MAC Protocols for Wireless Sensor Networks: Tackling the Problem of Unidirectional Links

Stephan Mank, Reinhardt Karnapke, Jörg Nolte
Distributed Systems/Operating Systems Group
Brandenburg University of Technology
Cottbus, Germany
{*smank, karnapke, jon*}*@informatik.tu-cottbus.de*

Abstract—**Experiments have shown that unidirectional links are quite common in wireless sensor networks. Still, many MAC protocols ignore their existence, even though they have a tremendous impact on the performance of both TDMA- and contention based protocols. In contention based protocols the medium may be assumed free when it is indeed busy. In TDMA based protocols two neighboring nodes might get assigned the same slot even though there is an unidirectional link between them. In this paper we discuss the influence of unidirectional links on communication protocols in wireless sensor networks, focusing on MAC protocols. We also present two protocols that do not only eliminate the negative side effects of unidirectional links, but use them for message transmission as well.**

Keywords-**Wireless Sensor Networks; MAC Protocols; Unidirectional Links**

I. Introduction

Wireless sensor networks are collections of small sensing and computation units that can cooperate with each other using over the air communication. Since these networks shall be deployed on a large scale (i.e. hundreds of nodes), the overall cost often dictates the usage of cheap radio transceivers. Many of these transceivers do not only lack hardware support for medium access control, their huge number also makes it near to impossible to calibrate all of them exactly the same, resulting in many differences in antennae characteristics. Due to these differences, which are also enhanced by the difference in orientation of the deployed nodes, a lot of unidirectional links (node A can send to node B but not vice versa) are introduced into the sensor network right from the beginning. Differences in height of position are also an influencing factor. After the sensor network is started, dynamic effects like atmospheric changes, animals walking by or people using electrical devices lead to an often changing radio neighborhood. These changes can be a complete breakage of links, or the transformation from a unidirectional to a bidirectional one and vice versa. Sometimes a unidirectional link changes its direction.

Most of todays sensor networks are meant to deliver the gathered data in one form or another to a sink for evaluation. But this requires multihop transmissions along changing routes. Finding a suitable route is the task of routing protocols, the MAC protocol only needs to supply one hop communication. Unidirectional links are a common phenomenon on both protocol layers - most routing protocols try to eliminate the negative effect they have on their routing choices, only some of them try to utilize them. MAC protocols face a harder problem, as the effect of a unidirectional link may not only be a wrong choice, but a lot of collisions leading to a bad channel utilization and packet loss.

In this paper we present the influences of unidirectional links on both protocol layers, and describe a way of increasing network connectivity, reliability and lifetime by using the unidirectional links in addition to the bidirectional ones in MLMAC-UL (TDMA-based) and ECTS-MAC (contention based), both presented first at Sensorcomm 2009 [1]. The effectiveness for both different approaches is shown in simulations as well as in experiments with TMote Sky sensor nodes.

This paper is structured as follows: Section II takes a closer look at unidirectional links, their occurrence in wireless sensor networks and their impact on routing and MAC protocols. Section III describes related work, while sections IV and V describe the protocols MLMAC-UL and ECTS-MAC respectively. In section VI our two protocols are evaluated, both with simulations and experiments on real sensor network hardware. We finish with conclusion and future work in section VII.

II. The Nature of Unidirectional Links

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

The authors of [2] describe an experiment they conducted in the Lüneburger Heide. The original aim was to evaluate a routing protocol, which is not characterized further in the paper. Rather, the observations they made concerning the properties of the wireless medium are described, focusing on the frequency of changes and the poor stability of links. These experiments were conducted using 24 Scatterweb ESB [3] sensor nodes, which were affixed to trees, poles etc, and left alone for two weeks after program start. One of the duties of the network was the documentation of the logical topology (radio neighborhood of nodes), which was evaluated by building a new routing tree every hour, e.g. for use in a sense-and-send application. The neighborhood was evaluated using the Wireless Neighborhood Exploration protocol (WNX) [2], which can detect unidirectional and bidirectional links. Once this was done, all unidirectional links were discarded and only the bidirectional ones were used to build the routing tree. Figure 1a shows one complete communication graph obtained by WNX, while figure 1b shows the same graph without unidirectional links, where a lot of redundant paths have been lost by the elimination. In fact, one quarter of the nodes are only connected to the rest of the network by a single link when unidirectional links are removed. If this single link breaks, the nodes become separated, even though there are still routes available. Thus, the removal of unidirectional links increases the probability of network separation severely.

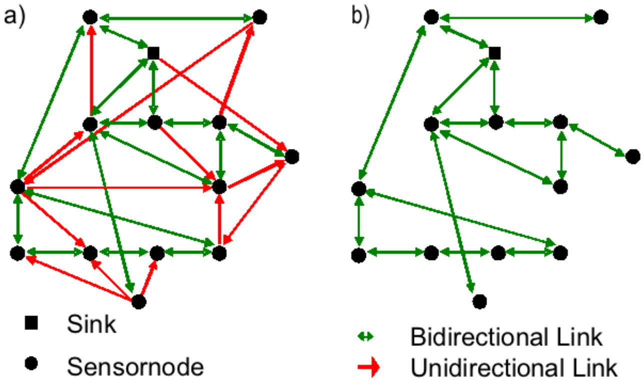

Figure 1. A Communication Graph from [2] (Presentation) [4]

The authors of [5] hold a similar view. They evaluate the three kinds of links (asymmetric, unidirectional, bidirectional) using protocols like ETX (Expected Transmission Count) [6]. These protocols search for reliable links, but most focus on bidirectional ones. This leads to the fact that a link with a reliability of 50% in both directions is chosen above one with 100% from node A to node B and 0% from B to A. If data needs to be transmitted only from A to B without need for acknowledgment, this choice is obviously wrong. To prevent this wrong choice, the authors of [5] propose a protocol called ETF (Expected Number of Transmissions over Forward Links), which is able to use unidirectional links. They also show that the reach of reliable unidirectional links is greater that that of reliable bidirectional links. In experiments with XSM motes [5] 7 times 7 nodes were placed in a square, with a distance of about 1 meter between nodes. In four sets of experiments at different times of day each node sent 100 messages at three different power levels. Then the packet reception rate was recorded, which is defined for a node A as the number of packets A received from a node B divided by the number of messages sent (100). Then the packet reception rates of nodes A and B are compared. If the difference is less than 10%, the link is considered bidirectional. If it is more then 90% the link is considered unidirectional. The XSM nodes offer 9 different transmission strengths, of which three were evaluated: the lowest, the highest and the third in between. Table I shows the results of the experiments.

