Unidirectional Link Triangle Routing for Wireless Sensor Networks

Reinhardt Karnapke and Jörg Nolte
Distributed Systems/Operating Systems group
Brandenburg University of Technology Cottbus
Cottbus, Germany
Email: {Karnapke, Jon}@informatik.tu-cottbus.de

***Abstract*—Experiments with wireless sensor networks have shown that asymmetric and unidirectional links do not only exist, but are indeed quite common. Still, many people argue that the gain in connectivity is not worth the effort of making them usable for routing protocols. In this paper, we follow the opposite approach and introduce Unidirectional Link Triangle Routing, which reduces the overhead and, therefore, makes unidirectional links usable on the routing layer.**

Keywords: Wireless Sensor Networks; Routing Protocols; Unidirectional Links

I. INTRODUCTION

Unidirectional links represent a serious problem for many routing protocols. Even though experiments with wireless senor networks have shown that they are quite common (e.g. [1], [2], [3], [4]), most routing protocols still ignore their existence or try to remove their implications by using blacklisting or similar methods.

Most of the available protocols that use unidirectional links need to inform the upstream node of its outgoing unidirectional link explicitly, which is often done proactively and introduces a lot of overhead [5]. One possible conclusion that is often drawn from this fact is that using unidirectional links in a routing protocol does not pay off. An alternative is to reduce the overhead produced by the protocols. We argue that the high number of unidirectional links found in experiments makes using them absolutely mandatory and introduce Unidirectional Link Triangle Routing, a routing protocol for wireless sensor networks with often occurring unidirectional links.

The mechanisms used by Unidirectional Link Triangle Routing to make unidirectional links usable without introducing too much overhead are described in Section II, followed by an evaluation both with simulations and real world experiments in Section III. Related work is shown in Section IV. We finish with a conclusion in Section V.

II. UNIDIRECTIONAL LINK TRIANGLE ROUTING

Unidirectional Link Triangle Routing (ULTR) has been designed to use unidirectional links instead of ignoring them or removing their implications. To make them usable, it can cooperate with a neighborhood discovery protocol if one is necessary for the application, or with the MAC layer if a TDMA protocol is used.

To make an existing neighborhood table usable for ULTR, a neighborhood table entry on node A should consist of at least three parts:

1) the ID of the neighbor (e.g. B),
2) the status of the link to that neighbor
3) the ID of a potential forwarding node

The status of a link is either bidirectional, unidirectional-incoming or unidirectional-outgoing. The forwarding node is only necessary if the link is unidirectional-incoming.

When a node wants to transmit a message to another node that is not included in its neighbor table or its routing table, it starts a route discovery by transmitting a route request (RREQ) message. This message is flooded through the network and creates routing entries for the source on all nodes it passes. The entries include only the next hop and the distance, resulting in a distance-vector protocol like AODV [6].

The differences start once the destination has been reached and transmitted the route reply. When a node receives a message that is not flooded, i.e., a route reply (RREP) or DATA message, it checks its routing table to find out which of its neighbors is the intended next hop, just like in AODV. Unlike AODV, there is another step after that one. Once the node knows the neighbor that has been chosen to forward the message, it checks its neighbor table to see if the link to that node is *currently* an unidirectional-incoming one. If it is, and a detour of one hop is possible, the node forwards the packet first to the detour node which, in turn, retransmits the message to the intended node. Otherwise, the message is silently discarded. Please note that broken links may be treated just like unidirectional-incoming ones.

Figure 1 shows a small part of a network and the corresponding neighborhood table entries used in this protocol: The nodes A, B and C are connected bidirectionally, with the exception of the link between nodes A and B, which is unidirectional and enables only transmissions from B to A. The neighborhood table of node A consists of two entries: one bidirectional entry for node C and a unidirectional-incoming one from node B, with node C denoted as designated forwarder. The neighborhood table of node B contains node A, which would not be possible without a two-hop neighborhood discovery protocol, as node B does not receive any messages from node A. The link is marked as unidirectional-outgoing, and, thus, does not need any forwarder. The second entry features node C with a bidirectional link, needing no forwarder either.

 ISBN: 978-1-61208-296-7

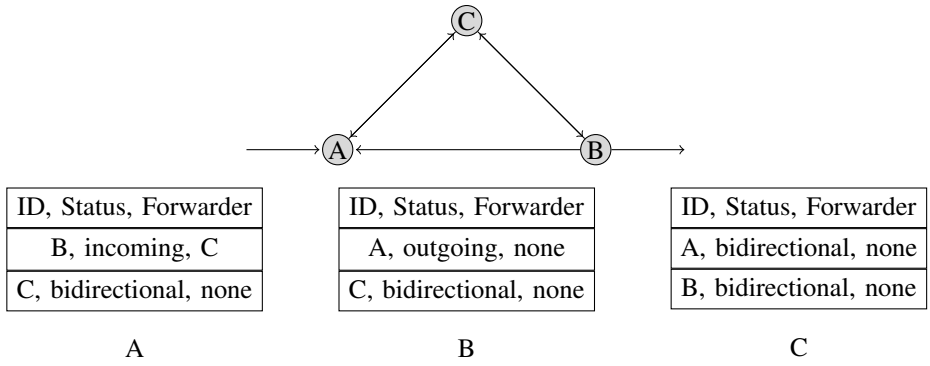


Fig. 1: Neighbor Table Entries in Unidirectional Link Triangle Routing

Finally, the neighborhood table of node C contains nodes A and B, both marked as connected through bidirectional links and not needing any forwarders.

