 10 % of its performance compared to the single pairing scenario in the network consisting of 100 nodes. This decrease in delivery ratio is due to the fact that building the initial route is one of the weaknesses in AODVBR. When searching for a route, the path has to be bidirectional to enable the route reply to use the same path as the route request. Once this path has been established, the fish bone structure that has been built with the route replies can be used to salvage data messages when links break. In the multiple pairings scenario, each node needs to search routes to 22 different nodes instead of only one. BuckshotDV starts with a delivery ratio of 95 % and increases its performance up to 99 %.

The number of messages transmitted by Buckshot Routing, BuckshotDV and the related work protocols is shown in Fig. 20. With twice the number of transmitted messages as Flooding, DSR once more transmitted the most messages by far. Buckshot Routing, AODVBR and Tree Routing transmitted far less messages, with Tree Routing producing the least number. When the results are compared to those of the single pairing scenario, only Buckshot Routing shows a significant difference. This is due to the fact that Buckshot Routing now needs 22 times as many floodings of the network, one for each new route discovery and node in the network instead of only one for each node. As Buckshot Routing does not transmit any route maintenance messages, route discovery and data transmission are the two factors that define its performance. Therefore, the increased number of route discoveries has a strong influence on the number of transmitted messages. It can also be seen that the number of messages transmitted by BuckshotDV has risen when compared to the single pairing scenario. While the number is still lower than that of Flooding it has gotten close. The fact that the number of transmitted messages rises can be explained by the increase in redundancy and the higher number of route searches as the route replies already use multiple redundant paths.

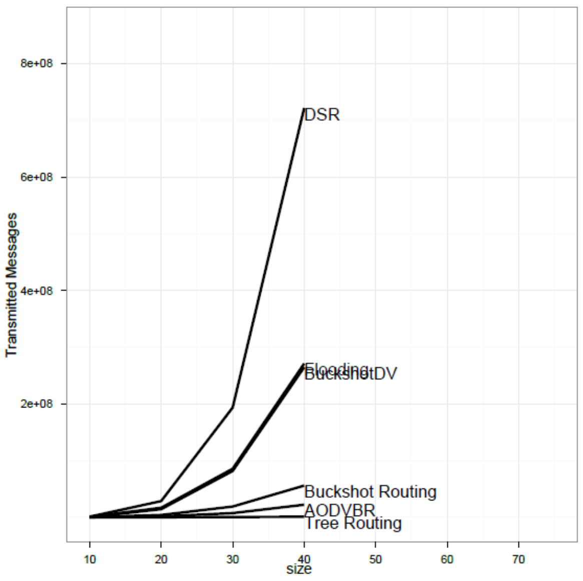

Fig. 20. Number of transmitted Messages, AODV-BR, Buckshot Routing, DSR, Flooding and Tree Routing, Scenario 3.

The cost of delivering a single data message, measured in transmitted messages, is shown in Fig. 21. Even though Buckshot Routing transmitted more messages than AODVBR, the much higher number of delivered messages results in a fairly good performance. Only Tree Routing transmitted fewer messages per application message delivered. However, this is once more due to the fact that the cost of delivery failure is small in Tree Routing. When the delivery ratio is also taken into account, Buckshot Routing emerges as the better protocol. On the downside, the increased number of messages transmitted by Buckshot Routing when compared to the single pairing scenario results in an increased cost of delivered messages is the reason why its performance decreases in the multiple pairings scenario.

The experiments for the multiple pairings scenario featured the same settings and locations as the experiments for the single pairing scenario (section V-D): The desk placement was used as single hop, and the stone pavement as multihop environment. The pole placement would have been redundant to the desk placement while the lawn placement would have been similar to the stone pavement environment.