Table I
Link Quality versus Transmission Strength

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

The results show that even when using the maximum transmission strength 12% of the links would have been discarded by ETX (Expected Transmission Count) [6] and similar link quality evaluation protocols that focus only on bidirectional links. As the lifetime is one of the major optimization goals in a sensor network and receiving/transmitting consumes a lot of energy, it is rather uncommon to have all nodes constantly transmit using the highest transmission strength. In fact, current research projects like e.g. [7] try to minimize power consumption by adjusting the transmission strength depending on the required reach and reliability.

The observations of [5] are concluded in three points:

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

A sensor network which monitors water pumps within wells is described in [8]. The sensors were used to monitor the water level, the amount of water taken and the saltiness of the water in a number of wells which were widely distributed. The necessity for this sensor network arose because the pumps were close to shore and a rise in saltiness was endangering the quality of the water. The average distance between wells was 850 meters and the range of transmission was about 1500 meters. Communication was realized using 802.11 WLAN hardware both for the nodes as well as for the gateway. For data transmission between nodes Surge_Reliable [9] was used, which makes routing decisions based on the link quality between nodes.

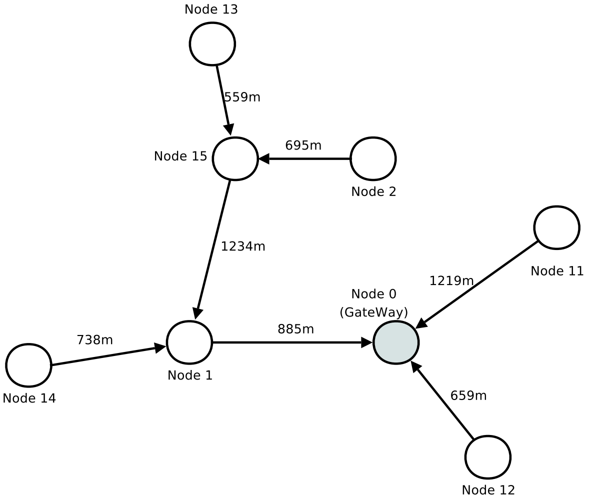

Figure 2. A Communication Graph from [8] that Follows the Theory

During the experiments the authors observed, that the (logical) topology of the network changed dynamically, even though all nodes were stationary. The authors claim that these changes were probably due to antenna size and changes in temperature and air moisture. In this context it is important to remember that the distance of nodes was far below the range of the transmitters (about 50%). While about 70% of the routing trees observed followed the theory (figure 2), there were a lot of strange exceptions. In one case the average distance between connected nodes even rose to 1135 meters, as nodes that should have been able to communicate directly with the gateway were connected to nodes on the far side instead. In one of these routing trees (figure 3), a single node had to take care of all communication with the gateway, even nodes that were on the other side were using it as next hop. The reason for this is that Surge_Reliable chooses the nodes with the best link quality, but only considers bidirectional links. If unidirectional links could have been used, the results could have been quite different.

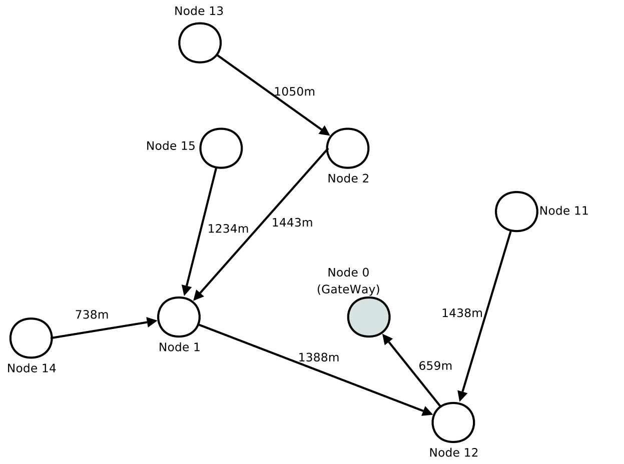

Figure 3. A Communication Graph from [8] with Long and Strange Links

VigilNet, a military sensor network for terrain surveillance is described in [10]. This project aims at the detection of moving vehicles using magnetic sensors attached to Mica2 sensor nodes. The transport of messages from the nodes to the sink was realized using a diffusion based algorithm, similar to Directed Diffusion [11], which produced a routing tree with root at the sink. To eliminate unidirectional links, a protocol called Link Symmetry Detection was developed. Each node periodically transmitted the list of its neighbors. A node that received such a neighbor list checked the list to determine if it was mentioned. If it was not, the link was an incoming unidirectional one. When building the routing tree after deployment, the transmission power of all nodes was halved. Now all nodes determined their parent node from the neighbor lists received with this half strength. At the end of this setup phase, all nodes switched to full transmission power. The intention behind this scheme was to ensure that the connection to the father node would not break. During the experiments, the authors noted that asymmetric links were far more common than expected. They put this fact down to differences in hardware, as the transceivers were not calibrated before the experiment. Another interesting effect seen in these experiments is that only about 2/3 of all nodes were able to communicate directly with the sink, because only bidirectional links were used.

III. Related Work

The problem of unidirectional links has been recognized before, and protocols have been developed which can use them.

The Multicast MAC protocol (MMP) [12] does not directly address the problem of unidirectional links, but it offers an easy way to realize a multicast communication, which can easily be increased to broadcast. BMMM [13] and Maclayer Multicast [14] follow a similar approach. MMP is an extension of the IEEE 802.11 MAC in DCF mode. The Request To Send (RTS) message of MMP contains the addresses of all nodes that should receive the multicast message. When a node receives this RTS, it waits a certain time, correlating to its position in the RTS, and sends a CTS. When the slots for all CTS messages have passed and the sender of the RTS has received at least one CTS, it begins

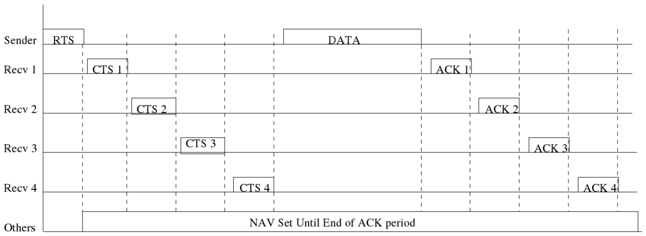

Figure 4. Message Propagation in MMP [12]

transmission of the data packet. After the transmissions, the acknowledgment messages are send by all of the receivers in the same order as the CTS messages (figure 4). While MMP needs to wait a time corresponding to the number of nodes addressed in the RTS message before sending data packets, the proposed ECTS-MAC waits only the time needed for a single ECTS message, thus providing much better scalability. Also, the size of the RTS is reduced drastically in ECTS-MAC, because the list of receivers is omitted (see section V).