The unidirectional link and the detour that is taken on the way back form a triangle. Therefore this protocol is called Unidirectional Link Triangle Routing.

ULTR is similar to the link layer tunneling mechanism proposed by the unidirectional link working group of the IETF [7], but does not require multiple interfaces on the nodes to communicate. Also, depending on the used neighborhood discovery protocol, it may even be able to work with triangles that include more than one unidirectional link, which the link layer tunneling mechanism cannot handle. Moreover, ULTR works completely on the routing layer, the link layer is not involved. This is an advantage when timeouts are used, because the extra hop and the resulting longer delay are not hidden from the routing layer.

A. Neighborhood Discovery

The neighborhood discovery protocol needed for ULTR may be quite simple and needs only be started on a node once it receives the first message from a neighbor, i.e., when the first route request message is flooded into the network. Once it has been started, the neighborhood discovery protocol should regularly transmit a message containing the IDs of all nodes from which this node has recently received messages and the status of its links to and from them. When a node receives such a hello message, it checks whether its ID is contained therein. If it is not, the receiving node knows that it is on the receiving side of a unidirectional link.

In protocols that do not use unidirectional links, a lot of overhead would now be necessary to inform the upstream node (the sender of the hello message) of the unidirectional link. In this protocol, the upstream node does not need to know about its existence. The receiving node only marks the link as unidirectional-incoming in its neighbor table.

When a node A receives a hello message via the bidirectional link from node C in which the upstream node of the unidirectional link is listed and the link to that node (from C to B) is marked as bidirectional, node A enters the sender of the hello message (node C) as a forwarding neighbor into the corresponding neighbor table entry (for node B). Please note that this would also be possible if there was a unidirectional link from C to B, but the proactive detection would introduce a large overhead and solve only one special case: If there is a unidirectional link from C to B and no other neighbor of A has a bidirectional link to B.

When a message (RREP or DATA) is sent the reverse way, it needs to be forwarded along a one-hop-detour. This message may be used to inform the upstream node of the link, which is then entered into the upstream node's neighborhood table as unidirectional-outgoing. Please note that for the routing alone this information would not be necessary, indeed it would be easy to hide the fact that the message has taken a detour. But for the sake of timers that may be used for retries on MAC, routing or transport layer it helps to know that the delay could be twice as high. In this case the information about this special link may be acquired "for free" and could be used to solve the problem described above. The information about the unidirectional-outgoing link may also be used by the MAC layer not only for retries, but also to determine the correct two-hop neighborhood of a node, which is a mandatory information for TDMA protocols.

B. Message Types

ULTR uses three message types: Route Request, Route Reply and DATA. Figure 2 shows an example for each of them:

A Route Request message contains the identity and sequence number of the source which are used for duplicate detection, followed by the identity of the destination. The hop count is incremented by one on each hop as usual, and the identity of the last hop is used to build the backward route. A Route Reply message contains sequence number and identity of the source for duplicate detection as well as the identity of the destination. For forwarding purposes the next hop and, if necessary, the forwarding node are included. The DATA packet contains the sequence number and identity of its source as well as the identity of its destination. This is followed once again by the identities of the next hop and, if suitable, the forwarding node. The last part of the DATA message contains the application data.

 ISBN: 978-1-61208-296-7

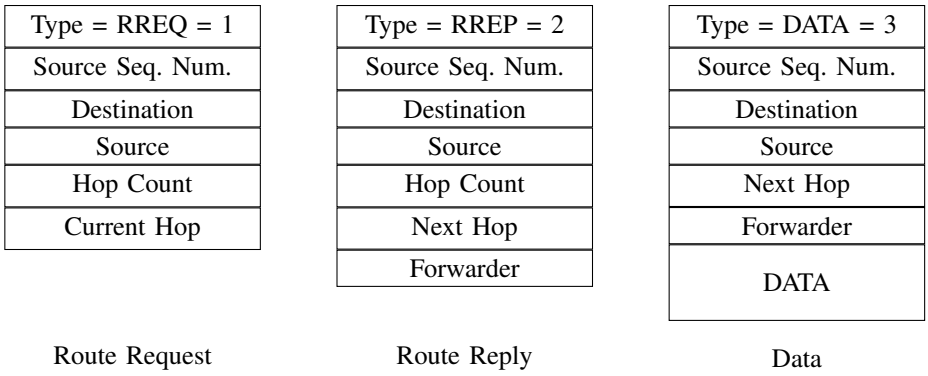


Fig. 2: Message Types Used in ULTR

C. ULTR without Neighborhood Discovery Protocol

ULTR relies on a neighborhood discovery protocol, which supplies information about incoming and outgoing unidirectional links. If neither the application nor the MAC protocol needs a neighborhood discovery protocol, a variation of ULTR with passive link detection may be used. But passive link detection means that sometimes a node does not know about links to its neighbors, even though they are available. Therefore, a second mode of operation is introduced: if a node does not have a link to the next hop in its neighbor table, it forwards the message nonetheless, with an additional flag telling its neighbors that any of them that do have an active link to the next-but-one hop (i.e., the siblings of the next hop) should also forward the message.