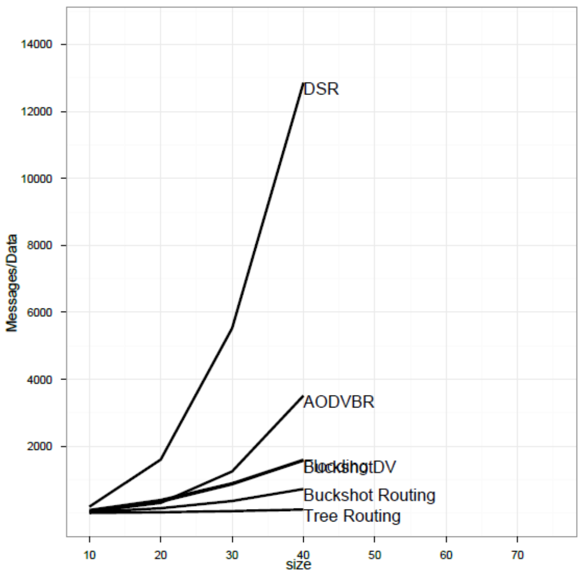

Fig. 21. Number of Messages transmitted to deliver a single application message, AODV-BR, Buckshot Routing, DSR, Flooding and Tree Routing, Scenario 3.

The delivery ratio achieved by all protocols in the multiple pairing scenario is shown in Fig. 22. In the desk experiments, all protocols reached 100 % delivery ratio.

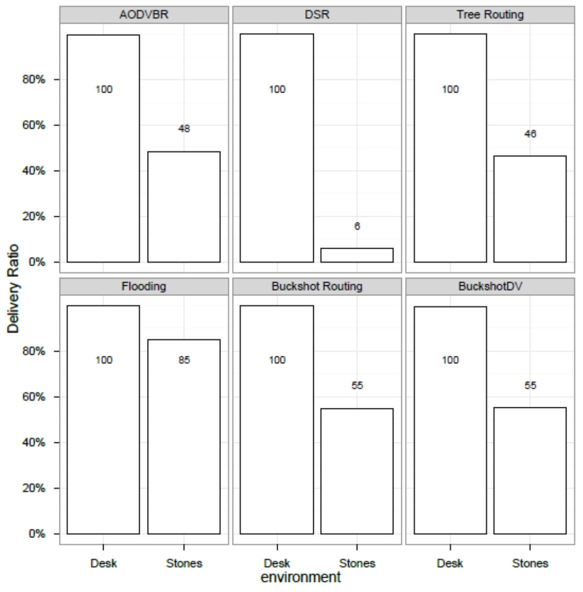

Fig. 22. Delivery Ratio of each Protocol achieved in the real experiments, Scenario 3.

On the stone pavement, Flooding has the highest delivery ratio, followed by Buckshot Routing and BuckshotDV. DSR performs worst. The reason for this can be found in the MAC layer, which has problems with a high network load produced by the many floodings of DSR.

This total number of transmitted messages is shown for all protocols in Fig. 23. Flooding once more has the highest number of transmitted messages for the single hop environment by far. On the stone pavement, Flooding also transmits the highest number of messages while Tree Routing transmits the smallest. Still, when considering that only 2160 application messages were generated it can be seen that Tree Routing often used its two retransmissions.

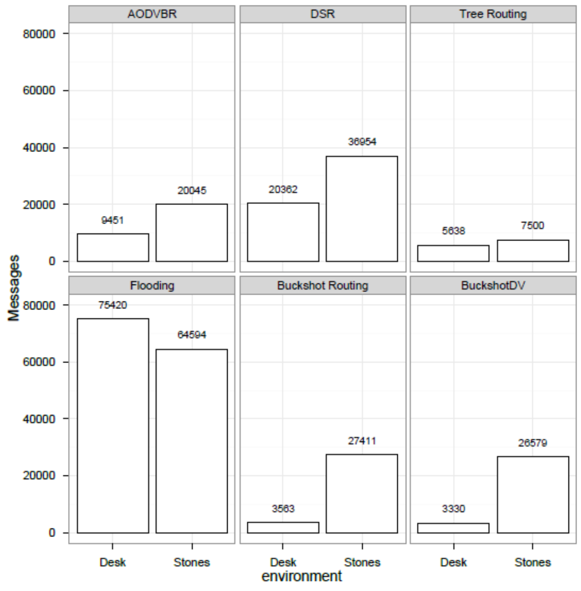

Fig. 23. Total Number of Messages transmitted by each Protocol, Scenario 3.

