BuckshotDV - A Robust Routing Protocol 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
Email: {Karnapke, Jon}@informatik.tu-cottbus.de

***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. In this paper, we present 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.**

Keywords–wireless sensor networks; routing; unidirectional links

I. Introduction

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 [1], grey area [2]) 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 [3].

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 [4]. 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.

II. Unidirectional Links in Wireless Sensor Networks

Different classifications of link quality are used in literature. Examples are included in [1][2][3][5], 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 [1]. 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 [2]. 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

 ISBN: 978-1-61208-374-2

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 [5]. 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 [3], 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.

III. Related Work

AODVBR [6] is an enhancement of Ad-Hoc On Demand Distance Vector Routing (AODV) [7][8], 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 [9][10][11] 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 [12] 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 [13], 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.

IV. BuckshotDV

Buckshot Routing and BuckshotDV are both based on a limited directional flodding. 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 Figure 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.

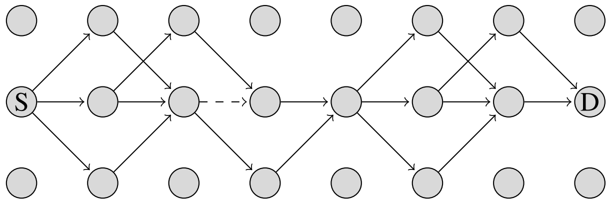


Figure 1. Multiple paths taken by a message in Buckshot Routing and BuckshotDV

The usefulness of the created redundancy can also be seen in Figure 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 [14]).

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 [15] or AODV [7], 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. As described in [13], Buckshot Routing needs to know the next-but-one hop, which means that this value has to be determined and kept in the routing table, too.

In BuckshotDV, a node enters its own ID along with the ID of the node from which it received a RREQ message into the RREQ before retransmitting it. 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).

Figure 2(a) shows an example of a RREQ message in BuckshotDV. 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, e.g., energy, would also be possible. The hop count is followed by the identities of the previous and the current hop.

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 also contain the ID of the node from which the RREP was received and the identity of the next-but-one hop in BuckshotDV (Figure 2(b)). 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 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.

Please note that the forwarding mechanism of BuckshotDV results in a spreading of messages across neighboring nodes (nodes adjacent to the initial path). This redundancy is what makes BuckshotDV resilient against link breaks and node failures. It is also the reason, why unidirectional links are implicitly circumvented.

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, resulting in a smaller header.

The data packet format used in BuckshotDV is shown in Figure 2(c). 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.

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.

V. EVALUATION

The evaluation includes simulations using the OMNeT++ framework [16], as well as outdoor experiments on 36 eZ430-Chronos sensor nodes from Texas Instruments [14]. These feature an MSP430 micro controller with an integrated CC1101 sub-gigahertz (868MHz) communication module [17].

Figure 3 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.

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.

Type = RREQ = 1
Source Seq. Num.
Destination
Source
Hop Count
Previous Hop ID
Current Node ID

(a) RREQ

Type = RREP = 2
Source Seq. Num.
Destination
Source
Hop Count
Next-but-one Hop
Previous Hop
Current Node ID

(b) RREP

Type = DATA = 3
Source Seq. Num.
Destination
Source
Next-but-one Hop
DATA

(c) DATA

Figure 2. Packet formats in BuckshotDV

(a) affixed to poles (b) placed on the lawn (c) on a stone pavement

Figure 3. A modified eZ430-Chronos sensor node from Texas Instruments

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

A. Simulation

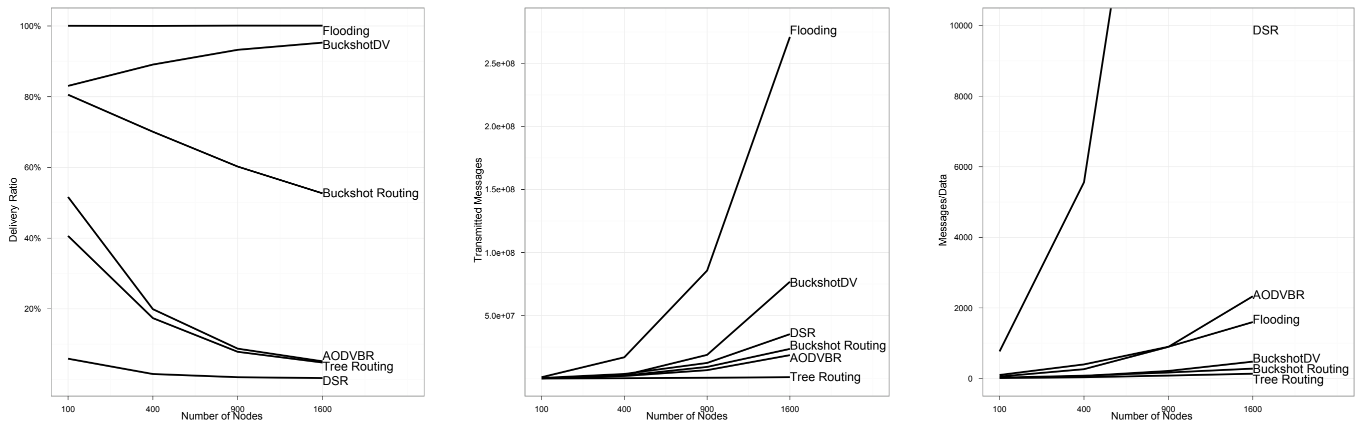
The simulated networks consisted of four different sizes of grids: 100 nodes (10x10), 400 nodes (20x20), 900 nodes (30x30) and 1600 nodes (40x40). A grid alignment was chosen to represent applications that need area coverage, where each node is equipped with sensors that have a range of one distance unit. To simulate a certain connectivity between nodes, we used the matrix-based simulation approach presented in [18]. 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.

