Reliable Multicast in Wireless Sensor Networks*

Gerald Wagenknecht, Markus Anwander, Marc Brogle, and Torsten Braun
Institute of Computer Science and Applied Mathematics, University of Bern
Neubrueckstrasse 10, 3012 Bern, Switzerland
{wagen|anwander|brogle|braun}@iam.unibe.ch

ABSTRACT

Multicasting in Wireless Sensor Networks (WSNs) is an efficient way to disseminate the same data to multiple receivers. For critical tasks such as code updates, reliability would be a desirable feature, in order to use multicasting for such scenarios. Due to the nature of WSNs, several problems exist that make realizing an efficient, reliable and energy consumption friendly implementation a challenging task. In this paper we describe the challenges of such an implementation and propose a solution for designing a reliable multicast solution based on IP Multicast and Overlay Multicast. We discuss several scenarios and depict the different advantages and limitations of the solutions proposed.

1. INTRODUCTION

1.1 Wireless Sensor Networks

A Wireless Sensor Network (WSN) consists of a number of sensor nodes, which are limited in terms of energy, CPU power, and memory. On the sensor nodes may run different applications for different tasks such as event detection, localization, tracking, and monitoring. Such applications should be configured and updated during the life-time of the sensor nodes and over the network [1]. An update with many unicast connections to the nodes is very inefficient and consumes resources such as bandwidth and energy. Thus it is obvious that multicast communication the management of WSNs may benefit by reducing the number of transmitted packets and by saving energy. To access WSNs via the Internet, a IP-based communication is required [2]. Thus multicast communication should be IP-based as well.

1.2 Multicast

Multicast is an efficient way to disseminate data to a group of receivers that are interested in the same content. Contrary to unicast, where the sender has to transmit the data for each receiver individually, multicast requires the sender to transmit the data only once. Thereafter, the network or other hosts interested in the data will replicate when required and forward the data to the receiving group members. In the Internet, the multicast paradigm has been implement in the form of IP Multicast. Interested receivers send an IGMP group join message, the routers process these messages according the IP Multicast protocol used (RSVP, PIM, etc.) and build the distribution tree among them. A sender now only sends a UDP Multicast packet to the group's address and the routers in the network then distribute the data according to the multicast tree that has been setup before. Although IP Multicast has been available for a while, it has not been widely deployed in the Internet today due to different reasons (configuration, ISP agreements, etc). To offer multicast functionality to the end-user, the concept of Application Level Multicast[3] (ALM), often also called Overlay Multicast, has been introduced. With ALM, which is based on the Peer-to-Peer[4] (P2P) paradigm, end-systems build the multicast tree among themselves, rather than relying on routers to handle multicasting on their behalf.

Numerous research has been done about multicast in WSNs. In [5] a multicast protocol called BAM (Branch Aggregation Multicast) is presented, which supports single hop link layer multicast and multi-hop multicast via branch aggregation. VLM^2 (Very Lightweight Mobile Multicast) [6] is a multicast routing protocol for sensor nodes, which is implemented on-top of the MAC protocol. It provides multicast from a base station to any sensor node, unicast connections from a sensor node to the base station, and supports mobility. In [7] the authors present an effective all-in-one solution for unicasting, anycasting and multicasting in wireless sensor networks and wireless mesh networks. The authors of [8] adapt ADMR (Adaptive Demand-driven Multicast Routing), a multicast protocol for MANETS, on a real wireless sensor node (MICAz). They show that the adaption is not a trivial task and a number of problems have to be solved. The authors of [9] analyze IP Multicast and show that it is possible to use it in WSNs. Further there are several multicast solutions for WSNs which are based on the geographical position of the sensor nodes in the network [10, 11].

1.3 Structure of the Paper

The remainder of the paper is structured as follows. In the next Section multicasting in WSNs is described showing the challenges, different designs of multicast, and a protocol stack for IP-based communication is proposed. Section 3

*This work has been supported by the Hasler Foundation under grant number ManCom 2060

discusses the advantages and limitations of our proposed solutions. An outlook in Section 4 closes this paper.

