Andreas Weigel, Christian Renner, Volker Turau, and Holger Ernst. Wireless Sensor Networks for Smart Metering. In Proceedings of the IEEE International Energy Conference, EnergyCon '14, Dubrovnik, Croatia, May 2014.

# Wireless Sensor Networks for Smart Metering

Andreas Weigel [#1], Christian Renner [*2], Volker Turau [#3], Holger Ernst [†4]

[#] *Institute of Telematics, Hamburg University of Technology*
*Schwarzenbergstr. 95E, 21073 Hamburg, Germany*
[1] andreas.weigel@tuhh.de
[3] turau@tuhh.de

[*] *Institute of Computer Engineering, University of Lübeck*
*Ratzeburger Allee 160, 23562 Lübeck, Germany*
[2] renner@iti.uni-luebeck.de

[†] *EMH metering GmbH & Co. KG*
*Neu-Galliner Weg 1, 19258 Gallin, Germany*
[4] holger.ernst@emh-metering.com

***Abstract*—Smart metering will change the way we assess and control energy consumption. The necessary two-way communication between energy utilities and smart meters yet requires further research. In particular, cheap, low-maintenance, and reliable alternatives to existing approaches must be analyzed. We believe that self-organizing wireless sensor networks are a promising candidate. In this paper, we perform a requirement analysis for smart metering using wireless sensor network technology; we propose an architecture including a set of protocols; and we conduct a real-world evaluation with a testbed of 64 smart meters. With regard to latency and network formation time, the system was able to fulfill the identified requirements. The results show that the attained reliability for some traffic patterns is low. Further analysis revealed the use of link-layer broadcasts for one-to-many communication as a cause.**

## I. Introduction

The term *smart grid* describes a vision of a modern electric power grid infrastructure for more sustainable energy production, management and consumption. In short it consists of a bundle of technologies from diverse areas such as information and communication, sensing and metering, and energy storage and distribution. The smart grid wants to counteract the challenges arising from the worldwide explosion of energy demand, the direct impact of energy supply on climate changes and the integration of renewable energy sources into the aspired *energy mix* [9].

One building block of the smart grid is *smart metering*. It consists of a two-way communication network to break the existing information asymmetry between electricity producers and consumers. Apart from a cost reduction of the billing process the main goal is to enable a demand response system between the two parties [20]. Such a system has potential to flatten bursty consumption curves by shifting consumption times. These shifts could be triggered by providing real-time price information to customers, which can be used by smart appliances to schedule their operation. Smart metering systems will foster the development of *virtual power plants*, i.e., clusters of distributed energy generation facilities [14]. These have the ability of load-aware power generation at a short notice.

The term *Advanced Metering Infrastructure* (AMI) refers to the infrastructure that enables the integration of smart metering into the smart grid. The major components of an AMI are the metering devices at the customer sites and a data concentrator (DC) used to readout the metering devices of a defined area, to transfer the data to the back office system of the utility companies, and to deliver data from the back office to individual meters. The communication infrastructure of an AMI can be split into two parts: communication between the DCs and the back office and communication between the DC and the meters. While the Internet—based on existing technologies such as DSL or UMTS—provides an ideal infrastructure for the former part, there exists no widely accepted solution for the second part. The main requirements of this part are low costs for maintenance and installation and flexibility w.r.t. new installations and services.

These requirements match the promises made by self-organizing wireless sensor networks (WSN) over the last decade. The main contribution of this paper is a proposal for an architecture including a set of protocols for a wireless network connecting a DC and the meters. This is the result of a joint project with a European producer of electricity meters, and the protocols form the basis of forthcoming product development. The paper includes a discussion of the implementation of the protocols and the results of an evaluation in a field test.

## II. State of the Art

In the following, we introduce existing techniques for smart meter communication and sketch their advantages and disadvantages.

*Powerline Communication (PLC).* PLC is a natural solution from the perspective of utility companies, because it is based on an existing communication infrastructure, hence being the most cost effective solution. However, in [18] Skriver has documented poor performance on meter readings due to grid disturbances, mainly caused by low-energy light bulbs. Other studies revealed that the characteristics of the PLC vary geographically [1].

*Cellular Network Communication.* Another option is to rely on traditional mobile communications, as discussed in [11]. While this solution simplifies setup and bears no need for installing additional infrastructure, it has several drawbacks, e.g., it infers notable communication cost and suffers from connectivity problems in many installation locations, such as cellars.

*Broadband Internet.* More and more households have broadband Internet that could be used for smart meter communication. However, this option is impractical, because using a customer's Internet connection implies legal issues (e.g., there has to be a contract between the end user and the utility company). Moreover, meters are rarely installed close to the modem or router, hence requiring Wi-Fi, which increases unit cost and energy consumption.