When this variation is used, some modifications to the message types are necessary (see Figure 3). Information about the last hop needs to be included in RREQ messages, in addition to the current hop. Both node IDs are stored in the routing table. A node decides which entry to use depending on the overheard status of the link. If the next hop is assumed to be connected by a bidirectional link, the normal next hop is used. Otherwise the message is set to detour mode and the next-but-one hop is used. The last hop is also used for implicit link detection: If a node overhears the transmission of a message in which it is denoted as last hop, it knows that the link between itself and the current hop denoted in the message is *currently* bidirectional.

A RREP message contains three node IDs instead of only two: The last hop ID and current hop ID are used to build the backward route for normal and for detour mode just as they are used in the RREQ. The next hop ID is used for forwarding. However, the RREP also contains a flag denoting the mode of transmission, which can take on the values ”normal” and ”detour”. It is evaluated upon message reception to decide whether a node shall forward the message or not. In normal mode it only forwards the message when it is denoted as the next hop in the message, in detour mode it also forwards the message if it has the next-but-one hop in its neighbor table.

The DATA message features the same three node IDs that are present in the RREP message. For routing purposes alone, the last hop ID would not be needed, but it is nevertheless included for link status detection. The mode flag is also present again, to enable the usage of a one-hop detour if the status of the next link is unknown or known to be unidirectional-incoming.

D. Cooperation with the MAC-Layer

Like all routing protocols that use unidirectional links, ULTR also needs a MAC that can transmit over unidirectional links. The information about the existence of the unidirectional links probably needs to be collected to a certain extend anyway, depending on the MAC protocol used. So either this may be retrieved from the MAC without additional cost, or the MAC protocol can query the routing layer for link information using an appropriate interface.

ULTR was designed specifically to use unidirectional links. This makes it imperative to use a MAC layer that can also transmit over unidirectional links. Any protocol that uses the standard ”request to send” - ”clear to send” mechanism is completely unsuitable, as no ”clear to send” message will ever be received over an outgoing unidirectional link. Moreover, nodes with an outgoing unidirectional link will never know that they could be disturbing the communication between two other nodes. There are some improvements that allow contention based protocols to work with unidirectional links, e.g., ECTS-MAC [8].

Some of the MAC protocols that use unidirectional links route their link layer acknowledgments back to the upstream nodes. For this, the neighborhood table used by ULTR could be reused.

Plan based MAC protocols need to know the two-hop neighborhood of each node to identify the collision domain. Within this domain, the varying parameter (e.g. frequency (FDMA), code (CDMA) or slot (TDMA)) needs to be unique for each node. Therefore, a neighborhood discovery protocol, which finds these two-hop neighbors, is needed. If the MAC protocol already has its own neighborhood discovery protocol, it only needs to make the gathered information available to ULTR.

The usage of such a neighborhood discovery protocol would also implicitly solve the ”special case” of a unidirec-

 ISBN: 978-1-61208-296-7

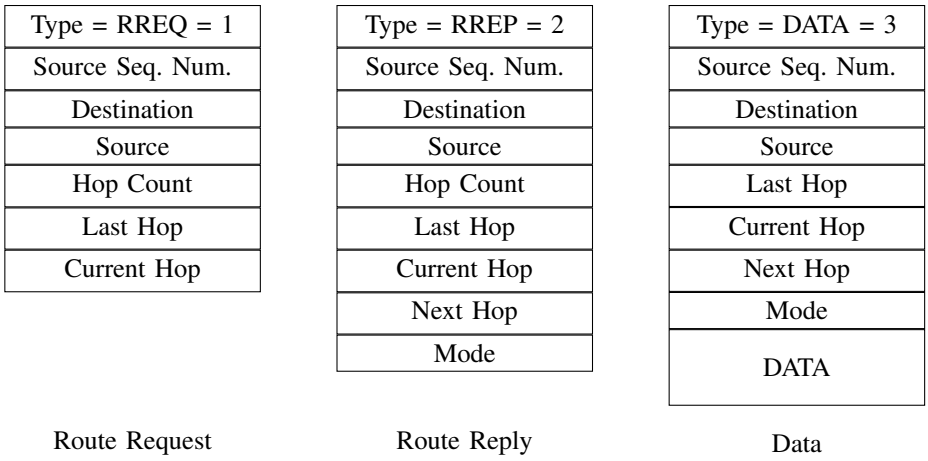


Fig. 3: Message Types in ULTR Without Neighborhood Discovery Protocol

tional link triangle with more than one unidirectional link, enabling ULTR to make use of such links as well.