AMAC [15] is built on top of the Sub Routing Layer (SRL) project [16], which is used to detect unidirectional links. When SRL is used with a routing protocol, it provides the abstraction of a network with only bidirectional links. To do this, it must identify unidirectional links, and find a suitable reverse route leading through multiple nodes. This is done using a reverse distributed Bellman-Ford algorithm. SRL also monitors the network for link changes. AMAC uses the information from SRL to make unidirectional links usable on the MAC layer. Four new types of messages are introduced to make communication over unidirectional links possible by forwarding protocol messages through neighboring nodes. AMAC uses a complex formula to identify the right nodes to forward all four types of messages, while the transmission of ECTS-messages in ECTS-MAC is done probabilistic. It defines 4 new messages: XRTS (Extended RTS), XCTS (Extended CTS), TCTS (Tunneled CTS) and TACK (Tunneled ACK). XRTS and XCTS are used to inform nodes about the communication that could normally not receive RTS and CTS, but which may still disturb the transmission because of their long communication range. The TCTS is sent by the destination of an RTS message if it was received over an unidirectional link. In this case direct sending of a CTS is not possible, therefore the TCTS must be forwarded by a neighboring node that can communicate with both participants of the communication (tunneled). Once the communication is complete, the destination sends a TACK message which is again tunneled for the same reason.

Another extension to IEEE 802.11 is BW_RES [17]. It is based on the principle of forwarding CTS packets to all nodes that may disturb the planned communication. To determine how far a BW_RES message must be forwarded, the transmission strengths of all nodes must be known. The lowest one equals one unit, the highest one N units. The authors show that a CTS message needs to be retransmitted 2N-1 times to ensure that it is heard at least N units distant. A node that receives a CTS message waits between 0 and 6 SIFS before transmitting the BW_RES packet to prevent collisions (figure 5). While this approach ensures that data communication in the presence of unidirectional links is possible, it delays the transmission and increases the network load proportional to the maximum difference in transmission strengths of nodes. In comparison, the network load produced by ECTS-MAC is rather low, depending on the chosen probability.

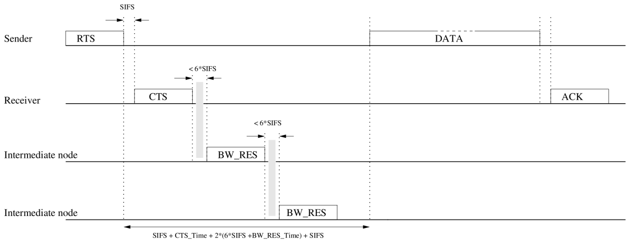

Figure 5. Message Propagation in BW_RES [17]

PANAMA (Pair wise Link Activation and Node Activation Multiple Access) [18] consists of two different algorithms. PAMA-UN (Pair wise link Activation Multiple Access Unidirectional Networks) is intended for unicast communication, while NAMA-UN (Node Activation Multiple Access for Unidirectional Networks) supplies broadcast communication. PANAMA is based on CDMA (Code Division Multiple Access) and uses DSSS (Direct Sequence Spread Spectrum). Also, Time is divided into slots. In each slot, nodes with orthogonal spread codes can transmit simultaneously. Codes are reassigned every slot, nodes compete for the codes by comparing their priority. The node with the highest priority has won the medium and all its neighbors configure their radio modules to use its spread code. The link characteristic (bidirectional or unidirectional) is a part of the bandwidth value which is featured in the computation of the priority. The main difference between NAMA-UN and PAMA-UN is the way priorities are computed. In NAMA-UN, the priority depends on the sending node, whereas in PAMA-UN it is calculated using all incoming links of both nodes participating in the communication. The most complex part of PANAMA is the calculation of priorities. Each node needs to know the exact priorities of all its neighbors at any time. It is 0 If the bandwidth from the sender to the receiver is 0 (unidirectional link from this node to its neighbor). A node wins the contention if its priority is higher than that of all its neighbors and there is no upstream-only-neighbor (neighbor with a unidirectional link to this node) that uses the same spread code. The priority of all neighbors

k in slot t is calculated as follows: $p_k^t = \sqrt[bw_k]{Rand(k+t)}$ where bw_k is the bandwidth of node k. $Rand$ is a random function which delivers a number between 0 and 1. $p_k = 0$ if $bw_k = 0$ In PAMA-UN the computation of the priority depends on all incoming links of both participating nodes x,y,: $p_{(x,y)}^t = \sqrt[bw_{(x,y)}]{Rand(x+y+t)}$. Both protocols, PAMA-UN and NAMA-UN depend on knowledge about the 2-hop neighbors of a node. To determine this, a neighborhood protocol is used, which transmits updates about the neighborhood of a node regularly. Each node can compute its 2-hop neighborhood by combining these messages from all its 1-hop neighbors. The update messages can contain information about multiple links. This information contains the ID of the neighbors, the status of the link (bidirectional or unidirectional), the type of change (add or delete a link/neighbor) and the current bandwidth. Depending on the rate of mobility the interval at which these messages are sent can be adjusted.

IV. MLMAC-UL

In previous work we introduced MLMAC [19], [20], a TDMA based MAC protocol for mobile wireless sensor networks. MLMAC divides time into frames, which are in turn divided into slots. Each node may use its own slot to transmit data to its neighbors, a slot reappears each frame. Nodes which have a common neighbor must have different slots to prevent collisions. For static networks it is fairly easy to find a schedule for all nodes that fulfills this property, for mobile nodes it is much harder. MLMAC uses an adaptive approach to enable each node in the sensor network to allocate a slot. In this approach there is no predefined starter node as in LMAC [21], rather the synchronization of nodes is started by the node that wants to transmit something first.