The delivery ratio of Buckshot Routing, BuckshotDV, DSR, AODVBR and Tree Routing is shown in Figure 4(a). 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.

However, this increase in robustness comes at a price: the increased redundancy means that a higher number of messages is transmitted. Figure 4(b) 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.

 ISBN: 978-1-61208-374-2



(a) Delivery Ratio (b) Transmitted messages (c) Transmissons per application message

Figure 4. Performance of all evaluated protocols achieved in the simulations

The cost of delivering a single application message to the destination measured in transmitted messages is shown in Figure 4(c). 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.

B. Experiments

In the experiments, four different placement 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.

The delivery ratio of each protocol, divided by different placements, is shown in Figure 5. 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 good 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.

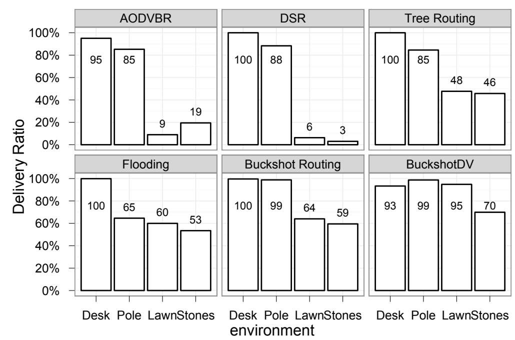


Figure 5. Delivery ratio of each protocol achieved in the experiments

The total number of messages transmitted by each protocol is shown in Figure 6. 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.

The cost of delivering a single application message to its destination measured in transmitted messages is shown in Figure 7. 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.

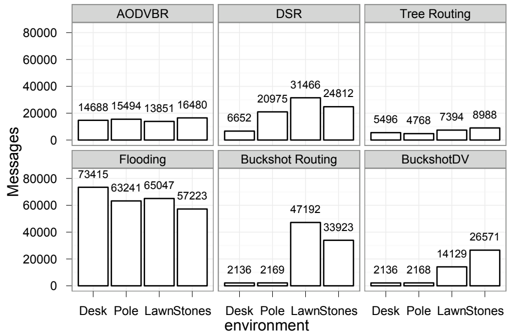


Figure 6. Number of messages transmitted by each protocol in the experiments

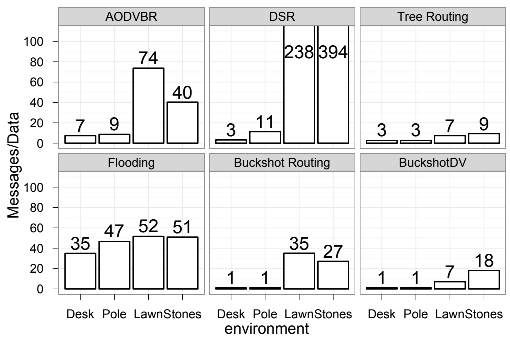


Figure 7. Number of messages transmitted to deliver a single application message in the experiments

VI. Conclusion

In this paper, we presented 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 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] 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. New York, NY, USA: ACM Press, 2003, pp. 14–27.

[2] J. Zhao and R. Govindan, “Understanding packet delivery performance in dense wireless sensor networks,” in SenSys ’03: Proceedings of the 1st international conference on Embedded networked sensor systems. New York, NY, USA: ACM Press, 2003, pp. 1–13.

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

[4] 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, ser. MobiHoc ’02. New York, NY, USA: ACM, 2002, pp. 12–23.

[5] J. Ortiz and D. Culler, “Multichannel reliability assessment in real world wsns,” in IPSN ’10: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks. New York, NY, USA: ACM, 2010, pp. 162–173.

[6] 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, Sepember 2000, pp. 1311–1316.

[7] 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, FEB 1999, pp. 90–100.

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

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

[10] 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.” [Online]. Available: https://tools.ietf.org/html/rfc4728 last accessed September 2014

[11] 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.

[12] C.-H. Lin, B.-H. Liu, H.-Y. Yang, C.-Y. Kao, and M.-J. Tsai, “Virtual-coordinate-based delivery-guaranteed routing protocol in wireless sensor networks with unidirectional links,” in INFOCOM 2008. 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 13-18 April 2008, Phoenix, AZ, USA. IEEE, 2008, pp. 351–355.

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

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

[15] 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. New York, NY, USA: ACM Press, 1994, pp. 234–244.

[16] A. Varga, “The omnet++ discrete event simulation system,” in Proceedings of the European Simulation Multiconference (ESM’2001), Prague, Czech Republic, Jun. 2001, p. 185.

[17] “Texas instruments cc430f6137, http://focus.ti.com/docs/prod/folders/print/cc430f6137.html.” [Online]. Available: http://focus.ti.com/docs/prod/folders/print/cc430f6137.html last accessed September 2014

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