IMPACT - A Family of Cross-Layer Transmission Protocols for Wireless Sensor Networks

Marcin Brzozowski
IHP Frankfurt/Oder
System Design group
brzozowski@ihp-microelectronics.com

Reinhardt Karnapke
BTU Cottbus
Distributed Systems/
Operating Systems group
karnapke@informatik.tu-cottbus.de

Jörg Nolte
BTU Cottbus
Distributed Systems/
Operating Systems group
jon@informatik.tu-cottbus.de

Abstract

For economic reasons sensor networks are often implemented with resource constrained micro-controllers and low-end radio transceivers. Consequently, communication is inherently unreliable and especially multi-hop communication suffers severely from packet losses. Transmission protocols that rely on implicit acknowledges for multi-hop communication are energy efficient but require symmetric communication links to work properly. In this paper we introduce IMPACT, *a family of transmission protocols that rely on implicit acknowledges and employ a cross layer approach to handle asymmetric links.*

1 Introduction

Sensor networks are collections of small sensor nodes with wireless neighborhood broadcast facilities. Since sensor networks shall be deployed in large scales (possibly thousands of nodes [2]), the overall costs dictate the use of cheap and simple radio transceivers for communication. These lack most of the common capabilities of WLAN or bluetooth networks. Even typical tasks like medium access control or the addressing of individual nodes in the direct radio neighborhood are left entirely to software layers [6]. Because of the small bandwidth and limited energy of sensor nodes it is essential to send as seldom as possible. Sending one byte, for instance, requires roughly as much energy as the computation of 1000 bytes [9].

This paper introduces a transmission protocol family called IMPACT (IMPlicit ACknowledgement Transmission protocol), which increases communication robustness while keeping the number of sent messages low. To conserve energy, IMPACT uses implicit acknowledges. This approach nearly halves the total number of sent messages in the best case, i.e. when communication is symmetric and the sender hears the propagation of its packet. In the worst case, when there are only asymmetric links, the total number of sent messages corresponds to the number in an approach with explicit acknowledges. In the presumed average case, when the source-destination path consists of symmetric and asymmetric links, sending of several acknowledgements can be avoided. Therefore, fewer packets are transmitted and energy is saved. Also, due to the cross-layer approach used, unidirectional links can be used that would otherwise have been discarded.

The remaining sections are structured as follows: In Section 2 we take a closer look at the strengths and weaknesses of link-layer, routing and transport protocols in the presence of unidirectional links. Section 3 describes our protocol while section 4 discusses our experiments and their results. Related work is shown in section 5. We finish with conclusion and future work in section 6.

2 Protocol Layers in the Presence of Unidirectional Links

In this section we take a closer look at the way unidirectional links are treated in link-layer, routing and transport protocols. Medium Access Control (MAC) in the link-layer suffers heavily from unidirectional links. Most protocols as-

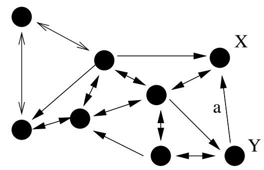

Figure 1. A mixed network. Communication is only possible in the direction indicated by the arrows.

sume that communication is symmetric i.e. that when node A hears node B node B can hear node A, too. This is reflected in the protocols by the usage of flow control which uses RTS/CTS signals or in the assumption that the medium is free when a node does not hear anything (CSMA). In the case of link *a* in figure 1 both approaches would fail because node *Y* can not hear node *X*. Other protocols which use a timed schedule for sending (TDMA) suffer from asymmetric links, too. A local TDMA slot needs to be defined in order to avoid collisions. A node that is downstream of an unidirectional link may reserve an unnecessary slot for the upstream node while the upstream node does not realize that it can disturb the communication of the downstream node. If no additional action is taken these nodes can end up using the same slot. If the traffic load is high, these nodes will often produce collisions.