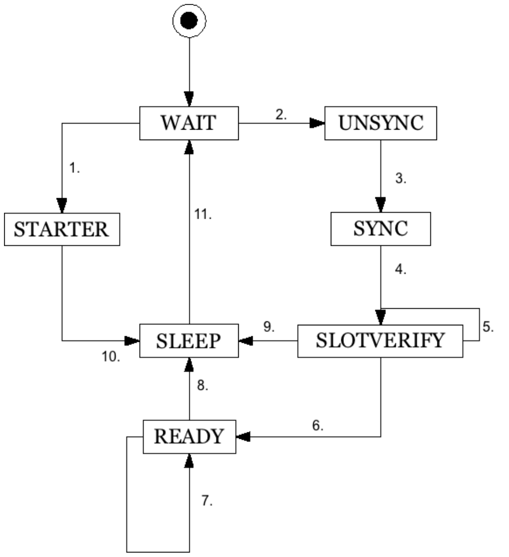

Figure 6. The Finite State Machine used in MLMAC [19]

In MLMAC a node may have one of 7 different states, and transitions from one to the other under certain conditions. The complete state-machine can be seen in figure 6. When nodes are first activated, MLMAC starts in the `WAIT`-state.

1) This node wants to transmit a message. It starts a global synchronization.
2) A control message from a node in `STARTER`- or `READY`-state was received. Synchronize local time.
3) After listening for one frame, choose a slot.
4) This nodes slot is active. Start transmitting a control message every frame in this slot
5) A collision seems to have occurred but the control message was received on a unidirectional link.
6) No collision occurred in the last frame.
7) A collision seems to have occurred but the control message was received on a unidirectional link.
8) A collision occurred and the link to the sender is bidirectional. Delete slot information.
9) A collision occurred and the link to the sender is bidirectional. Delete slot information.
10) A control message with a different, older synchronization was received. Remove all slot information.
11) After waiting for a certain time, return to the beginning and start again.

Most important for this work are the `ready`-state and the transition to the `sleep` state.

A node that has reached the `ready`-state is in a stable state, as long as no error occurs. If a collision occurs, the link from the sender is checked. If it is bidirectional, the node transitions into the `sleep`-state, because that is obviously wrong and a new slot has to be chosen. If the link is unidirectional, the node remains in the `ready`-state. The determination, whether a link is unidirectional or bidirectional is realized with a simple counter. Whenever transmissions are expected but not received, this counter is changed. After a predetermined number of missed messages, the link is considered unidirectional. This method has proven to be too ineffective for our purpose.

In this section we introduce the changes we made to MLMAC, to stop only detecting unidirectional links and ignore collisions that occurred because of them. Rather, MLMAC-UL uses a neighborhood discovery protocol to determine neighbors that can be used to inform the originator of a unidirectional link (the node that can be heard by the other one) about the link and make it usable to forward messages.

The first addition is an independent neighborhood discovery protocol, which is similar to the ones used in AMAC [15] and PANAMA [18] (see Section III). It transmits the neighborhood table of a node periodically but seldom. In the case of changes, only small update messages are sent. The periodic sending of tables is used to remove any errors resulting from loss of update packets.

Another change in MLMAC-UL is the fact that nodes can give up their slots. If a node has transmitted only status messages for a certain time (e.g., 6 frames) it will inform its neighbors that it is giving up the slot and that it may be used by another node. This is done by altering the status message a node transmits at the beginning of its slot. Moreover, a node may not only hold one slot in MLMAC-UL. Rather, each node can use as many slots as it needs by claiming any unused ones, when it has to transmit lots of data. Once the send queue is emptied, it can give the additional slots up one after the other. For this to be effective it is useful to define a larger frame size from the beginning, so that there are always enough free slots available (figure 7). This ability to hold more slots was introduced to reduce the delay and make MLMAC a better competitor against contention based protocols.

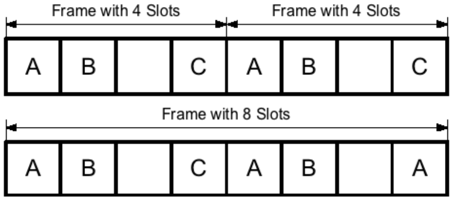

Figure 7. Using different Frame Sizes

Each node maintains a list of all its neighbors. Three entries define this list: The link quality, the unidirectionality status and the compressed neighborhood information from that neighbor. The link quality can be good (more than 90%reception rate), medium (between 30% and 90 % reception rate) or bad (less than 30% reception rate). The unidirectionality status can be either bidirectional, unidirectional_sender or unidirectional_receiver. The compressed neighborhood list is maintained by the neighborhood discovery protocol and used to identify the 2-hop-neighborhood of the current node.

The state machine of MLMAC-UL can be seen in figure 8. The arrows in the figure represent the transitions between states and are described in the following.

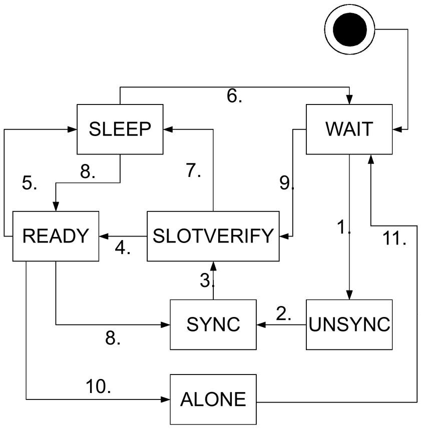

Figure 8. The State Machine of MLMAC-UL

1) When a node needs to acquire its first slot it switches into the state `UNSYNC`.
2) The node was in state `UNSYNC` for one frame. It chooses a slot and transitions into the `SYNC`-state. If no slot was empty, the node stays in its current state for another frame.
3) When its chosen slot arrives, the node changes to state `SLOTVERIFY`.
4) The node sends in its slots. After one frame, it reaches the `READY`- state.
5) If a negative acknowledgment for the last slot was received, the slot is deleted and the node changes to state `SLEEP`.
6) The node returns to the `WAIT`-state after a random amount of time.
7) Same as 5.
8) There is data to be transmitted and no neighboring node is transmitting. The node chooses a slot and an identification for the synchronization. After waiting for a random time it transmits the data and switches to `READY`.
9) If this node did not communicate before or it had previously given up one slot a new slot is acquired and the node changes into the state `SLOTVERIFY`.
10) No messages from neighbors were received for 5 frames even though this node is transmitting. This means that this node is either completely isolated, or has only unidirectional links to others, but no incoming link from any of them. This node switches to the `ALONE`-state and does not try to transmit anymore, even when data is available.
11) A message from a neighbor was received, which means that this node is no longer alone or a certain number of frames (e.g., 200) have passed. The node switches to `WAIT` and starts again.