This usage of a single neighborhood discovery protocol for both MAC and routing reduces communication overhead and memory consumption by far. It also ensures that both layers work on the same data. If they would use different algorithms, different storage sizes or replacement strategies, lots of problems could result, as described, e.g., in Murphy Loves Potatoes [9].

III. Evaluation

The performance of ULTR in the original form depends mainly on the quality of the link information supplied by the neighborhood discovery protocol used by the application. As it is not foreseeable which protocols will be used, the version of ULTR that uses its own passive link detection mechanism was evaluated and compared to four related work protocols: AODVBR [10], DSR [11], Tree Routing and Flooding. Tree Routing was chosen because it is still the most common routing protocol used in wireless sensor networks. DSR was chosen for its ability to handle unidirectional links, and AODVBR has an interesting way of recovering lost data messages. Flooding was also included to define an upper limit to the number of messages that may be delivered in the simulations, where the delivery ratio of a protocol is defined as the number of messages delivered by that protocol divided by the number of messages delivered by Flooding. Determining the maximum number of messages that may be delivered by Flooding was necessary, because the used connectivity model changes links between nodes often and does not guarantee that a path between two nodes exists at all.

The evaluation was split into two parts, simulations and real world experiments. The simulations were based on the discrete event simulator OMNeT++ [12] with the MiXiM [13] extension and were used to evaluate the performance of the selected routing protocols in the presence of unidirectional links, without interference from the MAC layer. Excluding the MAC layer kept the results interpretable.

As the influence of the real hardware and the MAC layer may be quite strong in a sensor network, real word experiments formed the second part of the evaluation. The real word experiments were realized using 36 sensor nodes of type eZ430 Chronos [14] from Texas Instruments and used the CCA MAC supplied by the hardware.

A. Simulation Results

The networks used in the simulations consisted of four sizes of grids: 100 nodes (10×10), 400 nodes (20×20), 900 nodes (30×30) and 1600 nodes (40×40). The grid layout was chosen to represent an application that needs area coverage, with each sensor node placed one distance unit from its direct neighbors above, below, to the right and to the left. However, as mentioned before, the connectivity between nodes was not simply determined by their distance, but rather by a certain probability, that depended only partially on the distance. Also, links changed often and unidirectional links were common in the simulations, as they have been shown to be in the real world. The results presented here are averages of more than 600 simulations for each network size.

The delivery ratio achieved by each protocol in the simulations is shown in Figure 4. It may be seen that the delivery ratio of the related work protocols strongly declines with increasing number of nodes and therefore network diameter. For Tree Routing, this may be explained by the absence of a real route maintenance mechanism. When the forwarding of a message fails due to link break or because the link turned unidirectional incoming, two retransmissions are tried. If the link becomes available again during those two retries, the message has gained one hop, otherwise it is lost. AODVBR uses an intelligent way to recover lost data messages, building a fish bone structure during route discovery. However, this mechanism is used only for data messages, once the initial route has been established. The main problem of AODVBR is this establishment of the initial route, which consists of a route request (RREQ) flooding and a route reply (RREP) transmission along the inverted path taken by the fastest RREQ. If the RREP is lost, e.g., due to the presence of a

 ISBN: 978-1-61208-296-7

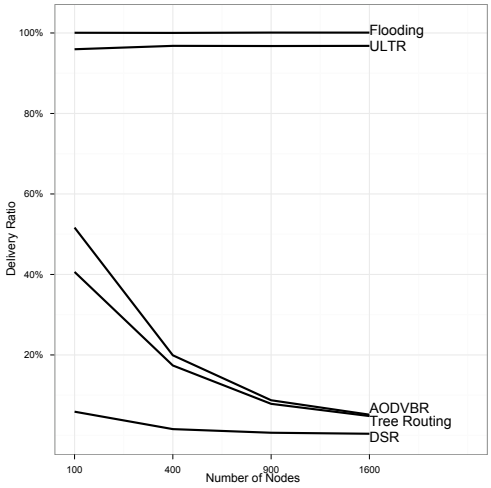


Fig. 4: Delivery Ratio achieved in the Simulations

unidirectional link on the path taken by the RREQ, no initial route may be found. This problem arises quite often due to the nature of unidirectional links, namely their longer reach which often exceeds that of bidirectional links by far. DSR can work in the presence of unidirectional links, which is one of the reasons it was chosen for comparison. However, it suffers heavily from dynamic link changes, as the route maintenance mechanism of DSR in the mode that uses unidirectional links produces a lot of overhead. ULTR on the other side delivers more than 95% of the number of messages delivered by the reference protocol Flooding.