*Wireless Sensor Network Technology.* Wireless sensor networks operate in license-free frequency bands and enable communication at virtually zero operation cost. They have recently been adopted for metering installations [5]. At 868 MHz, communication is even possible through concrete walls. Research has produced numerous protocols for data collection (e.g., CTP [4] and Arbutus [13]) and dissemination (e.g., Trickle [7], DIP [8] and CBFR [16]). There are several proposals for introducing reliability to a wireless sensor network at the transport layer. Among them are RCRT [10], Flush [6] and PIP [15]. Many of the presented ideas are combined within our protocol and complemented with functionality for missing features such as smart meter registration and association.

## III. Requirements Analysis

To develop a wireless sensor network architecture that complies with the needs and standards of real-world smart metering, we first carried out a requirements analysis for the communication between the DC and the meters. This analysis was conducted in collaboration with an industrial project partner in the smart meter branch. In the following, we present details of our main findings.

*Communication Partners.* In general, the targeted smart metering network consists of three entities: smart meters, a data concentrator, and mobile clients.

- Smart meters record, e.g., consumption statistics and load profiles, and are able to execute commands and perform firmware updates. They are equipped with a low-power microcontroller and a wireless communication unit for easy and cheap deployment. To be attractive for the end user, their purchase price and their energy consumption must be within statutory provisions.
- Data concentrators are the gateway for a network of smart meters. They are higher-power computing devices which are the data sink for collection of meter data and they distribute, e.g. commands or firmware updates. We will use the term data concentrator and data sink interchangeably in the remainder of the paper.
- Mobile clients are operated by technicians of utility companies. They are intended to obtain manual readings from the smart meters and network statistics without physical access. Within this paper, we do not further elaborate on this requirement.

*Communication Patterns.* The predominant use case of the smart grid is to collect smart meter readings. Here, the number of data sinks should be small to reduce operation costs. This implies that a single data sink should collect data from as many smart meters as possible. However, in rural regions and in harsh communication environments—e.g. cellars—direct communication is impossible, so that many-to-one communication must be performed via multiple hops.

Data distribution is the second most relevant use case. It comes in two different flavors of one-to-many communication. The first is the installation and update of tariff tables on the smart meters regularly with low latency. The second is needed for individual ad-hoc queries, e.g., to install a user-specific tariff table, remote monitoring and error analysis (e.g., in case of customer complaints or meter failure). Additionally, support for firmware updates should be available. To achieve low costs, these should be automatically distributed over the air, i.e., without any technician involved.

Apart from data collection and distribution, there must be a mechanism for new meters to join an existing network. This particularly involves registration and association. Here, the new meter must communicate with an unknown data sink. As an indicator for successful integration into the network for a technician installing the meter, the duration of a successful association should not exceed certain limits. More details on these limits are given in Section V.

*Addressing.* Smart meters are assumed to have an immutable built-in, unique identification number of at least eight octets. Since packet-oriented low-power radios restrict packet size (usually to 128 byte), efficient, network-wide addressing requires shorter addresses. Unfortunately, these numbers (i.e., their coding) are manufacturer-dependent, so that a generic mapping of identification numbers and short network addresses is impossible. Pseudo random assignment (e.g., using a hash function) introduces the risk of address collisions and therefore requires collision resolving. Since smart meters have to sign into a network (see above), we suggest a centralized assignment of 2 byte network addresses by a data sink.

*Quality of Service.* Smart metering requires reliable, end-to-end communication to ensure that all data from the meters arrives at the data sink and commands or configuration data is received by a meter. There also exist data-dependent latency demands—e.g., while forwarding meter readings allows delays of several minutes to even hours, sending commands to individual meters should net exceed a few minutes.

*Independence of Metering Format.* Industry has produced different meter data formats that are used and implemented by the manufacturers. To decouple the data network from the underlying smart meters, all network protocols must be data-format-agnostic. In particular, data fragmentation is required to transport larger data blobs—the smart meters used for the evaluation may produce readings with sizes of up to 10 kbyte.

*Legal Issues.* In addition to functional requirements, legal issues have to be considered[1]. The most important aspect is that of channel access, which is defined by the European Telecommunications Standards Institute (ETSI) [3]. We target the 868 MHz frequency band—the 2.4 GHz ISM band is massively used by Wi-Fi, Bluetooth etc., resulting in heavy interference [2]—where channel access requires the observance of a pre-defined, network-wide maximum duty cycle or the application of a combination of frequency hopping and listen-before-talk.