V. ECTS-MAC

ECTS-MAC (Extended Clear To Send MAC) is a contention based protocol for sparse networks with rare communication. It is similar to BW_RES [17] (see Section III), because it also tries to forward the CTS message to reduce the probability of collision. Unlike BW_RES, it does not calculate distances and power levels. Also, all ECTS messages are sent at the same time, whereas all BW_RES messages are sent one after another. This leads to more collisions of ECTS messages, but saves a lot of time. When a node receives a CTS message it forwards it with a certain probability (figure 9). Experiments have shown that 50% seems to be the optimal value for sparse networks. If the probability is less, the ECTS message is not received by enough neighbors. If it is higher, the ECTS packets collide more often. These collisions are also the reason why the ECTS-MAC should only be used in sparse networks, as the ECTS packets would increase the network load in a dense network too much. To a certain extend, this effect is alleviated by reducing the probability of sending, but this also leads to more nodes that do not receive the ECTS message. The ECTS-MAC uses the neighborhood discovery protocol described in the previous section to detect unidirectional links. This is necessary to enable transmitting via a unidirectional link, because acknowledgments need to be forwarded to the sender using a second node.

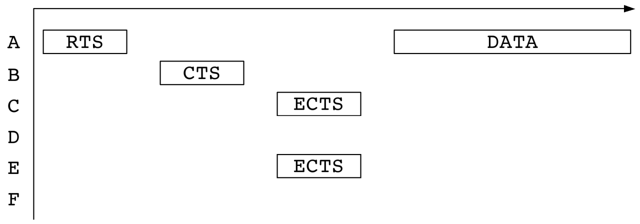

Figure 9. Propagation of ECTS Messages

VI. Evaluation

To measure the performance of MLMAC-UL and ECTS-MAC, we evaluated them against two other protocols: The original MLMAC and a modified version of MMP (Multicast Mac Protocol) [12] (see Section III). As its name suggests, MMP was designed for multicast, not for broadcast. We changed its behavior to enable broadcast transmissions, and to enable it to use unidirectional links. For this, we once again used the neighborhood discovery protocol described above. We call the resulting protocol NMAC (Neighborhood MAC). The functionality of NMAC is depicted in figure 10. Because of the neighborhood discovery protocol, the node knows how many neighbors it has and addresses them all in the RTS packet. When a node receives a RTS message it waits for a time corresponding to its position in the RTS before transmitting a CTS. If it has received at least one CTS, the sender of the RTS transmits the data package after the time for all CTS messages has passed.

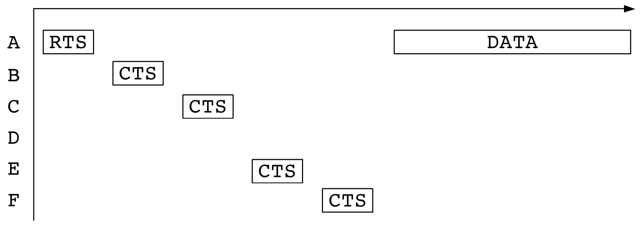

Figure 10. Message Propagation in NMAC

For our evaluations we used the discrete event simulator OMNeT++ [22] as well as real sensornet hardware. In all simulations the nodes transmitted with 19.2 KBit per second and a transmission strength of 10 milliwatt. For our real experiments we used TMote Sky sensor nodes from MoteIV corporation. These feature a MSP430 microcontroller with a frequency of 8MHz, 10 kB of Ram and 48 kB of flash. The radio module is IEEE 802.15.4 compatible and transmits 250kB/s, we configured the transmission strength to -25dBm to enable a multi hop scenario.

A. Single Hop Scenario Simulation

In this scenario the application behavior for a direct one-hop-neighborhood was simulated. The application tried to send as fast as possible. It generated a packet with 110 bytes data every 20 milliseconds, up to a total of 500 packets. In this simulation, all 4 protocols achieved a packet reception rate of nearly 100%. Figure 11 shows the amount of application data transmitted by each protocol. The figure shows that for 2-4 nodes the contention based protocols are able to transmit more data than the TDMA protocols. When more nodes are used, MLMAC-UL can gain an advantage because of the usage of multiple slots per node. As the number of slots was not changed for the MLMAC, it always delivers the same amount of data.

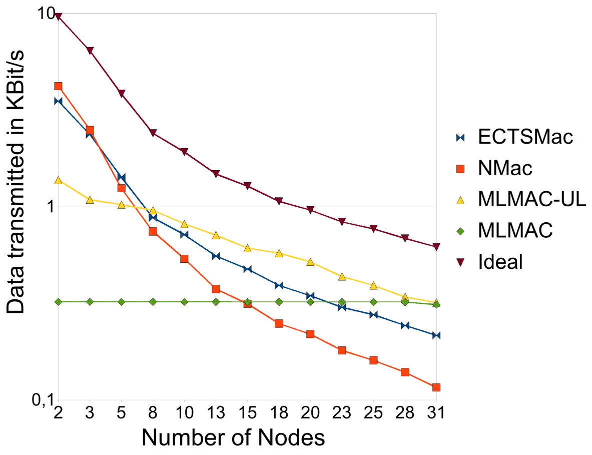

Figure 11. Data Transmitted 1-Hop Scenario

B. High Load Scenario Simulation

In this scenario a rectangle of 6 time 8 nodes was simulated. The application tried to send as fast as possible. It generated a packet with 110 bytes data every 20 milliseconds, up to a total of 500 packets. The node in the upper left corner started transmitting, each other node began transmitting its 500 packets after it had received the first packet. From a certain time on, all nodes want to transmit at nearly the same time, thus leading to a high network load. We evaluated the number of packets that were transmitted flawlessly against the number of nodes in the 2-hop-neighborhood.

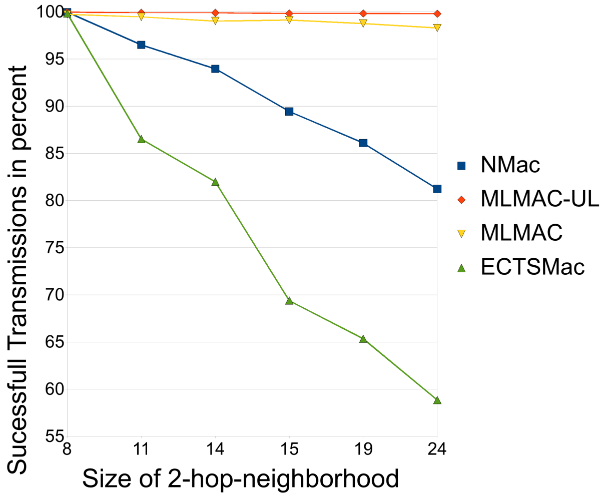

Figure 12. Delivery Ratio Rectangle Scenario

Figure 12 shows the percentage of successfully delivered packets for all 4 evaluated MAC protocols. As expected, the two TDMA based protocols were much better suited for this scenario then the contention based ones, which produced too many collisions.