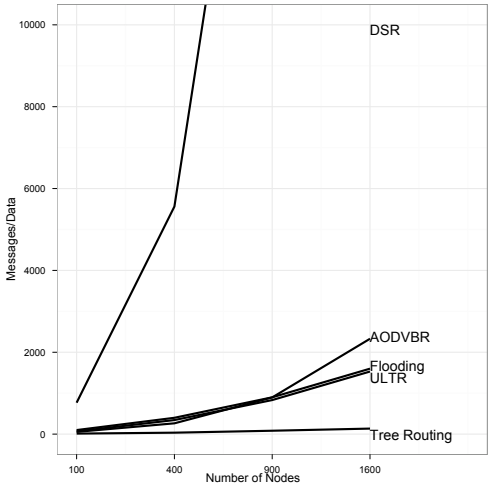


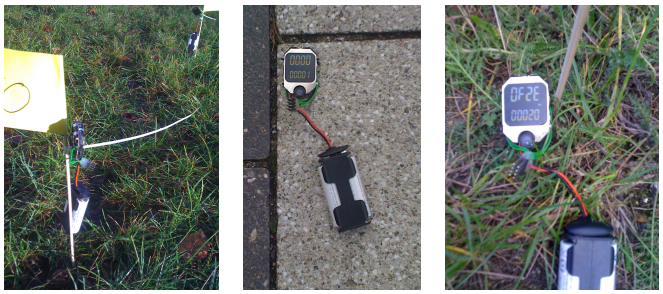
Fig. 5: Number of Messages transmitted to deliver a single Application Message in the Simulations

The efficiency of the protocols measured in messages transmitted in order to deliver a single application message is shown in figure 5. The low number of delivered messages and extremely high number of transmitted messages absolutely disqualify DSR for the simulated type of network with its often changing links. Interestingly, Tree Routing shows the best performance when only the network load is considered. This is due to its simplicity and the low cost of transmission failure: Where other protocols use complex mechanisms to repair the broken route, Tree Routing only uses its two retransmissions, resulting in a maximum cost of 3 times n transmissions, where n is the route length. For comparison: An unsuccessful forwarding of a message in DSR results in a route error message being transmitted to the originator of the message, which then starts a new route discovery by flooding the network with RREQ messages. Once one of these reaches the destination, the network is flooded again with the route reply, due to the operation in unidirectional link mode.

When considering only this figure, Tree Routing seems to be the optimal choice. However, as shown earlier, it has a very low delivery ratio for larger networks. Therefore, it should only be used in networks with a small diameter and when the network load is more important than the delivery of all messages.

B. Real World Experiments

For the real world experiments, 36 sensor nodes were placed in a grid of 6 times 6 nodes in four different locations: On a desktop, affixed to poles, placed on a lawn and placed onto a stone pavement. The transmission power was set to 0dBm.



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

Fig. 6: A modified eZ430-Chronos Sensor Node

The desktop placement is a one-hop environment, where each node was able to communicate directly with each other one. In the pole placement (figure 6(a)) the nodes were fixed to poles using cable binding, at a height of approximately 20 centimeters above ground with a distance of one meter between nodes, resulting in route lengths between 1 and 2 hops. The stone (figure 6(b)) and lawn (figure 6(c)) placements also used a distance of one meter between nodes, but resulted in route length of up to 5 hops due to the shorter reach of nodes placed on the ground.

Figure 7 shows the delivery ratio of each protocol for the real world experiments, sorted by placement. In the pole placement, AODVBR, DSR and Tree Routing achieved roughly the same delivery ratio. Flooding performed worse due to the high number of messages generated and the resulting MAC layer problems. ULTR performed worst with a delivery ratio of 59%. This is due to problems with the passive link detection, which does not work well in networks with a low diameter. Please note that in the real world experiments 2100 application

 ISBN: 978-1-61208-296-7

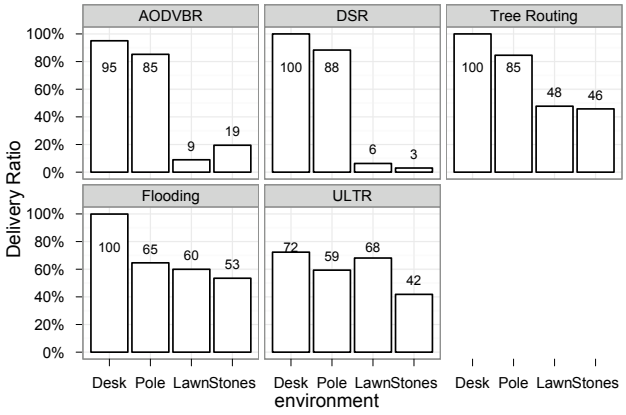


Fig. 7: Delivery Ratio achieved in the Real World Experiments

messages were generated and the delivery ratio was defined as the number of messages delivered by a protocol divided by 2100.