Utility companies desire high-resolution and timely collection of meter readings, implying high data volume using slow wireless links. To reduce network communication and improve bandwidth utilization, lossless data compression is the only allowed method, as in many countries smart meter readings must be transmitted as a stand-alone recoverable entity, even forbidding differential data transmission. While being an important technique for smart metering, we do not elaborate on data compression further within this paper as it can be seen as an orthogonal approach reducing the data volume.

## IV. Software Architecture and Protocol Stack

To cope with the identified requirements (see Section III) we developed and implemented communication protocols mainly at the network and the transport layer. At link layer level the listen-before-talk (LBT) mechanism as described in [3] was used to satisfy regulatory requirements. Note that while a smart meter's energy consumption should stay within certain bounds, it is also attached to a permanent energy source and we did not focus on low power transmission techniques.

### A. Network Layer

Protocols on the networking layer can be divided into three major tasks:

- ”many-to-one“ communication, i.e., unicast from nodes towards the sink
- ”one-to-many“ communication, i.e., unicast from the sink towards nodes
- association and registration of nodes with/at a sink

*1) Many-to-One Communication:* We realized many-to-one communication with a tree routing protocol. Routing paths are built by means of beacon messages, which are broadcasted by each node once in a network-wide fixed period. Each beacon contains the hop count (i.e., distance to the sink in hops), the path-ETX (expected number of transmissions along the path to deliver a single packet), and a sequence number. Each node stores this information and tracks the link-PRR (packet reception rate) of its neighbors. Here, the link-PRR towards a neighboring node $x$ is continuously updated in each beacon period by applying an exponentially weighted moving average (EWMA) filter according to

$$\mathrm{PRR}_{x,n} = \begin{cases} \alpha_h \cdot \mathrm{PRR}_{x,n-1} + (1-\alpha_h), & \text{beacon rcvd} \\ \alpha_h \cdot \mathrm{PRR}_{x,n-1}, & \text{no beacon rcvd} \end{cases} \tag{1}$$

Updating the PRR once per period overcomes the problem of stalled values for deteriorating or dead links, as found in [17]. For $\alpha_h$, we used a conservative value of 0.95. Due to memory constraints, the maximum size of the neighbor table is limited. If it is full, a newly identified neighbor replaces the entry with highest path-ETX among a set of candidates. This set of candidates consists of all nodes with a higher hop count than the new neighbor, all blacklisted nodes (see below), and nodes whose link-PRR has deteriorated below 67%.

After the first beacon period, a parent is chosen from the set of neighbors based on the total path-ETX (combined neighbor's path-ETX and link-PRR). In case of a broken link, i.e., a negative confirmation from the link layer, a node blacklists its parent and selects a new parent as explained above. Moreover, nodes re-evaluate their decision after each beacon period. The parent is replaced by the (non-blacklisted) neighbor with lowest path-ETX, if the latter has

- a smaller hop count and lower path-ETX, or
- same hop count and a path-ETX smaller by a small threshold, or
- a higher hop count but a path-ETX smaller by a large threshold.

The thresholds are used to prevent nodes from rashly switching to a new parent; we used values of 0.25 and 3, respectively. While the former value aims at reducing churn within the routing tree in the presence of short-term link variations, the latter prevents routing loops in most cases by discouraging nodes to choose a parent further down in the tree. If a really ”bad“ neighbor with low hop count is coincidentally chosen as parent, the detection of link breakages for data transmissions and the blacklisting mechanism prevent this node from disconnecting a node indefinitely.

*2) One-to-Many Communication:* One-to-many routing with a classic routing table necessitates an entry (destination, next hop) for the set of reachable nodes, which in the worst and not completely improbable case comprises the

[1] Note that in this paper, we do not elaborate on security aspects

whole network. Therefore, we only use a subtree table at each node, which reduces the memory demand by 2 byte per entry by omitting the next hop. One-to-many traffic packets are only re-broadcasted if the destination address is found in the subtree table and an associated one byte counter is below a threshold. This counter is incremented for each re-broadcast and reset for each received many-to-one packet. Within a network with stable routes, packets are thus only re-broadcasted along a single path. To construct this subtree table, the network layer observes many-to-one traffic and thereby learns about the nodes within its subtree. The expected regular network traffic suffices to build up reverse paths; in particular, nodes always use (N)ACKs at the transport layer for enhanced reliability, causing the counter to be reset regularly.