Even if the MAC protocol can work with unidirectional links, very similar problems arise at the routing layer. Most routing protocols try to ignore/remove unidirectional links if they are considered at all. Normally a route request packet is flooded into the net and collects the identities of the nodes it passes. When it reaches the destination a route reply is sent back along the cumulated path to the source. If the route includes only one asymmetric link this approach fails. The route reply cannot be sent back and, as additional route requests are normally ignored, no route can be established. AODV [8] for example offers the usage of periodic ”Hello messages” and/or a ”Blacklist” to eliminate unidirectional links and ignore malicious nodes. DSR [5] can be used on unidirectional links by introducing a new packet which works as a route request but is sent from the destination. This way, unidirectional links are not ignored as in AODV but rather they are used in the route discovery. The cost for this usage is high, though. The second route request is flooded like the first but it is much larger as it contains the route from the source to the destination and accumulates the reverse path. This leads to a communication cost which is many times as high as it would be if only bidirectional links were present.

Only if both MAC and routing protocols are able to handle unidirectional links, transport layers may work. The information whether a route uses unidirectional links is hidden from the transport layer. As the transport layer uses a routing layer, it relies on the lower protocols to take care of these problems. If a packet gets lost, the transport layer can therefore not decide what the reason for this loss is. It assumes that it has to retransmit the packet. There are some protocols, like the well known TCP, that rely on end-to-end acknowledgements alone. In these protocols the error correction is done between the two ends of the communication, the source and the destination. One of the major advantages of these protocols is that the nodes in between can be quite dumb. All they have to do is to remember a route and forward the packets. Furthermore, if at least the routing layer supports unidirectional links, the transport layer can use them too.

But pure end-to-end error recovery has many disadvantages, particularly in networks where the error rate is high. Although a packet could have been lost just before arriving at the destination, its duplicate needs to be forwarded from the beginning. All nodes that have already flawlessly received and sent the packet need to forward its duplicate again. Thus, energy is unnecessarily consumed. Obviously, the probability of packet loss grows with the source-destination distance, making pure end-to-end approaches insufficient in sensor networks. A hop-by-hop error recovery mechanism has to be used, because it increases robustness significantly. Additionally, a lost packet is repeated almost immediately, since waiting time is calculated for communication with the next node. Due to the short timeouts, the message buffer can be recycled earlier. This leads to a smaller number of buffers needed in each node. In contrast to the end-to-end approach, copies of a lost packet are not repeated by the source, but by the node that detected the loss. Therefore, copies are not forwarded through nodes, which have already received and sent the packets correctly. Consequently, node energy is conserved.

In contrast to the end-to-end approach, where nodes forward packets and forget about them, in the hop-by-hop case all nodes need to use a timer for every message, remember the transmision time and buffer not acknowledged packets. This way, part of the communication cost is traded against computation and data size. Hop-by-hop recovery however cannot be realized in the transport layer above the routing. It has to be taken care of in the link layer. Also, support from the routing layer is required to deal with unidirectional links.

3 The IMPACT Protocol

The IMPACT (IMPlicit ACknowledgement Transmission) protocol consists of pure IMPACT and its extensions IMPACT-CLA (Cross-Layer-Acknowledges) and IMPACT-DR (Dynamic Rerouting). The three parts of IMPACT can be combined in any desired form, therefore they are implemented in the COPRA-Framework [7] which allows free configuration of any of its parts. In this section the three parts of IMPACT are discussed in detail.

3.1 Pure IMPACT

In wireless networks the transmission of a message is always a broadcast. Consequently all neighbors of a transmitting node receive the message. While in other protocols normally all but one of them ignore this message, IMPACT takes advantage of the broadcast property. In our protocol we decided against sending explicit ACKs. Instead, a node that transmitted a message listens to all communication taking place in its surroundings. If the next node forwards the message, it is clear that the message was received, even though no ACK was sent. Please note that it is sufficient to overhear the unique ID of the message, the rest can be ignored. This way, listening for an implicit ACK does not consume more energy than waiting for an explicit acknowledgement on the sending node and the forwarding node saves the energy for the explicit ACK.