In the lawn and stone experiments, the influence of the route maintenance was stronger, as routes broke more often due to the increase of the average route length. It may be seen once more that the route maintenance mechanism of DSR is not usable in highly dynamic scenarios. AODVBR suffers from the higher probability of having a unidirectional link within its initial routes and Flooding produced too many messages, leading to many collisions. Tree Routing delivered 46 to 48 percent of messages, which may be explained by the node topology. Nodes close to the sink, i.e., those within two hops, were able to deliver their messages most of the time due to the two transmission retries. Those further off were able to deliver only seldom. ULTR performs best in the lawn placement, and nearly equal to Tree Routing in the stone experiments.

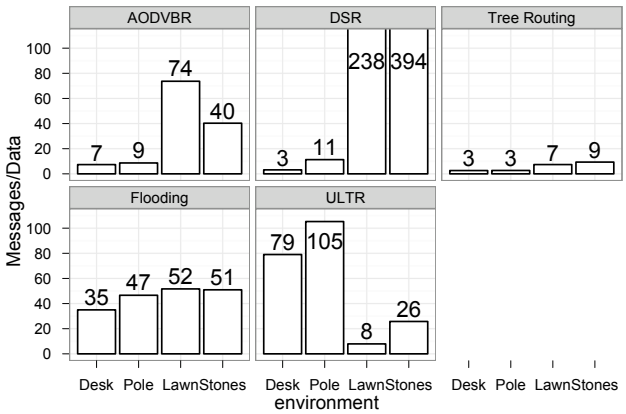


Fig. 8: Number of messages transmitted to deliver a single application message in the Real World Experiments

The cost of delivering an application message measured in transmitted messages is shown in figure 8. As seen in the simulations, Tree Routing is very efficient when the number of messages is considered. This fact is also apparent in all real world experiment placements. When the network load is the limiting factor, Tree Routing should therefore be used, even though it does not reach 50% delivery ratio in the lawn and stone experiments. The other end of the spectrum may be seen in DSR, which transmits 283 and 394 messages per delivered data message in the lawn and stone scenarios respectively. ULTR in its current version should not be used for single hop or 1 - 2 hop scenarios, as the passive neighborhood detection only starts to work in larger networks. There, the performance of ULTR is better than that of all related work protocols, except for Tree Routing. In the evaluated scenarios, ULTR would be the protocol of choice for the lawn placement, and Tree Routing should be used on the stones. The simulations have shown, however, that Tree Routing runs into strong problems when the number of nodes increases, meaning that for larger networks ULTR should be used in both placements, as it scales much better with the network diameter.

C. Importance of Timeouts

The implemented version of ULTR with passive link detection is heavily dependent on the timeouts that are used for the links. If it is set too low, the links are deleted before they may be used, even though they might still exist, resulting in a local broadcast on every hop. If it is set too high, links are assumed to exist, but have broken a long time ago.

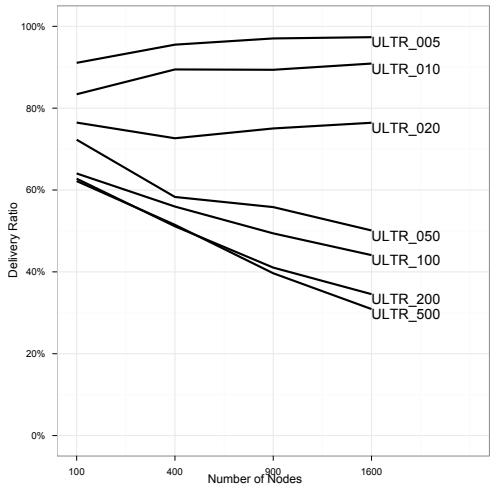


Fig. 9: Delivery Ratio achieved in the Simulations for different timeouts

The implementation of ULTR uses a timer that fires every 100 ms, and has a parameter called `linkTimeout` that defines how many times that timer must fire before a link is removed from the neighbor table. The results presented above were achieved with a `linkTimeout` of 5, and resulted in a lot of message transmissions but also fairly high delivery ratio. To quantify the impact of the `linkTimeout`, the performance of ULTR was measured with different values of `linkTimeout`: 5,10,20,50,100,200 and 500.

Figure 9 shows the delivery ratio achieved by ULTR with the seven different timeouts. It seems that the delivery ratio

 ISBN: 978-1-61208-296-7

is constantly decreasing with increasing timeout lengths. This is not surprising, since a link that has been removed from the neighbor table results in a local broadcast. All nodes that receive this message and know the intended next-but-one hop retransmit the message, adding a lot of redundancy. Therefore, removing a link too early does not result in message loss, but in unnecessary network load. However, if the link is deleted too late, i.e., a link is assumed to exist where it has already broken, the message gets lost. Therefore, when considering only the delivery ratio, using a small timeout seems favorable.