*3) Association and Registration:* To enable nodes to find nearby networks and join them to acquire a short network identifier, we designed an association protocol: Nodes willing to join a network broadcast an association request, which is forwarded by already associated nodes towards their corresponding sink. Several mechanisms are deployed to enhance the association procedure and to prevent a message explosion within the network:

- Potential forwarders back off and check if other nodes forward the same association request; if so, they suppress forwarding.
- Starting with short intervals, nodes send out association requests in exponentially increasing intervals, enabling short response times for single nodes joining the network while reducing the caused network load in situations where many or all nodes want to join a network simultaneously (e.g. after a blackout)
- Nodes refrain from forwarding twice within a certain interval, which depends on a node's active time (since last reboot); this mechanism limits generation of network load by potentially malicious entities (note that before association, communication between arbitrary nodes is extremely hard to secure).

The sink reacts on an incoming association request by assigning a short network identifier to the node and sending back an association reply. This reply is sent via the standard one-to-many routing protocol back to the entry node, which broadcasts it for the originating node. While this message exchange can be utilized for mutual authentication and/or the deployment of a network key, security concerns and mechanisms were not within the focus of our study.

### B. Transport

A protocol which we named reliable block transport (RBT) provides reliable end-to-end delivery of large[2] data blobs. RBT uses a three way handshake for connection setup, negative acknowledgments (NACKs) to provide reliability and round-trip time (RTT) estimation for timely retransmissions. The RTT estimation is inspired by TCP [12] and is done by the receiver every time the next message after a NACK is received, using EWMA filtering for the RTT itself and its variability RV according to

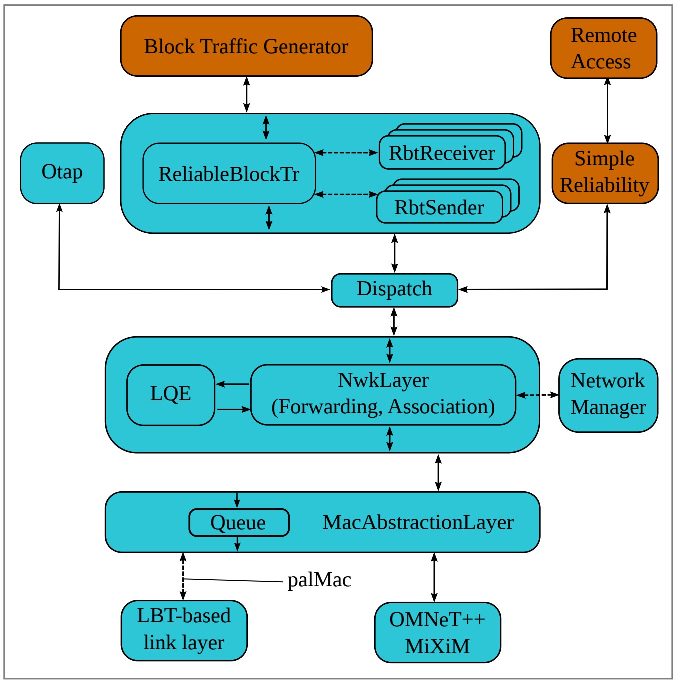

Figure 1. Modules of the protocol stack; Dashed line represent direct function/method calls, solid lines OMNeT++-style message passing

$$\mathrm{RV}_n = \beta \cdot \mathrm{RV}_{n-1} + (1-\beta) \cdot |\mathrm{RTT}_{n-1} - \mathrm{RTT}_{\mathrm{new}}| \quad (2)$$

$$\mathrm{RTT}_n = (\alpha \cdot \mathrm{RTT}_{n-1}) + (1-\alpha) \cdot \mathrm{RTT}_{\mathrm{new}} \quad (3)$$

$$\mathrm{RTO} = \mathrm{RTT}_n + 2 \cdot \mathrm{RV}_n \quad (4)$$

(with $\alpha = 0.9$ and $\beta = 0.75$) and results in the actual timeout value RTO. Receivers keep track of the sequence number up to which fragments have been completely received, a list of missing fragments and a list of out-of-order fragments. This information together with the current estimate for RTO are passed back to the sender within a NACK message. NACKs are sent by the receiver when missing packets are detected or the sender asks for one because either its sending window is approaching its boundary or the transmission is finished. In order to allow several block transfers simultaneously, the concept of ports is introduced to the protocol. Thereby, multiple instances of senders and/or receivers at the same and different ports can be created to handle protocol execution for several transmissions. The tuple (destination address, destination port, source address, source port) uniquely identifies a certain traffic flow and is bound to one sender and receiver instance each.