While this protocol theoretically saves a lot of transmissions, in practice it suffers heavily from unidirectional links. If the used link is not bidirectional the sending node does not hear the forwarding of its message. The problems arising thereof are discussed in section 3.2.

There are two cases where IMPACT needs to send explicit ACKs, even in pure bidirectional scenarios. First, the destination has to send an explicit acknowledgement after receiving a packet. Since it does not forward the packet, the last sender does not receive an implicit acknowledgement. Second, a node needs to send an explicit acknowledgment, when it receives a duplicate.

In wired networks duplicate suppression is often implemented end-to-end only, as it is only needed for consistency of data. In our protocol it is implemented on each node because transmission of duplicates along a path is a waste of energy. Thus, after recognizing a received packet as a duplicate, a node does not forward it. If no explicit acknowledgment was sent, the last sender would not be aware that its packet reached the next node and try to retransmit despite the fact that the packet was already forwarded.

3.2 Handling of Asymmetric Links

If a link changes from bidirectional to unidirectional, the sender does not hear the implicit ACK. Thus it would send the packet again, and finally give up and inform the routing layer that it has to try to find a different route. This is realized in IMPACT-DR and is a prime example for cross-layer issues as it demonstrates the need for cooperation between data-link- and routing layer. This way, IMPACT-DR helps to optimize communication paths and chooses those that are mostly symmetric.

The use of implicit acknowledgements leads to an energy-aware communication, but requires symmetric communication. As it cannot always be guaranteed that a symmetric path exists [13], we additionally adapted our approach to work with asymmetric links, too. The approach is based on the fact that explicit acknowledgements are passed through the routing layer before they are sent (IMPACT-CLA). Thus, it is possible to send an explicit acknowledgement to a node that cannot be reached directly, by forwarding a packet through several hops. To do this, it is necessary to know the identity (ID) of the last hop. As this ID normally is not known by the data-link layer, the cross-layer issue increases.

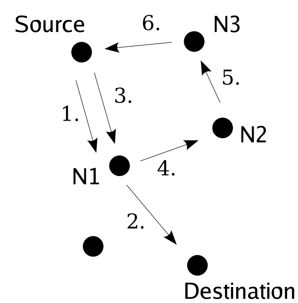

Figure 2. Handling of asymmetric communication in IMPACT-CLA

Figure 2 illustrates our way of dealing with asymmetric links. The arrows represent sent messages and their order, communication is always asymmetric. The source sends a packet to the destination by forwarding it to the node N1 (1.). After receiving the packet, the node forwards it to the destination (2.). Since the communication between the source and the node N1 is asymmetric, the source does not receive an implicit acknowledgment and, after a timeout, sends the packet again (3.). The node N1 receives the packet and recognizes it as a duplicate. An explicit acknowledgment needs to be sent, and is passed to the routing layer. The routing layer discovers that the acknowledgment cannot be sent directly to the source because of the asymmetric link. If the routing layer does not know a way to the source, it tries to find one. At last, the acknowledgement is forwarded to the node N2 (4.) and travels to the source through N3 (5. and 6.). Please note that for simplicity reasons we neglected the explicit ACK from the destination to N1 in this example.

We assume that in general the source to destination way consists of several symmetric and asymmetric links. If there is always a way back to the last sender, the approach works satisfactorily. The efficiency of IMPACT-CLA can be tuned, e.g. by limiting the number of hops for route discovery when sending explicit ACKs, since locality can be assumed.

4 Experiments

All our experiments were conducted either on modified RCX robots or in SERNet, our simulator. The Lego RCX robots feature a Renesas H8/300 micro-controller. This is a 16 Bit processor with a frequency of 16 MHz. These robots have been additionally equipped with a radio module of type ER400TRS [3] which we use instead of the included infrared module. SERNet is a Simulator/Emulator for wireless sensor networks which was developed at our chair. One of its major advantages over simulators like ns2 which have better models of the real world is that it enables the simulation of interrupts. This allows the usage of the same code in the simulator and on the sensor nodes. Therefore, no re-implementation is necessary as would be the case with ns2. We use SERNet mostly as a development tool in which we debug the final code. When debugging in SERNet is done, the code can be transferred directly to the real nodes.