The number of transmitted messages divided by the number of delivered application messages is used to measure the performance of all protocols in Fig. 24. For the desk placement, Buckshot Routing and BuckshotDV show the best performance. Tree Routing is placed shortly thereafter, with AODVBR following while DSR and Flooding are far off.

When the sensor nodes were placed on the stone pavement, Tree Routing needed the least number of transmissions to deliver a single application message, which is once again due to the low cost of delivery failure. When only the cost of an application message delivery is considered, Tree Routing performs best. However, Buckshot Routing and BuckshotDV delivered much more application messages but also needed more messages to reach this increase in delivery ratio. If the delivery ratio is most important, BuckshotDV would be chosen for such small networks and this application scenario. If the network load is more important, Tree Routing should be chosen.

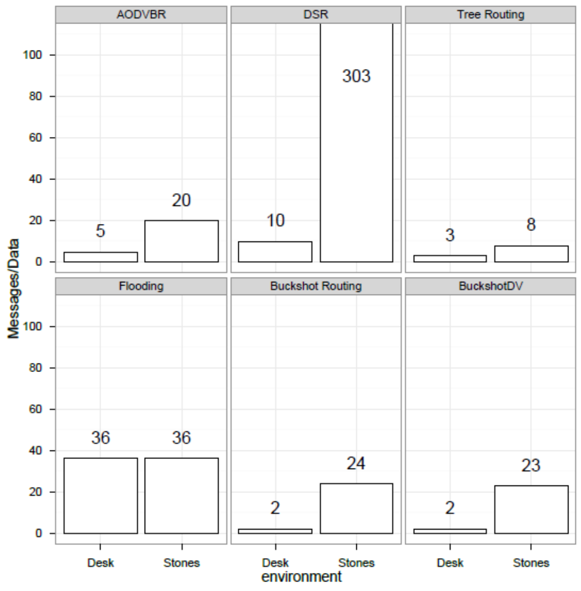

Fig. 24. Total Number of Messages transmitted by each Protocol divided by the number of delivered data messages, Scenario 3.

6. Conclusion

In this paper, we presented a closer look at BuckshotDV, a distance vector routing protocol for wireless sensor networks which uses unidirectional links implicitly. We evaluated its performance and compared the results to those achieved by AODVBR, DSR, Tree Routing, Flooding, and our original source routing version of Buckshot Routing. The evaluation included three different application scenarios: Sense-and-send with a single destination for all nodes, single pairing with one communication partner for each node and multiple pairings with changing communication partners at runtime. The experiments that we conducted with 36 sensor nodes from Texas Instruments show the feasibility of our approach, while the simulations of up to 1600 nodes were used to evaluate the scalability. Simulation results indicate that while Buckshot Routing can only be used for sensor networks with a moderate diameter, BuckshotDV can indeed be used in large scale networks. However, we did not posses enough hardware to prove this indication in large scale experiments.

The evaluation shows that BuckshotDV can operate in sensor networks with unidirectional links, and use them to increase its delivery ratio without introducing additional overhead. BuckshotDV does not need to inform upstream nodes of unidirectional links. Rather, those links are used implicitly. The implicit usage of multiple links makes BuckshotDV resilient against link changes and node failures, and removes the need for explicit route maintenance. Routing tables are implicitly updated with each received message, introducing no communication overhead and only negligible computation overhead on the nodes. The fact that the route maintenance overhead is marginal in BuckshotDV shows its usability for networks with frequent topology changes.

References

[1]. R. Karnapke and J. Nolte, Buckshotdv - a robust routing protocol for wireless sensor networks with unstable network topologies and unidirectional links, in *Proceedings of the 8th International Conference on Sensor Technologies and Applications*, 2014, pp. 60–65.

[2]. A. Woo, T. Tong, and D. Culler, Taming the underlying challenges of. reliable multihop routing in sensor networks, in *Proceedings. of the 1st International Conference on Embedded Networked Sensor. Systems (SenSys '03),* New York, NY, USA: 2003, pp. 14–27.

[3]. J. Zhao and R. Govindan, Understanding packet delivery performance in dense wireless sensor networks, in *Proceedings of the 1st international Conference on Embedded Networked Sensor Systems SenSys '03,* New York, NY, USA, 2003, pp. 1–13.