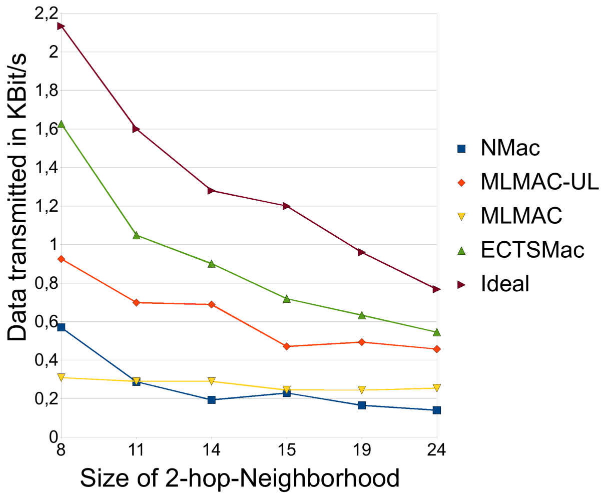

Figure 13. Data Transmission

Figure 13 shows the average amount of data each node was able to transmit for all 4 MAC protocols and the theoretical maximum. As can be seen, the ECTS-MAC is able to transmit most application data, followed by MLMAC-UL, MLMAC and NMAC. It is important to keep in mind here that this in the amount of application data transmitted, not received. If you correlate the bytes transmitted to the delivery ratio of the protocols, the performance of the ECTS-MAC drops considerably. The original MLMAC suffers from the fact that nodes may transmit only each frame, whereas MLMAC-UL allows each node to use multiple slots.

Another evaluation using the 6 times 8 nodes rectangle was used to determine the protocols' ability to deal with unidirectional links. To do this, we varied the rate of these from 0 to 70% in steps of 10. Figure 14 shows that MLMAC and MLMAC-UL can cope with the unidirectional links much better than ECTS-MAC and NMAC. Please note that these results were achieved using the neighborhood discovery protocol described above. If it is disabled, the performance of MLMAC-UL drops considerable, because it can no longer detect the unidirectional links and slots are given up too often. For the other protocols the impact is neglectable.

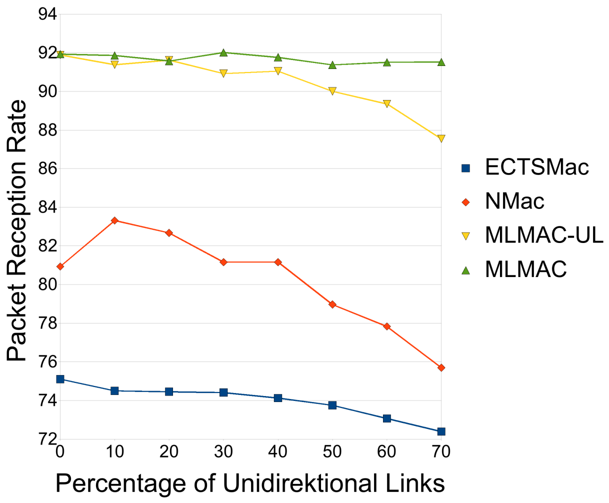

Figure 14. Variation of Unidirectional Links

C. *Mobility Simulation*

In this scenario the 4 protocols were evaluated using mobility with different speeds, random starting points and random destinations. The application sent packets of 110 Byte every second. This leads once again to a high network load. Figure 15 shows the number of received packets for each protocol for the different speeds. Once again, MLMAC and MLMAC-UL provide the best results, with ECTS-MAC performing only a little worse. The strong problems of NMAC are the result of a high rate of collisions. This is due to the fact that nodes which are leaving each others vicinity and thus produce a high number of transmission errors are seen as unidirectional links by both nodes and thus not addressed in the RTS message. They don't forward the CTS, which leads to another rise in collisions. The problem gets worse when nodes re-enter each others vicinity shortly after leaving it, because their links remain marked as unidirectional too long.

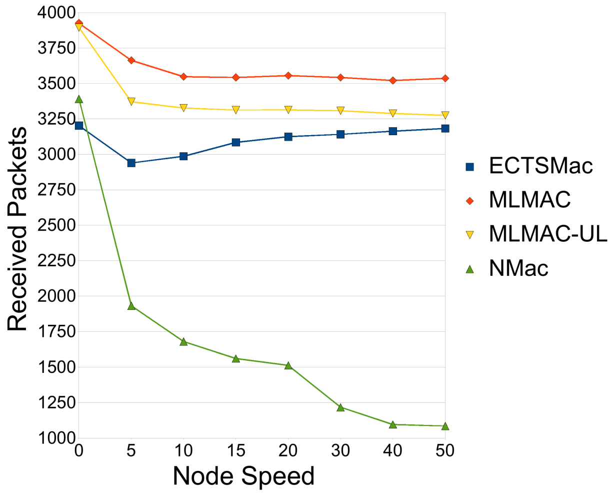

Figure 15. Received Packets

D. *Simulated Flooding over 50 hops*

In this set of simulations, the performance under low network load is evaluated. We simulated a line of 6 to 51 nodes, where each node was only able to communicate with its direct neighbors. Table II shows the time needed by each protocol to deliver a message over 50 hops. The times for the TDMA protocols are divided once using 5 slots and once using 31. Even though there were enough unused slots, the MLMAC-UL did not acquire new ones, because there was not much data to be sent and the send queue only ever held one packet. This leads to nearly the same time (one frame) needed as when using the original MLMAC, as the time for one hop only depended on the frame length. For all protocols, the time needed to reach the last node increased linearly with the number of nodes in use.

Table II
Time Needed for 50 Hops (ms)

NMAC	3,77
MLMAC-UL 31 slots	70,29
MLMAC 31 slots	69,25
ECTS-Mac	3,63
MLMAC-UL 5 slots	12,32
MLMAC 5 slots	10,79

E. *Packet Overhead*

This last evaluation in the simulator was based on the same topology as the high load scenario, but the size of the data generated by the application was varied between 20 and 110 Byte. It transmitted at random intervals between 0 and 5000 milliseconds. Figure 16 shows the relative overhead each protocol produced (protocol bytes/total bytes) for an increasing size of the 2-hop-neighborhood. The calculation includes the periodic messages from the TDMA based protocols and the RTS, CTS and ECTS messages from the contention based protocols. It can be seen that NMAC produces by far the highest overhead, followed by the ECTS-MAC. Thus, contrary to common belief, sending periodic status messages does not produce a high overhead.