The main difference between end-to-end error recovery and pure IMPACT is supposed to be seen in a multi-hop communication. To examine such a communication, it suffices to use a line as network topology. At one end of the line is the source and at the other the destination.

4.1 Experiment description

In each experiment cycle many packets were sent by the application layer on the source node. The application layer on the destination counted how many packets arrived. By dividing the number of received packets by sent packets we got the success rate, which is called reliability in this paper. Additionally, we measured how many packets were sent by each node.

In the line experiments on SERNet and in the RCX network we performed similar experiments. The application layer on a source node sent data to the destination periodically. When using the simulator 50 packets were sent in each simulation, in the RCX experiments 30 packets were sent. First, we carried out a series of experiments using only an end-to-end error recovery mechanism. On the simulator we ran simulations with the end-to-end approach with a different number of retries (from 3 retries to no retries at all). Each end-to-end variant was run with various distances between source and destination (on the simulator from 2 to 14 hops; in the sensor network from 2 to 5 hops). In the sensor network we experimented only with one type of end-to-end approach, which was the one with 3 retries. To achieve higher accuracy, experiments were repeated 10 times on the simulator and 3 times in the RCX network. Next, all experiments were repeated with the same settings, but using IMPACT. This sums up to a total of 910 simulations and 24 real experiments for the line experiment.

In this experiment we wanted to confirm that end-to-end protocols were insufficient for sensor networks. We also wanted to show that this insufficiency does not only show when a large number of nodes is used. Therefore, we used only 6 robots with a maximal source-destination distance of 5 hops. This 5 hop distance met our needs and we confirmed end-to-end insufficiency for wireless sensor networks, details about which are given in the next section. To achieve a distance of 5 hops, we arranged the nodes in a row. To find proper distances between two nodes, we carried out simple experiments. A node was sending echo request packets to its previous neighbor. Simultaneously, the node was being moved farther away. The movement proceeded as long as echo reply packets were arriving. The node, then, was placed where the communication was still possible. Finally, we put all nodes on the corridor in such a way, that they could communicate only with the appointed neighbors.

We assumed distances between all of the nodes would be almost identical, as the transmission power of all nodes was set to the minimum. The distances were not equal though, they differed by far (from 1,4 meters to more than 6 meters). For this reason we think that it is mandatory to experiment in a real network, and not only in a simulator. Although each node was able to receive data only from its neighbors, other nodes could still affect communication. This is because the range of interference is much larger than the range of communication.

4.2 Experimental Results

In this section the results of the simulation and the real experiment with the line topology are discussed. First, reliability of both approaches, IMPACT and end-to-end error recovery, is analyzed. As stated before, when we speak of reliability, a success rate is meant. Next, we examine the amount of packets sent in both approaches. As mentioned earlier, sensor node energy is limited and has to be used accordingly. In comparison to other node activities, sending and receiving of data consumes an enormous amount of energy. For this reason, analysis of the sent packet count is essential and is discussed in detail.

In general, hop-by-hop error recovery is assumed to be more appropriate for wireless sensor networks than end-to-end. The former should deal better with packet losses, which are common in sensor networks. In other words, IMPACT is expected to show greater robustness than the end-to-end approach.