2. MULTICAST IN WIRELESS SENSOR NETWORKS

2.1 Challenges

Due to the nature of WSNs, the IP Multicast implementation can not be simply ported from existing solutions for wired networks. In WSNs energy, memory and CPU power are limited. This implies the following challenges for multicast in such environments. In wired networks, routers are handling packet replication and forwarding, clients just send and receive simple IP UDP datagrams. On the other hand, WSNs would need to introduce the router functionality for IP Multicast management into each sensor node. Group management is normally concentrated on the routers that communicate with each other to handle multicast trees. The management for multiple groups and multicast trees requires memory and processing power, which is limited on sensor nodes. Also the default implementations of IP Multicast are designed to scale on large network groups with multiple receivers and senders. In practical WSNs typically the amount of nodes is rather low. Also the amount of active trees and general management communication should be kept to a minimum. Existing Overlay Multicast [3] solutions (such as Scribe/Pastry, CHORD, Bayeux) are normally not taking the wireless nature and limited capabilities of sensor nodes into account. In general it is also a bad idea to have overlay connections established all the time, which would lead to higher energy consumption and therefore reduces the lifetime of WSNs. Several other issues concerning liveliness, wireless communication and collisions exist. Also reliability for a WSN multicast solution would also be desirable, because code updates and other critical tasks could then be solved efficiently using multicast.

2.2 Designing Multicast in WSNs

Multicasting in WSNs can be designed in different ways. We will look at two approaches, reliable IP Multicast and Overlay Multicast. For both approaches we will look at source-driven and receiver-driven designs, both centrally managed as well as de-centrally organized. Generally we will distinguish between two node types. Branching nodes have to duplicate packets and store state information about receivers and/or about other branching nodes. Forwarding nodes have less or no information about the multicast state and just forward the multicast data to one neighbor. We will also limit our discussion to core-based trees, where only the dedicated root node will disseminate the data, while other senders would need to transmit the data to the root node first for dissemination. An example topology with some branching nodes, forwarding nodes and three group members is shown in Figure 1.

2.2.1 *Overlay Multicast*

For the source driven scenario we can use a de-centralized as well as a centralized approach. Generally we distinguish between active and inactive multicast trees. While data is transmitted to a multicast group, the tree is in the active state with all required TCP connections for the overlay links established. New nodes are not allowed to join the group

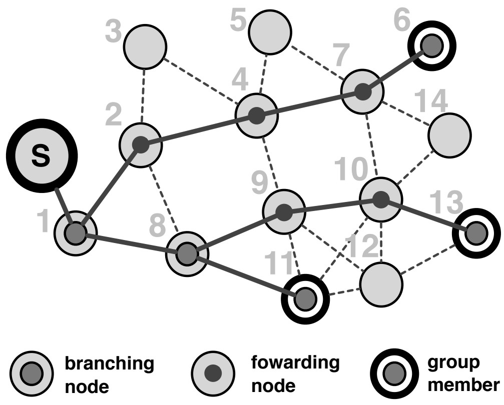

Figure 1: Example topology showing branching nodes, forwarding nodes and group members

while its tree is active, though the joins are cached and processed later. When no data needs to be transmitted, the tree is inactive and all required TCP connections to build the distribution overlay are closed. Nodes can only join to a multicast group while its tree is inactive. This limitation ensures, that subscribed members get all data for a dissemination session, because late joins are avoided.