[2] large compared to the max. frame size of 127 byte (minus protocol header overhead)

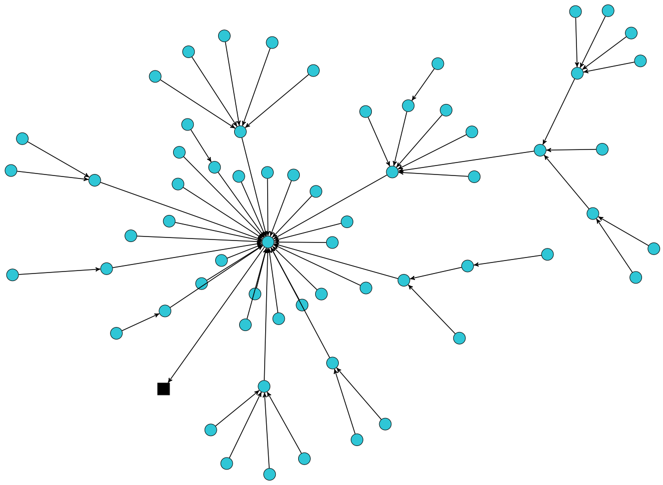

Figure 2. Snapshot of routing paths within the network, sink depicted as black square; positions only reflect logical, not physical locations

## C. Implementation

Figure IV-C visualizes the described structure of our protocol stack, which was implemented for the CometOS framework [19]. By means of CometOS, we were able to use the protocol implementation above the link layer within the OMNeT++[3] simulation environment as well as on our hardware platform, which is based on a LPC1763 ARM Cortex-M3 microcontroller and the sub-1 GHz transceiver CC1120. The transceiver used narrow band channels of a width of 25 kHz, using GFSK modulation. With this setup, a theoretical datarate of 19.2 kbps was achieved. The protocol logic is implemented in terms of CometOS modules which are mapped to OMNeT++ modules within the simulation environment. Using the facilities of the framework to create a python base station—that connects to a real-time-scheduled simulation or a real test bed—we were able to test and verify the developed protocol code and control our experiments.

# V. Evaluation

To evaluate our protocol stack, we deployed 64 smart electricity meters, which were equipped with our transceiver module (see Section IV-C), over three floors of an office building and conducted experiments over a duration of three weeks. All presented results in this section were collected from this testbed. Figure 2 shows a snapshot of the routing configuration of the network. The results are compared against the requirements we developed with our project partner.

## A. Data Traffic

We identified latency requirements for the regular collection of metering data from the whole network of down to 1 h and for individual communication of 10 min (e.g., commands for load shedding) to 1 h. We restricted our experiments to four traffic patterns that are expected to represent these use cases. A data transfer of a data block of 1 kbyte, for both many-to-one and one-to-many communication directions, represents the collection of large load profiles from the meter and the setup of a meter with a large configuration file, respectively. Additionally, the transmission of smaller command messages is represented by sending data packets of 75 byte payload, again, for both communication directions. While the testbed was deployed, we could execute five runs for each traffic pattern. For the many-to-one communication patterns, transmissions were initiated by the sink one after the other. Latency was measured as the duration between passing the data request to the transport layer and retrieving its confirmation.

The results in form of scatter plots for one experiment are shown in Figure 3. For this experiment we additionally introduced a rate restriction on the data link layer which was configured to wait for an average duration of approximately two transmissions before sending another packet. Rationale for this restriction was the high number of link-layer losses we could observe and the high probability of collisions caused by hidden terminals when multiple fragments of a large data block were sent out immediately one after the other.

As can be seen in Figure 3, the latency for the complete transmission of a 1 kbyte block reaches from 3 s to 6 s at nodes directly attached to the basestation node to about 12 s to 18 s at the nodes farthest down in the tree for both directions of communication. This easily fulfills the stated requirements for individual communication with a meter for small and large data blocks. Considering the collection of metering data, within the given network the latency requirements can be fulfilled. Note that the stated latency requirement of 1 h implies small data packets. Reading out complete load profiles, which causes large data blocks of 1 kbyte to 10 kbyte would only be required once a day.

On the other hand, we also observe, that a high variation of latency values for the one-to-many communication direction leads to comparatively large confidence intervals for some nodes with a distance of three or more hops. Whereas for the small data packets the BRR is almost 100 % for all nodes, we can observe the loss of exactly three large data blocks for the many-to-one direction and an even larger number of losses for the one-to-many direction, with the BRR of some nodes approaching 70 %.