We show simulation results on the left side of figure 3, namely reliability of end-to-end approaches on the simulator. When measuring with no retries only end-to-end acknowledgements were used. We set the error rate to

20%. Using this loss rate, communication on small distances (only a few hops) with end-to-end recovery is already quite lossy (e.g. by the source-destination distance of 4 hops little more than 80% of data reached the destination when using 2 or 3 retries). For greater distances (13 hops and more), end-to-end recovery was completely insufficient. In that case, despite the fact that a lost packet could be repeated three times, less than 20% of data arrived at the destination. For distances longer than 10 hops the achieved reliability

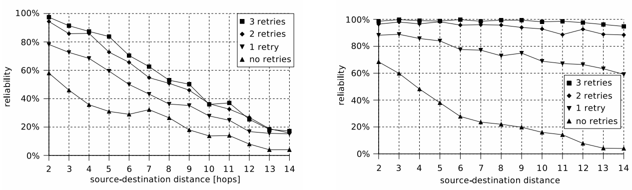

Figure 3. Simulator: a) end-to-end reliability b) IMPACT reliability

of all tried end-to-end variants differed only slightly. This observation made us expect that from a certain distance, in a network with error prone communication, increasing the number of retries does not significantly strengthen the robustness of an end-to-end approach.

The right side of figure 3 shows the reliability IMPACT achieved in simulations and compares them to the result-set using no retries. In contrast to the end-to-end results, the differences between IMPACT simulations when varying the number of retries were quite noticeable. By the distance of 14 hops the IMPACT version using 3 retries achieved almost 95% reliability. IMPACT with 1 retry by the same distance, however, achieved only 60% success rate.

The big difference between the two approaches, IMPACT and end-to-end error recovery, is illustrated on the left side of figure 4. End-to-end reliability depended tremendously on the hop distance between source and destination, whereas IMPACT's reliability decreased only slightly with increasing distance. Obviously, IMPACT can achieve sufficient robustness in wireless sensor networks, even for great distances and it outperforms end-to-end error recovery by far.

We also performed experiments with a real network. As it is much more time consuming to work with a real sensor network than with a simulator, less IMPACT and end-to-end variants were examined. We evaluated only approaches with three retries. Experiments up to a distance of five hops were carried out. The right side of figure 4 shows the achieved reliability of both error recovery mechanisms. The difference between them is enormous, especially at a distance of 5 hops. In that case, the end-to-end approach was nearly unable to deliver any data to the destination. Al-

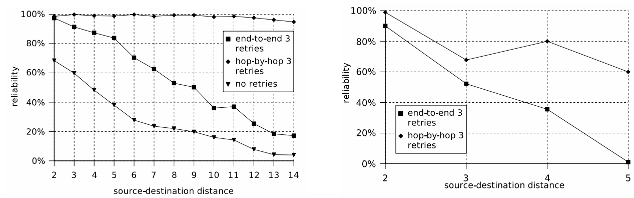

Figure 4. reliability, end-to-end vs. IMPACT: a) simulator b) RCX

though each lost packet was retransmitted three times, only approximately 1% of them reached the destination. In contrast, IMPACT achieved 60% reliability. Obviously, using end-to-end means could not satisfy robustness needs in this scenario. In contrast, IMPACT was able to deal with so frequent packet losses.

Realizing the importance of energy savings, we measured the total number of sent packets in the end-to-end and IMPACT simulations. After collecting this data we were able to analyze the achieved energy efficiency of both approaches. The left side of figure 5 illustrates the total num-

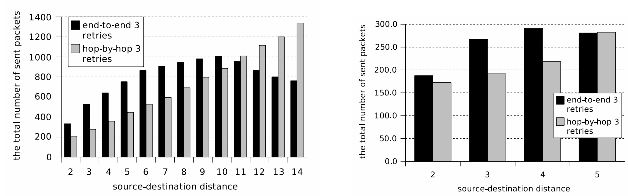

Figure 5. Total packets sent: a) simulator b) RCX

ber of packets sent using IMPACT and the end-to-end approach in the simulations. In both cases three retries were sent at most. The figure shows results for all measured distances (from 2 to 14 hops). The total amount of sent packets using IMPACT was smaller for distances up to 10 hops. As shown on the figure, the total number of sent packets using IMPACT increased with the distance. With a greater distance more nodes were involved in sending data, therefore the total number of sent packets was higher. In end-to-end error recovery, however, the total amount of sent packets grew from the beginning (the distance of 2 hops) until it reached its maximum at a distance of 10 hops. Then, the total number of sent packets shrank with the distance. Two factors affected the total number of sent packets in the end-to-end approach. First, with growing distance, the number of sending nodes increased. Hence, the total number of sent packets increased with the distance, similarly to the simulation of the IMPACT-protocol. Second, end-to-end acknowledgements were sent. With a growing distance fewer data packets reached the last nodes or the destination. Therefore, fewer end-to-end acknowledgements were sent which leads to the lower number of packets.