In the de-centralized and source-driven approach, the source sends the list of all receivers of the multicast data to its one-hop neighbors. The neighbor nodes then check if all receivers in the list can be reached through a single of their next-hop neighbors. In this case, such a node just forwards the list to its next hop and remembers that it acts as a forwarder for that multicast group. If nodes from the list can be reached via different neighbors, the list is split accordingly and the partial lists (with the node's own address as source of the message) are forwarded to the respective neighbors. The node splitting the list becomes a branching node and opens a TCP connection for the overlay link to the sender of the original list, when the corresponding tree is activated. Finally, if such a list message arrives at the receiver (group member), then that node prepares the overlay link to the source of the message (normally the last branching point). Therefore, TCP connections for the overlay network links are only established between the source, branching nodes and the receivers (see also Figure 1) when the corresponding multicast tree becomes active. New nodes are added to the multicast tree by sending a list message including the new nodes as described before. Only when a tree is inactive, new nodes can be added, Therefore, no connections are established directly, but potential new overlay links are just prepared and only established upon activation of the tree. If a branching node now receives a list message with new nodes, it changes the source address of the list message, splits the list if required, and forwards it/them further. If a forwarding node has to become a branching node, it prepares the overlay link to the source of the list message, splits the message and forwards the new list messages further as described before. This new branching node tells the source of the original list message which receivers it handles in the future. Therefore, the previous branching node removes the overlay link that previously was using this new branching node as forwarder. Upon reactivation of the modified tree, the overlay link connections are opened from the source, via branching nodes to the the receivers. New and old receivers then become aware of their new group membership or of the change of branching nodes for existing group memberships. Nodes can also be removed using remove list messages accordingly and the forwarding and branching nodes have enough information (as with adding new nodes) to modify the resulting tree.

In the source-driven centralized approach, the source node determines all required branching nodes ahead. Therefore, the source also creates the complete distribution tree that is required for a multicast group. The branching nodes are then notified, process the information and further forward these notifications. If new receivers need to be added to a tree while it is inactive, the source calculates the new and/or modified branching nodes. Only these nodes are notified about the changes in the tree, branching nodes that do not need to be changed require no notification. Removing receivers from the tree is done similarly, branching nodes that need to be modified or removed are notified accordingly.

For the centralized receiver-driven approach, the join messages from the receivers are forwarded to the source, which manages the tree as described for the source-driven centralized approach.

In the receiver-driven de-centralized approach, receivers send the join message to their neighbor responsible for the default route. If this node is not a forwarder or branching node for that group it becomes a forwarding node (only knowing that it is on the path of an overlay link when the tree would become active) and forwards the join message further. Intermediate nodes, which are already branching nodes of the requested group drop the join message and prepare the overlay link to the new receiver. Forwarding nodes receiving join messages, become new branching nodes, prepare the new overlay link and send this information (about becoming a branching node) towards the source, dropping the original join message. A branching node receiving such a message modifies its overlay link in that direction. Therefore, the overlay link of which the new branching node has been acting just a forwarder before, is removed and replaced by an overlay link to this new branching node. Receivers that want to leave a group send a leave message towards the source. Forwarders on the path update their status for that group and forward the leave message further. Branching nodes receiving a leave message update their status, remove the overlay link to the leaving node, and discard the leave message. If the branching node has just one overlay link left, it has to change its status to a simple forwarding node for the remaining receiver and removes the affected overlay link. Further, it sends a notification towards the source and all intermediate nodes update their states accordingly. They forwarding the message until it reaches a branching node, which then establishes the overlay link to the remaining receiver.

To support end-to-end reliability in overlay multicast, the receivers have to acknowledge the receipt of each multicast message or acknowledge the receipt accumulated after a series of messages. Branching nodes aggregate and forward the acknowledgments. In case of missing acknowledgments, they send negative acknowledgments further towards the source. Branching nodes also take care of retransmission of lost packets and therefore need to cache the multicast data up to a certain degree. Hop-to-hop reliability is supported by underlaying protocols as described in Section 2.2.3.

2.2.2 Reliable IP-based Multicast

Contrary to Overlay Multicast (which uses TCP) we do not have a reliable end-to-end transport protocol. Instead we are using UDP. End-to-end reliability is realized using acknowledgment messages as described above. Branching nodes know only that their one-hop neighbors are forwarding the packets on their behalf. Acknowledgments be handled on one-neighbor basis (hop-to-hop), and not between branching nodes as for Overlay Multicast.