[4]. S. Lohs, R. Karnapke, and J. Nolte, Link stability in a wireless sensor. network - an experimental study, in *Proceedings of the 3rd International Conference on Sensor Systems and Software,* 2012, pp. 146 – 161.

[5]. M. K. Marina and S. R. Das, Routing performance in the presence of. unidirectional links in multihop wireless networks, in *Proceedings of the 3rd ACM International Symposium on Mobile ad hoc Networking &. Computing, (MobiHoc'02),* New York, NY, USA, 2002, pp. 12–23.

[6]. J. Ortiz and D. Culler, Multichannel reliability assessment in real world. WSNs, in *Proceedings of the 9th ACM/IEEE International. Conference on Information Processing in Sensor Networks (IPSN '10),* New York, NY, USA, 2010, pp. 162–173.

[7]. S.-J. Lee and M. Gerla, AODV-BR: Backup routing in ad hoc networks, in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'2000),* Chicago, IL, September 2000, pp. 1311–1316.

[8]. 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, 1999, pp. 90–100.

[9]. C. Perkins, E. Belding-Royer, and S. Das, Ad hoc on-demand distance vector (AODV) routing, http://www.ietf.org/rfc/rfc3561.txt, 2003.

[10]. D. Johnson, D. Maltz, and J. Broch, DSR The Dynamic Source Routing, Protocol for Multihop Wireless Ad Hoc Networks, *Addison-Wesley,* Ch. 5, 2001, pp. 139–172.

[11]. D. Johnson, H. Yu, and D. Maltz, The dynamic source routing protocol (DSR) for mobile ad hoc networks for ipv4, https://tools.ietf.org/html/rfc4728, Available: https://tools.ietf.org/html/rfc4728 last accessed September 2014.

[12]. D. B. Johnson and D. A. Maltz, Dynamic source routing in ad. hoc wireless networks, in Mobile Computing, *Kluwer Academic Publishers,* 1996, pp. 153–181.

[13]. C.-H. Lin, B.-H. Liu, H.-Y. Yang, C.-Y. Kao, and M.-J. Tsai, Virtualcoordinate-based delivery-guaranteed routing protocol in wireless sensor networks with unidirectional links, in *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM' 08),* Phoenix, AZ, USA., 13-18 April 2008, pp. 351–355.

[14]. D. Peters, R. Karnapke, and J. Nolte, Buckshot routing - a robust. source routing protocol for dense ad-hoc networks, in *Proceedings of the Ad Hoc Networks Conference*, 2009, Niagara Falls, Canada, 2009, pp. 268–283.

[15]. Texas instruments ez430-chronos, http://focus.ti. com/docs/toolsw/folders/print/ez430-chronos.html?DCMP=Chronos&HQS=Other+OT+chronos last accessed. September 2014.

[16]. C. E. Perkins and P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in *Proceedings of the Conference on Communications Architecture, Protocols and Applications (SIGCOMM'94),* New York, NY, USA, 1994, pp. 234–244.

[17]. A. Varga, The OMNET++ discrete event simulation system, in *Proceedings of the European Simulation Multiconference (ESM'01),* Prague, Czech Republic, Jun. 2001, p. 185.

[18]. A. Koepke, M. Swigulski, K. Wessel, D. Willkomm, P. Klein Haneveld, T. Parker, O. Visser, H. Lichte, and S. Valentin, Simulating wireless and mobile networks in OMNeT++: The MiXiM vision, in Proceedings of the 1st Int. Workshop on OMNeT++, mar 2008. Available: http://www.st.ewi.tudelft.nl/ koen/papers/mixim.pdf

[19]. Texas instruments cc430f6137, http://focus.ti.com /docs/prod/folders/print/cc430f6137.html, http://focus.ti.com/docs/prod/folders/print/cc430f6137.html. last accessed September 2014.

[20]. R. Karnapke, S. Lohs, A. Lagemann, and J. Nolte, Simulation of. unidirectional links in wireless sensor networks, in *Proceedings of the 7th International. ICST Conference on Simulation Tools and Techniques,* Lisbon, Portugal, 2014, pp. 118–125.