In simulations with great distances IMPACT sent more packets than end-to-end (for 14-hop distance: IMPACT sent around 1340 packets and end-to-end 760). Costs of sending more data using IMPACT were worthy, though. IMPACT outperforms end-to-end in achieved reliability (95% to 17% respectively), as illustrated on the left side of figure 4. We observed similar results in the sensor network experiments. The right side of figure 5 shows the total number of sent packets for various distances. When using IMPACT, their number grew with the distance, whereas in end-to-end the total amount achieved its maximum at the 4 hop distance and then shrank. For the 5 hop distance the number was almost the same for both approaches. The reliability using IMPACT was around 60%, whereas in the end-to-end approach almost no data reached the destination (around 1%), as shown on the right side of figure 4. Clearly, the only way to achieve robustness in this sensor network is to use the IMPACT-protocol or a similar hop-by-hop error correction. End-to-end is undoubtedly insufficient. Cross-layer acknowledges increase the usability further.

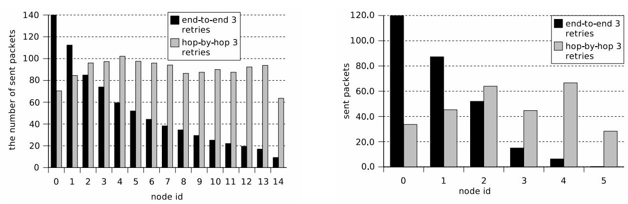

Figure 6. sent packet per node, end-to-end vs. IMPACT**: a) simulator b) RCX**

Figure 6 shows the number of packets sent by each node. It illustrates results from the simulator and from the sensor network, with the distances of 14 hops for the former and 5 hops for the latter. The left side of the diagrams represent the source (node 0), the right sides the destination (node 14 on the simulator and node 5 on the sensor network).

In the end-to-end approach the number of packets sent per node shrank with the distance from the source. This can easily be explained by the decrease in probability of reception. When using IMPACT the number of packets sent per node changed only slightly, due to the hop by hop operation which causes a higher probability of getting a packet through the net. In the end-to-end approach with more than 10 hops the nodes near the source would have unnecessary wasted their energy, as nearly no packet arrived at the destination. If the application used was a sense-and-send application, as most are nowadays [11], this would lead to a part of the net being drained and the other part being useless. IMPACT consumes roughly the the same amount of energy at each node, which leads to a longer lifetime of the net.

5 Related Work

The authors of Dynamic Source Routing [5] propose a routing protocol, that uses the broadcast character of wireless networks in routing maintenance. A node is responsible for a packet until it has reached the next hop, which can be noticed if an explicit ACK is received or through a passive acknowledgement, which is the same as our implicit acknowledgement.

The RMST (Reliable Multi-Segment Transport) protocol is introduced in [10]. RMST is a NACK-based protocol, which provides guaranteed delivery and fragmentation/reassembly for applications that require them. The authors analyze the reliability of implementations on different layers, which includes a comparison of hop-by-hop and end-to-end approaches. They come to the conclusion that the best implementation of a reliability protocol involves both the transport- and MAC layer. They emphasize however, that MAC layer reliability should not be used when sending route discovery packets. Our approach incorporates the routing layer, as we think that it is necessary for handling of asymmetric links.