For the source-driven de-centralized approach the source sends the join list to its direct neighbors that should act as forwarders. The next forwarder is determined on a hop-to-hop basis. If a node has to become a branching node, it remembers from which neighbors it expects acknowledgments. Joins and leaves are handled by appropriate messages that could cause forwarders to become branching nodes (and vice versa) triggering a modification of the expected acknowledgments state for a node.

In the centralized source-driven approach the source sends the list of all branching nodes to the closest branching node, which processes and forwards them further to the nearest branching nodes on the path. Intermediate nodes become forwarders and store the status for the involved multicast group. Acknowledgments are handled directly between the branching nodes. Additional joining nodes trigger an update of the affected branching nodes all initiated directly by the source. Leaves are handled accordingly triggering updates of the branching and forwarding nodes involved.

In the receiver-driven centralized approach joins are sent to the source, which then acts as in the source-driven centralized approach. In the de-centralized receiver-driven approach, joins cause intermediate nodes to react as in the Overlay Multicast case. They either become forwarding nodes for that group if they are not handling that group yet or become branching nodes if applicable. Acknowledgments are handled the same way as described above in the de-centralized source-driven approach.

2.2.3 Protocol Stack

Figure 2 shows a possible protocol stack for reliable multicast in WSNs. End-to-end reliability for reliable IP-based Multicast is ensured by REMC (Reliable Multicast) and for Overlay Multicast by SNOMC (Sensor Network Overlay Multicast). To avoid unnecessary end-to-end retransmission or caching in branching, we use a hop-to-hop reliable network protocol realized by H2HR (Hop-To-Hop Reliability) in combination with the MAC protocol. This allows to directly delete cached packets after successful transmission to the next hop. To optimize the mentioned protocols, information is exchanged across different layers using the cross layer interface. For example, the MAC layer informs H2HR about the successful (or unsuccessful) transmission of a packet. H2HR is caching the packet until it has been transmitted successfully. Additional neighborhood information for deciding how to forward multicast packets and the involved paths is also requested from the cross layer interface by REMC and SNOMC. TSS (TCP Support for Sensor Nodes) [2] supports optimizations of TCP-specific mechanisms, such as intermediate caching, local retransmission and acknowledgment recovery and regeneration. Additional energy-saving is achieved by disabling the radio interface by the MAC layer when no transmission is required.

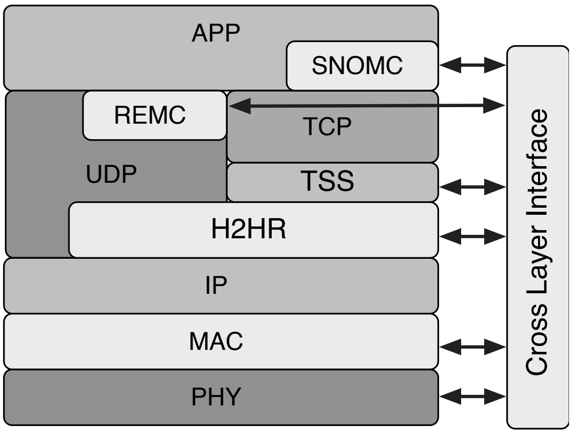

Figure 2: Protocol stack of a wireless sensor node using reliable unicast and multicast communication

3. DISCUSSION & CONCLUSION

We have shown several possible design concepts for reliable multicast in WSNs. Each approach its own advantages and limitations, depending on the scenario in which it is used.

For small scale WSNs, the centralized approach helps to save energy and resources on the sensor nodes because the source (generally the base-station) is handling the tree management. The sensor nodes do not need to store a lot of status information.

The de-centralized approach is useful in large scale environments, where robustness and easier tree construction can be achieved by letting the sensor nodes manage the tree construction and group handling themselves.