Though communication is always bidirectional, both effects can be explained with the fact, that the one-to-many communication does not utilize link layer ACKs and retransmissions (see Section IV): The actual data-carrying packets using this "less reliable“ routing direction are more numerous and much larger (75 or 90 byte additional data payload) than the corresponding end-to-end negative acknowledgments and therefore are more prone to being corrupted by collisions and bit errors. Another possible explanation for the unsatisfactorily low BRR can be found

[3] confer http://www.omnetpp.org/

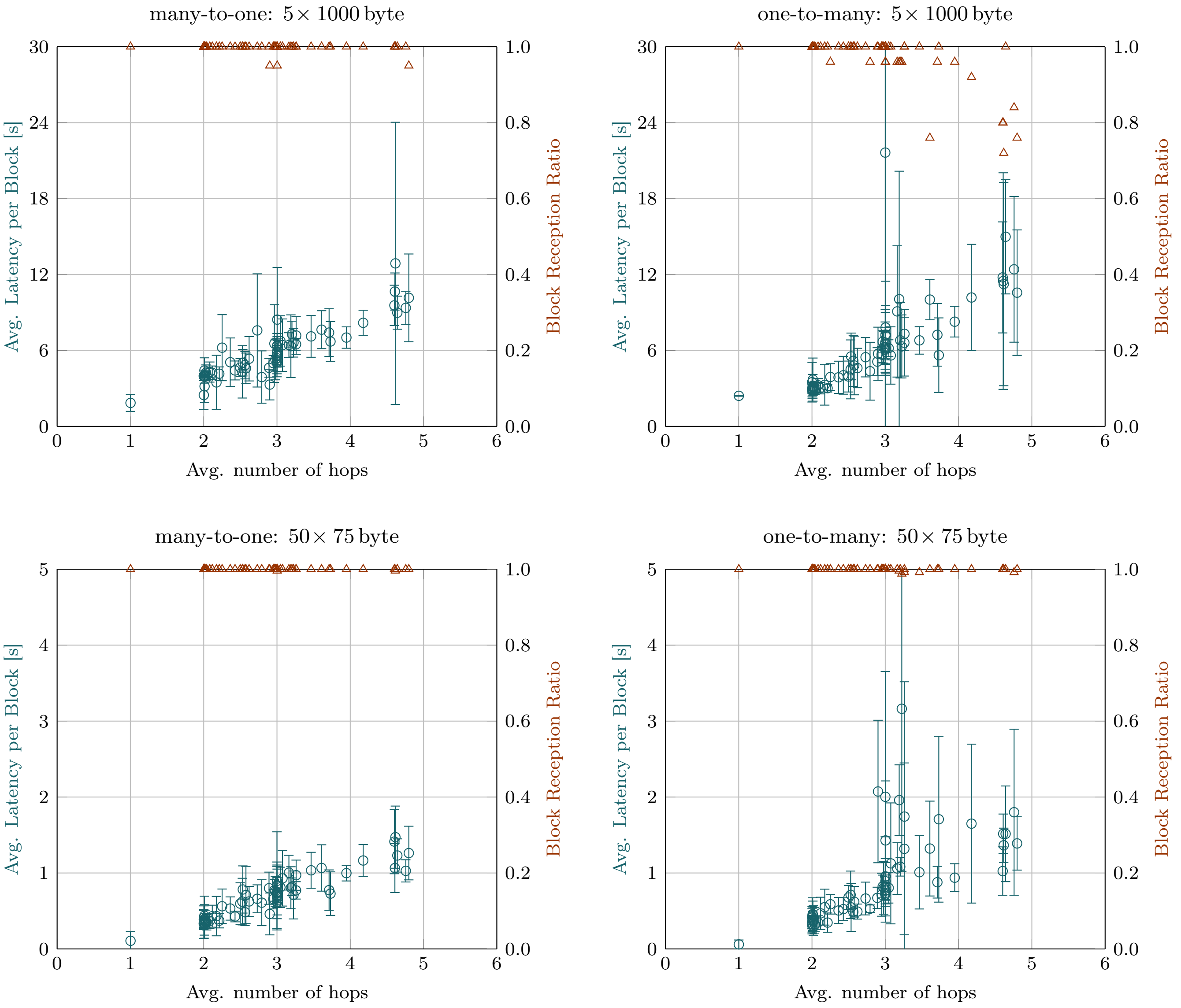

Figure 3. Latency and block reception rate of each node for the four traffic patterns; error bars denote 95% confidence intervals for latency; MAC inter-packet-interval: 100 ms

in the usage of a single fixed, albeit large[4], timeout at the receiver instances. With increasing round-trip times receivers may therefore have given up too early, i.e., while the sender (which uses RTO to determine its timeouts) was still trying to send packets, causing higher than necessary loss rates. Unfortunately, we could not verify the severeness of this issue due to the lack of corresponding logging data and could not repeat the experiment before the testbed had to be taken down again.