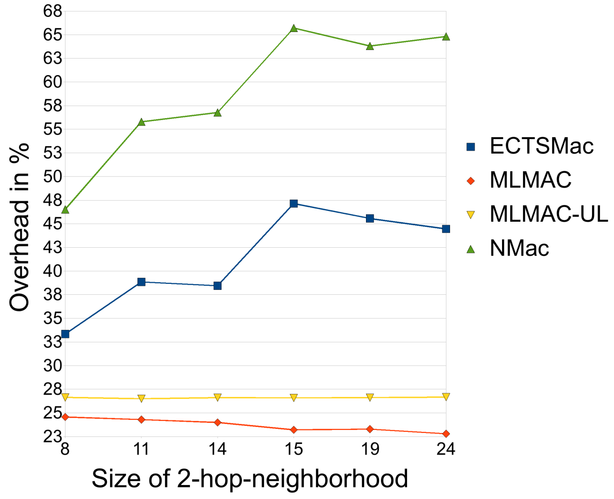

Figure 16. Protocol Overhead

F. *Direct Neighborhood Experiments*

In these experiments the application sent 500 packets of size 110 Byte every 10 Milliseconds. They were performed using 3, 7, 11 and 16 nodes. Figure 17 shows that for a small number of nodes all protocols perform relatively well. With an increasing number of nodes the performance of first NMAC and then MLMAC drop considerably. This is due to the increased number of CTS and ECTS messages, which lead to a high network load, a lot of collisions and thus a low throughput.

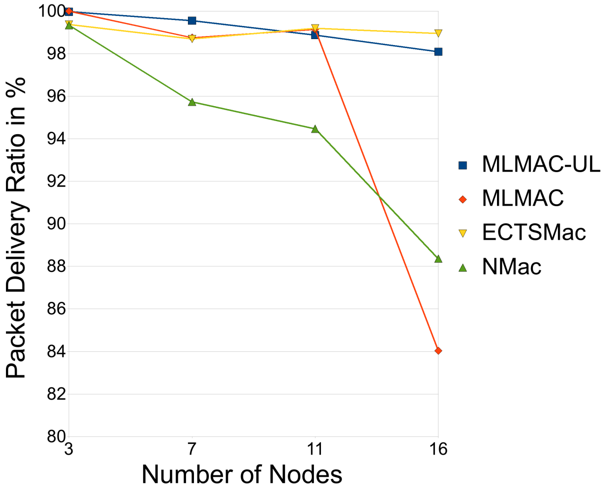

Figure 17. Packet Delivery Ratio Single Hop Experiments

G. High Load Scenario Experiments

For these experiments we placed 14 TMote Sky sensor nodes on the floor in a building. As there are no ways to define link quality in a real experiment we could only measure it. Figure 18 shows the resulting communication graph. It can be seen that the radio neighborhood of the nodes and the link quality differ a lot.

The application was once again the one producing the high network load. The left side of Figure 19 shows the average time needed to transmit one packet. MLMAC-UL and ECTS-MAC were the fastest ones, with MLMAC following and NMAC bringing up the rear. On the right side of the figure you can see the total number of received packets for each protocol. All protocols received nearly the same amount of messages, with only NMAC being considerably better

But this fact has to be put in perspective: all 4 protocols were evaluated one after another, using the same nodes and, most important, the same batteries. NMAC was the first protocol to be evaluated, which means that is had the advantage of fresh batteries which have been shown to have a positive effect on the range of the transceivers and thus link quality.

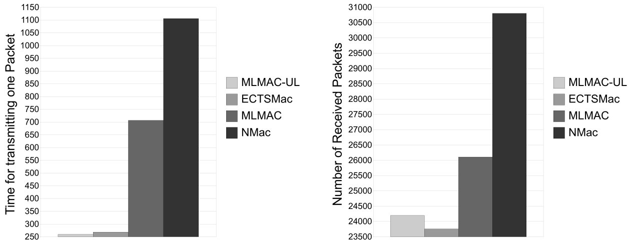

Figure 19. Time to Send(l) and Number of received Packets(r)

H. Memory Consumption

On Figure 20 the memory consumption of the protocols is shown, with 3 slots for the TDMA protocols on the left and 16 slots on the right. It is also differentiated whether the neighborhood discovery protocol was used or not, only the original MLMAC is shown only once, because it never uses that protocol. Please note that the numbers shown are for RAM consumption, the usage of flash memory follows the same distribution. On the figure it can be seen that MLMAC-UL needs most memory and ECTS has the lowest memory consumption. Combining this fact with the other results leads to the observation that for networks with low memory allowance and few nodes the ECTS-MAC should be chosen while the MLMAC-UL is best suited for denser networks with high load.

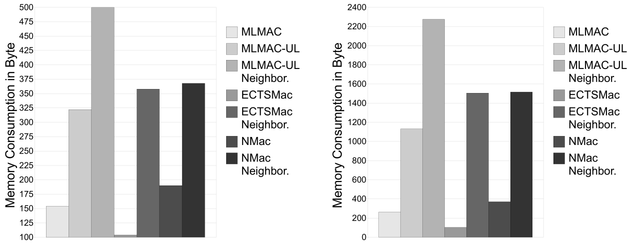

Figure 20. Memory Consumption for 3(l) and 16 Slots(r)

VII. Conclusion and Future Work

In this paper we have discussed the influence of unidirectional links on communication protocols. We presented two MAC protocols that can utilize unidirectional links to increase network connectivity, reliability and lifetime. MLMAC-UL is an enhancement to MLMAC, a TDMA based protocols for wireless sensor networks. ECTS-MAC is a contention based protocol that informs nodes that are connected to a sending node through unidirectional links about the impending communication by forwarding the CTS messages through multiple hops. Both protocols were evaluated by comparison with other protocols in simulations