The Overlay Multicast approach is easy to implement but triggers more control messages in the underlaying layers as well as in the overlay management layer. By distinguishing between active and inactive trees, we can reduce the energy consumption due to the fact that we only establish TCP connections when data has to be transmitted. This results in an Overlay Network established on demand.

On the other hand, reliable IP-based multicast using UDP could be even more efficient and energy-consumption friendly, but requires more cross layer interactions and higher implementation effort.

4. OUTLOOK

There are still many open questions regarding the design of multicast support in WSNs. When using reliability on end-to-end basis, problems such as acknowledgment implosion, handling of negative acknowledgments, etc. have to be analyzed in more detail. Generally also congestion control and the resulting limitations have to be considered. Multicasting on the link-layer could also improve the performance in combination with our presented approaches. To determine efficiency, energy consumption and overall performance, we plan to simulate the different scenarios and solutions in the OMNET++ simulator [12]. Both approaches should be implemented on real sensor nodes using Contiki [13] as operating system.

5. REFERENCES

[1] G. Wagenknecht, M. Anwander, T. Braun, T. Staub, J. Matheka, S. Morgenthaler: MARWIS: A Management Architecture for Heterogeneous Wireless Sensor Networks. *International Conference on Wired/Wireless Internet Communications (WWIC'08), Tampere, Finland, May 2008.*

[2] M. Anwander, G. Wagenknecht, T. Braun: Management of Wireless Sensor Networks using TCP/IP. *International Workshop on Sensor Network Engineering (IWSNE'08), Santorini Island, Greece, Jun 2008.*

[3] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, N. D. Georganas A Survey of Application-Layer Multicast Protocols *Communications Surveys & Tutorials, 9(3), 58-74, 2007*

[4] K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim A survey and comparison of peer-to-peer overlay network schemes *Communications Surveys & Tutorials, 7(2), 72-93, 2005*

[5] A. Okura, T. Ihara, and A. Miura: BAM: Branch Aggregation Multicast for Wireless Sensor Networks. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference (MASS'05), Washington, DC, USA, Nov 2005.*

[6] A. Sheth, B. Shucker, and R. Han: VLM2: A Very Lightweight Mobile Multicast System For Wireless Sensor Networks. *IEEE Wireless Communications and Networking Conference (WCNC'03), New Orleans, LA, USA, Mar 2003.*

[7] R. Flury, and R. Wattenhofer: Routing, Anycast, and Multicast for Mesh and Sensor Networks. *IEEE International Conference on Computer Communications (INFOCOM'07), Anchorage, Alaska, USA, May 2007.*

[8] B. Chen, K. Muniswamy-Reddy, and M. Welsh: Ad-Hoc Multicast Routing on Resource-Limited Sensor Nodes. *International Workshop on Multi-hop Ad-Hoc Networks (REALMAN'06), Florence, Italy, May 2006.*

[9] J. S. Silva, T. Camilo, P. Pinto, R. Ruivo, A. Rodrigues, F. Gaudncio, F. Boavida: Multicast and IP Multicast Support in Wireless Sensor Networks. *Journal of Networks, 3(3), 19-26, Mar 2008.*

[10] D. Koutsonikolas, S. Das, Y. C. Hu, and I. Stojmenovic: Hierarchical Geographic Multicast Routing for Wireless Sensor Networks. *International Conference on Sensor Technologies and Applications (SENSORCOMM'07), Valencia, Spain, Oct 2007.*

[11] J. A. Sanchez, P. M. Ruiz, and I. Stojmenovic: Energy Efficient Geographic Multicast Routing for Sensor and Actuator Networks. *Computer Communications, 30(13), 2519-2531, Jun 2007.*

[12] OMNeT++: Discrete Event Simulation System. *Website: http://www.omnetpp.org.*

[13] A. Dunkels, B. Grönvall, T. Voigt: Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. *EmNetS'04, Tampa, FL, USA, Nov 2004.*