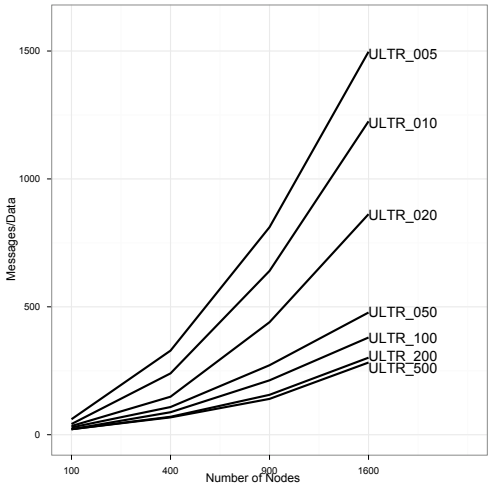


Fig. 10: Number of messages transmitted to deliver a single application message in the Simulations for different timeouts

However, when the network load is considered, the choice seems to be quite the opposite. Figure 10 shows the cost of delivering a single application message measured in transmitted messages. When using the smallest timeout of 5, about 1500 messages are transmitted for each application message delivered in the network consisting of 1600 nodes, which is quite close to the cost of flooding the message. Therefore, the decision which timeout should be used is a tradeoff between delivery ratio and network load. However, there are limits to the choice: Increasing the timeout above 200 does not change delivery ratio or efficiency much. Also, as the delivery ratio is most often more important than the network load, it is unlikely that a timeout of more than 50 would be used, because higher timeout values lead to a delivery ratio of less than 50%. Still, even this is much more than what the related work protocols achieved, making ULTR a fine choice for the evaluated network types.

IV. Related Work

LRS, a link relay service, is proposed in [15]. The authors require all nodes to transmit messages containing their ID and a sequence number regularly, to detect incoming neighbors. Nodes that receive those messages answer with a message of their own, enabling the detection of bidirectional links. For unidirectional or asymmetric links, LRS is used, which floods the messages over a specified number of hops. The main difference between LRS and ULTR is that ULTR uses a chosen forwarding node instead of flooding over multiple hops to circumvent unidirectional links. In the second mode, when neighborhood information is only gathered passively, ULTR does not need to detect neighbors actively as LRS does. Moreover, when a unidirectional link needs to be passed, only nodes that have an active link to the intended next hop forward the message, which once more reduces the network load compared to LRS.

The authors of DEAL [16] describe mechanisms which may be used to detect and exploit asymmetric links in dense wireless sensor networks. They introduce *Source Specific Relay* (SSR), which is used to select a neighboring node X as forwarder, when a link between two nodes A and B is asymmetric. An enhancement, called *Dynamic Driven Maintenance* (DDM) is also described. DDM uses a combination of SSR and broadcast mechanisms to react to changes in the nature of asymmetric links. The third contribution of [16] is called *Asymmetry-Aware Caching* (AAC) and deals with memory requirements on sensor nodes. Due to the limited memory, neighbor tables need to be restrained to a fixed size, leading to eviction of nodes if more neighbors than neighbor table entries are available. Choosing the right entry to evict is far from simple and AAC is used to make this decision, but the exact mechanism is not specified. DEAL is concerned with the detection of asymmetric links on the link layer, ULTR makes use of these links on the routing layer. Moreover, ULTR can also use unidirectional links, which DEAL cannot detect. Finally, ULTR with passive detection can even operate completely without neighborhood management, removing the costs of link detection.

Try-Ancestors-Before-Spreading (TABS) [17] uses any nodes that are closer to the root of a routing tree as forwarders, when the direct parent of a node did not forward the message. To realize this, a node stores a message it has forwarded until the retransmission by its parent node is overheard or a timeout expires. If the timeout expires the message is retransmitted, allowing all nodes that are closer to the sink by a so called *minimum progress limit* to forward the message, regardless of their parent-child relation. The *minimum progress limit* is lowered with each consecutive retransmission, finally reaching a value of zero, indicating that even siblings of the transmitting node may forward the message. Even though this approach increases delivery ratio, it is still necessary for nodes to store messages and wait for acknowledgments. These acknowledgments may only be received if the links are bidirectional, resulting in unnecessary retransmissions if the links are unidirectional. ULTR can use unidirectional links directly, if a neighborhood protocol is provided. If no neighborhood protocol is provided and ULTR uses only passive detection, the forwarding mechanism enables the implicit usage of unidirectional links. Also, no explicit acknowledgments are used in ULTR, reducing the cost further. Moreover, the implicit usage of unidirectional links enables ULTR to work in environments with low link stability, whereas TABS needs fairly stable links for the routing tree. This is also reflected in the used testbed: The changing power supply of battery powered nodes lead to frequent link changes in our testbed, whereas the USB powered testbed used for TABS had a constant power supply, resulting in much more stable links.

ABVCap_Uni [18] uses virtual coordinates to enable geographic routing in sensor networks without geographic information. The authors claim that *ABVCap_Uni* enables the

 ISBN: 978-1-61208-296-7

usage of unidirectional links through definitions of clusters and rings. The overhead of building these clusters and rings is high, though. The simulation used for evaluation did not feature any message losses, and all links were static, leading to a delivery ratio of about 69 - 87%. If the network load induced by constant rebuilding of the clusters and rings would have been included, the delivery ratio would decrease further. As shown in Section III, ULTR was evaluated both in simulations and on a real sensor node testbed.

ieARQ and *E-ieARQ* are introduced in [19]. The authors are concerned with acknowledgment losses due to asymmetric links, which in turn lead to unnecessary retransmissions and a waste of energy. They show that implicit ARQ can lead to an avalanche effect and propose two enhancements of ARQ, which remove the avalanche effect and reduce power consumption by adding an explicit acknowledgment if the implicit one was lost. This approach does not take unidirectional links into account, though, in which case a direct transmission of an acknowledgment is impossible. ULTR operates without acknowledgments or retransmission, removing the avalanche effect completely.