### *B. Network Formation*

Apart from the data traffic, we also measured the time it took for the whole network to form up. The requirements for the formation procedure were identified as 1 min for a single meter and a duration of up to 10 min for the whole network. We emulated a simultaneous network-wide reboot (e.g., the situation after a power outage) by sending each node a command to reboot at approximately the same time[5] and stored the duration until a node successfully completed its association. Additionally, we evaluated the duration of the association procedure for two small sets of nodes. The first was a group of seven spatially co-located nodes, the other a group of six nodes scattered over the whole network.

The results of these experiments are shown in Figure 4. Within the smaller node sets, all nodes manage to as-

[4] The used timeout was the same which was used for connection setup (5 s)

[5] starting with 60 s, we reduced an offset by half the measured round-trip-time

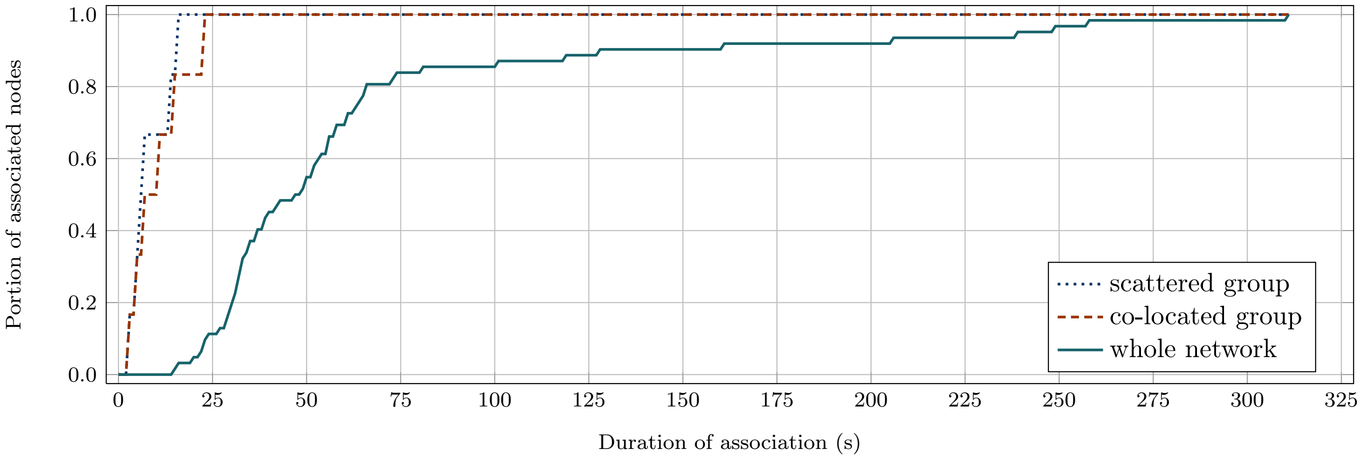

Figure 4. Ratio of associated nodes versus duration of association for three different

sociate with the sink within 23 s (co-located) and 16 s (scattered). During a whole network reboot, 80 % of the nodes are associated after 65 s, the remaining 20 % then need up to 311 s for successful association. This fulfills the stated requirements within our network and justifies the expectation, that for larger networks of up to 150 nodes, the requirements can be fulfilled as well.

### C. Analysis

While the requirements with regard to duration of association could be fulfilled within the deployed network, the observed overall reliability leaves room for improvement. A more thorough analysis of the link layer data concerning unicast packets revealed an overall link error rate, i.e., the final loss of a packet after a maximum of three retransmissions, of 0.5 %, measured over all nodes in the network. A single packet error is caused by either losing the message or the ACK. Under the simplified assumption, that both causes are equally probable, we can approximate the probability $p$ for a successful single packet transmission from the link error rate $p_{\text{fail},4}$ after a total of 4 transmission attempts by

$$p_{\text{fail},4} = (1 - p^2)^4 \quad \Longleftrightarrow \quad p = \sqrt{1 - \sqrt[4]{p_{\text{fail},4}}}. \qquad (5)$$

Over all links in the network, we get $p = 85.6\,\%$, which translates into a probability of failure of about 14 %. Considering, that for the constrained flooding approach we rely on link-layer broadcasts which are always unacknowledged and often have to travel multiple hops, we identify this comparatively high link layer error rate as one of the main causes for the observed reliability issues.