In [12], the authors introduce PSFQ (Pump Slowly, Fetch Quickly), a scalable transport protocol. The authors propose a hop-by-hop error recovery model, where each node is responsible for loss detection and recovery. Opposing to ours, which is based on ACKs, this protocol is based on NACK messages. If a node receives an out-of-order packet, it sends a NACK message to its immediate neighbors and requests the missing fragment. The authors compare hop-by-hop and end-to-end approaches and suggest that end-to-end recovery is not a good candidate for a reliable transport layer in wireless sensor networks.

Another hop-by-hop approach is presented in [1]. The authors introduce a reliability layer for a Multi path Ondemand Routing protocol, which collaborates with the routing layer and tries to find another path to the destination, when the current one does not work anymore. The similarity to IMPACT lies in the combination of reliability and routing but IMPACT is independent of the routing protocol used.

In [4], the authors analyze the problem of different levels of information importance in sensor networks. According to the importance of the information contained, packets should be sent with desired reliability. That is, more energy can be spent for sending critical data and less for not so important packets. The authors analyze several hop-by-hop approaches, particularly from the perspective of the total packet overhead needed to achieve the desired reliability. This approach can easily be integrated into our protocol, by introducing a priority which could lead to a higher number of retries in the retransmission or to earlier notification of the routing layer.

6 Conclusion and Future Work

We have presented a cross layer retransmission protocol family called IMPACT, which increases communication robustness in wireless sensor networks significantly while conserving energy on paths that are at least partly symmetric. Because the presented approach uses implicit acknowledgements, almost no additional messages are needed to acknowledge received data. The approach with implicit acknowledgements involves several cross-layer issues which have an impact on design and implementation complexity. IMPACTs solution with cross-layer acknowledges and dynamic rerouting enables the usage or removal of unidirectional links, and leads to an energy-aware communication.

We ran our implementation on a simulator and in a real sensor network. After performing experiments in the sensor network, we realized how much results differed from simulator experiences. As far as we know, no simulation could produce the results we obtained in the field experiment. Thus, we regard experimenting in a real sensor networks as particularly important. Our experiments have shown that not only in huge networks consisting of thousands of nodes, but also in small ones with a source-destination distance of a few hops, link layer retransmissions are crucial.

Although the implemented approach (IMPACT, IMPACT-CLA and IMPACT-DR) works quite well, we plan to carry out more research concerning performance issues. We also consider adding congestion control to our communication model. The broadcast character of communication in wireless networks can help in implementing this.

References

[1] E. Biagoni and S. H. Chen. A Reliability Layer for Ad-Hoc Wireless Sensor Network Routing. In *Proceedings of the 37th Hawaii International Conference on System Sciences - 2004*, 2004.

[2] I. Chatzigiannakis, S. Nikoletseas, and P. G. Spirakis. Efficient and robust protocols for local detection and propagation in smart dust networks. *Mob. Netw. Appl.*, 10(1-2):133–149, 2005.

[3] circuit design inc. low power radio solutions. www.lprs.co.uk/main/product.info.php?productid=154.

[4] B. Deb, S. Bhatnagar, and B. Nath. Information assurance in sensor networks. In *WSNA '03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 160–168, New York, NY, USA, 2003. ACM Press.

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

[6] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for "smart dust". In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 271–278, New York, NY, USA, 1999. ACM Press.

[7] R. Karnapke and J. Nolte. Copra - a communication processing architecture for wireless sensor networks. In *Euro-Par 2006 Parallel Processing*, pages 951–960. Springer, 2006.

[8] C. E. Perkins and E. M. Royer. "ad hoc on-demand distance vector routing.". In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA*, pages 90–100, FEB 1999.

[9] 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.

[10] F. Stann and J. Heidemann. RMST: reliable data transport in sensor networks. In *Proceedings of the First International Workshop on Sensor Net Protocols and Applications*, pages 102–112. IEEE, 2003.

[11] Turau, Renner, and Venzke. The heathland experiment: Results and experiences.

[12] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: a reliable transport protocol for wireless sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 1–11, New York, NY, USA, 2002. ACM Press.

[13] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27, New York, NY, USA, 2003. ACM Press.