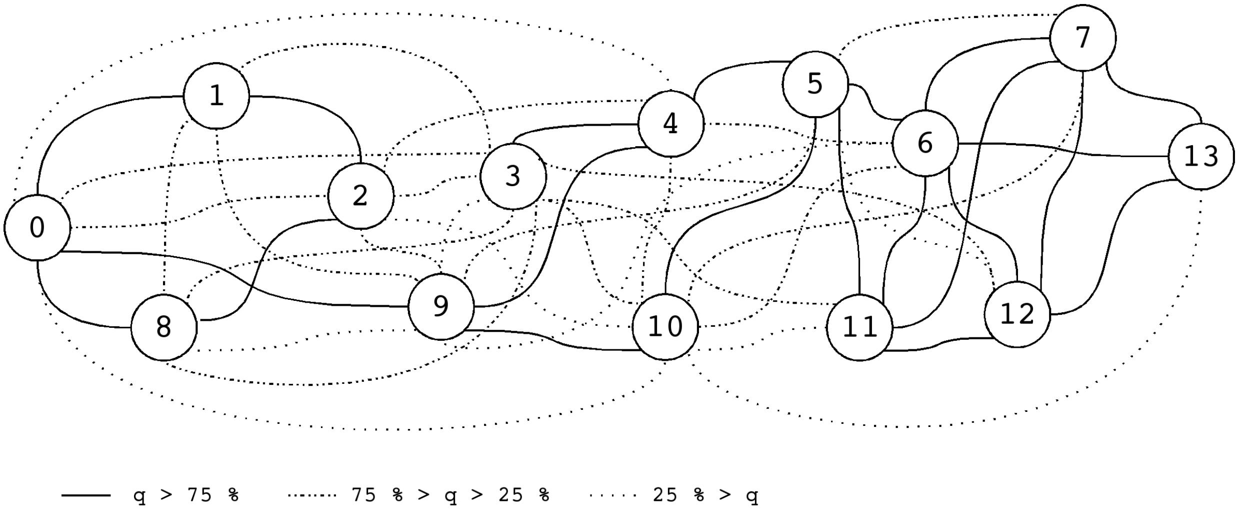

Figure 18. Link Graph where q denotes the Link Quality

and experiments with TMote Sky sensor node hardware. Both protocols show good results for different scenarios.

The choice of protocol depends strongly on the intended scenario and thus the application. If the network load is expected to be fairly low, ECTS-MAC is a good candidate. For high load scenarios however, MLMAC-UL performs far better due to the high number of messages generated by ECTS-MAC and the resulting collisions. Of course, node density and memory size are also important factors as The memory footprint of MLMAC-UL is considerably larger that that of ECTS-MAC. If the application needs lots of memory on a typical sensor node, MLMAC-UL simply might not fit in. In high density networks TDMA protocols normally suffer from a large frame size. If only a few nodes need to transmit data, they still have to wait for the slots of all other nodes to pass before transmitting again. In this case MLMAC-UL would prevail over ECTS-MAC only because of its adaptive nature, as nodes that need to transmit more data can acquire additional slots and release them once they are not needed anymore.

In the future we will continue our research on routing protocols that can make use of unidirectional links, building on the MAC protocols presented here. The possibilities of sharing information across layers, e.g. the data gathered by the neighborhood discovery protocol of MLMAC-UL, will also be explored.

References

[1] S. Mank, R. Karnapke, and J. Nolte, “Mlmac-ul and ects-mac - two mac protocols for wireless sensor networks with unidirectional links,” in *Third International Conference on Sensor Technologies and Applications*, (Athens, Greece), 2009.

[2] 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.

[3] J. Schiller, A. Liers, H. Ritter, R. Winter, and T. Voigt, “Scatterweb - low power sensor nodes and energy aware routing,” in *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005.

[4] V. Turau, “The heathland experiment: Results and experiences, presentation. www.tm.uka.de/forschung/spp1140/events/kolloquium/2005-11/turau.pdf.”

[5] 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), pp. 11–21, ACM Press, 2007.

[6] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 134–146, ACM, 2003.

[7] R. Min and A. Chandrakasan, “A framework for energy-scalable communication in high-density wireless networks,” in *ISLPED '02: Proceedings of the 2002 international symposium on Low power electronics and design*, (New York, NY, USA), pp. 36–41, ACM, 2002.

[8] T. L. Dinh, W. Hu, P. Sikka, P. Corke, L. Overs, and S. Brosnan, “Design and deployment of a remote robust sensor network: Experiences from an outdoor water quality monitoring network,” in *LCN '07: Proceedings of the 32nd IEEE Conference on Local Computer Networks*, (Washington, DC, USA), pp. 799–806, IEEE Computer Society, 2007.

[9] 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), pp. 14–27, ACM Press, 2003.

[10] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, and B. Krogh, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 1–38, 2006.

[11] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, 2003.

[12] H. Gossain, N. Nandiraju, K. Anand, and D. P. Agrawal, "Supporting mac layer multicast in ieee 802.11 based manets: Issues and solutions," in *LCN '04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, (Washington, DC, USA), pp. 172–179, IEEE Computer Society, 2004.

[13] R. M. Yadumurthy, A. C. H., M. Sadashivaiah, and R. Makanaboyina, "Reliable mac broadcast protocol in directional and omni-directional transmissions for vehicular ad hoc networks," in *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, (New York, NY, USA), pp. 10–19, ACM, 2005.

[14] S. Jain and S. R. Das, "Mac layer multicast in wireless multihop networks.," in *COMSWARE '06: Proceedings of the 1st International Conference on Communication System Software and Middleware*, 2006.

[15] G. Wang, D. Turgut, L. Bölöni, Y. Ji, and D. C. Marinescu, "A simulation study of a mac layer protocol for wireless networks with asymmetric links," in *IWCMC '06: Proceedings of the 2006 international conference on Wireless communications and mobile computing*, (New York, NY, USA), pp. 929–936, ACM, 2006.

[16] V. Ramasubramanian, R. Chandra, and D. Mosse, "Providing a bidirectional abstraction for unidirectional ad hoc networks," 2002.

[17] N. Poojary, S. V. Krishnamurthy, and S. Dao, "Medium access control in a network of ad hoc mobile nodes with heterogeneous power capabilities," in *in: IEEE International Conference on Communications (ICC), 2001*, pp. 872–877, 2001.

[18] L. Bao and J. J. Garcia-Luna-Aceves, "Channel access scheduling in ad hoc networks with unidirectional links," in *DIALM '01: Proceedings of the 5th international workshop on Discrete algorithms and methods for mobile computing and communications*, (New York, NY, USA), pp. 9–18, ACM, 2001.

[19] S. Mank, R. Karnapke, and J. Nolte, "An adaptive tdma based mac protocol for mobile wireless sensor networks, best paper award," in *International Conference on Sensor Technologies and Applications*, 2007.

[20] S. Mank, R. Karnapke, and J. Nolte, "Mlmac - an adaptive tdma mac protocol for mobile wireless sensor networks," in *Ad-Hoc & Sensor Wireless Networks: An International Journal, Vol.8 Nr.1-2*, 2009.

[21] L. van Hoesel and P. Havinga, "A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches," in *INSS, Japan*, Jun 2004.

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