## VI. Conclusion

Wireless sensor networks present an attractive option to realize parts of an advanced metering infrastructure in many locations. To fulfill the identified requirements for such a network, we presented a protocol stack based on a self-organizing routing protocol, supporting bi-directional communication between meters and a data sink. An association service enables nodes to find and join a network and could be used as basis for security functions. Reliability and transport of large data items is provided by a block transport layer. The implementation of modules for CometOS greatly simplified testing and verification of all developed protocols within the OMNeT++ simulator before moving the same code to the actual hardware platform. We evaluated the implemented protocols within a deployment of 64 smart electricity meters equipped with our 868 MHz transceiver with regard to latency and block reception rate and network formation time. We identified the usage of link layer broadcasts for the one-to-many communication as responsible for the high loss rates and advise against using similar mechanisms in lossy networks. On the other hand, requirements with regard to latency and the duration of network formation could be fulfilled within our test network.

## References

[1] S. Bannister and P. Beckett. Enhancing Powerline Communications in the Smart Grid using OFDMA. In *Proc. Australasian Universities Power Engineering Conf.*, AUPEC, 2009.

[2] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. A. Zúñiga. JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation. In *Proc. IEEE Int. Conf. Inform. Proc. in Sensor Networks*, IPSN, 2011.

[3] Final draft ETSI EN 300 220-1 V2.4.1 (2012-01). Technical Report REN/ERM-TG28-434, ETSI, 2012.

[4] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection Tree Protocol. In *Proc. Conf. Embedded Netw. Sensor Systems*, SenSys, 2009.

[5] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and Implementation of a High-Fidelity AC Metering Network. In *Proc. Intl. Conf. on Information Processing in Sensor Networks*, IPSN, 2009.

[6] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica. Flush: A Reliable Bulk Transport Protocol for Multihop Wireless Networks. In *Proc. Intl. Conf. Embedded Netw. Sensor Systems*, SenSys, 2007.

[7] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks. In *Proc. Symp. on Networked Systems Design and Implementation*, NSDI, 2004.

[8] K. Lin and P. Levis. Data Discovery and Dissemination with DIP. In *Proc. Intl. Conf. on Information Processing in Sensor Networks*, IPSN, 2008.

[9] OECD. ICT Applications for the Smart Grid: Opportunities and Policy Implications. OECD Digital Economy Papers 190, OECD Publishing, 2012.

[10] J. Paek and R. Govindan. RCRT: Rate-Controlled Reliable Transport Protocol for Wireless Sensor Networks. *ACM Trans. Sen. Netw.*, 7(3):20:1–20:45, 2010.

[11] O. Pauzet. The Future of Smart Metering: The Case for Public Cellular Communications. *Metering Intl. China Edition*, (3):39–40, September 2011.

[12] V. Paxson, M. Allman, J. Chu, and M. Sargent. RFC 6298: Computing TCP's Retransmission Timer, June 2011. Status: Proposed Standard.

[13] D. Puccinelli and M. Haenggi. Reliable Data Delivery in Large-Scale Low-Power Sensor Networks. *ACM Trans. Sen. Netw.*, 6(4):28:1–28:41, July 2010.

[14] D. Pudjianto, C. Ramsay, and G. Strbac. Virtual power plant and system integration of distributed energy resources. *Ren. Power Gen. IET*, 1(1):10–16, 2007.

[15] B. Raman, K. Chebrolu, S. Bijwe, and V. Gabale. PIP: A Connection-oriented, Multi-hop, Multi-channel TDMA-based MAC for High Throughput Bulk Transfer. In *Proc. ACM Conf. on Embedded Networked Sensor Systems*, SenSys, 2010.

[16] A. Reinhardt, O. Morar, S. Santini, S. Zöller, and R. Steinmetz. CBFR: Bloom Filter Routing with Gradual Forgetting for Tree-structured Wireless Sensor Networks with Mobile Nodes. In *Proc. Intl. Symp. on a World of Wireless, Mobile and Multimedia Networks*, WoWMoM, 2012.

[17] C. Renner, S. Ernst, C. Weyer, and V. Turau. Prediction Accuracy of Link-Quality Estimators. In *Proc. Europ. Conf. on Wireless Sensor Networks*, EWSN, 2011.

[18] G. Skriver. Smart Grids Turn to Wireless Systems. Technical report, EETimes, 2012.

[19] S. Unterschütz, A. Weigel, and V. Turau. Cross-Platform Protocol Development Based on OMNeT++. In *Proc. Intl. Wksp. on OMNeT++*, OMNeT++, 2012.

[20] W. Wang, Y. Xu, and M. Khanna. A Survey on the Communication Architectures in Smart Grid. *Comput. Netw.*, 55(15):3604–3629, October 2011.