V. Conclusion

Unidirectional links represent a huge problem for routing protocols in wireless networks and especially in sensor networks. Even though experiments show that unidirectional links are quite common, most of today's routing protocols are not able to make use of them. In this paper we introduced ULTR, a routing protocol for wireless sensor networks with often changing and unidirectional links. We presented two versions of ULTR, one which depends on a neighborhood discovery protocol to supply link information and one that uses passive link detection. The former should be used when a neighborhood discovery protocol is included on the sensor nodes anyway. This could be the case when a TDMA MAC is used, which needs to know its two hop neighborhood, when the application needs connectivity data or when a monitoring software is included, which also monitors links and link changes. The latter version, the one that uses passive link detection, should be used when no other source of connectivity data is available.

We evaluated ULTR both in simulations and in real world experiments with eZ430-Chronos sensor nodes from Texas Instruments and compared the performance of ULTR to that of Tree Routing with retransmissions, DSR and AODVBR. The results show that ULTR is much better suited to an environment with often changing links and unidirectional links than the protocols chosen for comparison.

References

[1] 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.

[2] L. Sang, A. Arora, and H. Zhang, "On exploiting asymmetric wireless links via one-way estimation," in *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM Press, 2007, pp. 11–21.

[3] Turau, Renner, and Venzke, "The heathland experiment: Results and experiences," in *Proceedings of the REALWSN'05 Workshop on Real-World Wireless Sensor Networks.*, Jun 2005. [Online]. Available: citeseer.ist.psu.edu/732032.html

[4] 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.

[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*, ser. MobiHoc '02. New York, NY, USA: ACM, 2002, pp. 12–23. [Online]. Available: http://doi.acm.org/10.1145/513800.513803

[6] 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.

[7] E. Duros, W. Dabbous, H. Izumiyama, N. Fujii, and Y. Zhang, "A link-layer tunneling mechanism for unidirectional links, http://www.faqs.org/rfcs/rfc3077.html," Mar 2001.

[8] S. Mank, R. Karnapke, and J. Nolte, "Mac protocols for wireless sensor networks: Tackling the problem of unidirectional links," in *International Journal on Advances in Networks and Services, vol 2 no 4*, 2009, pp. 218 – 229.

[9] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture," in *Proc. 14th Intl. Workshop on Parallel and Distributed Real-Time Systems (WPDRTS)*, Apr. 2006.

[10] 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. [Online]. Available: citeseer.ist.psu.edu/lee00aodvbr.html

[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." [Online]. Available: https://tools.ietf.org/html/rfc4728

[12] A. Varga, "The omnet++ discrete event simulation system," in *Proceedings of the European Simulation Multiconference (ESM'2001)*, Prague, Czech Republic, Jun. 2001.

[13] 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 *1st Int. Workshop on OMNeT++*, mar 2008. [Online]. Available: http://www.st.ewi.tudelft.nl/ koen/papers/mixim.pdf

[14] "Texas instruments ez430-chronos, http://focus.ti.com/docs/toolsw/folders/print/ez430-chronos.html?DCMP=Chronos&HQS=Other+OT+chronos." [Online]. Available: http://focus.ti.com/docs/toolsw/folders/print/ez430-chronos.html?DCMP=Chronos&HQS=Other+OT+chronos

[15] J. Du, W. Shi, and K. Sha, "Asymmetry-aware link quality services in wireless sensor networks," in *Proceedings of the 2005 international conference on Embedded and Ubiquitous Computing*, ser. EUC'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 745–754. [Online]. Available: http://dx.doi.org/10.1007/11596356_74

[16] B. B. Chen, S. Hao, M. Zhang, M. C. Chan, and A. L. Ananda, "Deal: discover and exploit asymmetric links in dense wireless sensor networks," in *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*, ser. SECON'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 297–305. [Online]. Available: http://dl.acm.org/citation.cfm?id=1687299.1687333

[17] J. Faruque and A. Helmy, "Tabs: Link loss tolerant data routing protocol for multi-hop wireless sensor networks," *Sensor Networks, Ubiquitous, and Trustworthy Computing, International Conference on*, vol. 0, pp. 11–18, 2010.

[18] 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.

[19] R. P. Liu, Z. Rosberg, I. B. Collings, C. Wilson, A. Y. Dong, and S. Jha, "Overcoming radio link asymmetry in wireless sensor networks," in *PIMRC'08*, 2008, pp. 1–5.

 ISBN: 978-1-61208-296-7