Packet-Oriented Communication Protocols for Smart Grid Services over Low-Speed PLC

Michael Bauer, Wolfgang Plappert, Chong Wang, Klaus Dostert
Institute of Industrial Information Technology
Universität Karlsruhe (TH)
Karlsruhe, Germany
E-mail: bauer@iiit.uni-karlsruhe.de

***Abstract*— Advanced Metering Infrastructures require reliable communication links. This paper proposes a protocol stack for smart metering based on the parameters of a robust low-speed PLC physical layer. The protocol stack under consideration implements IPv6 on top of a contention-free media access control sub-layer, thereby allowing for maximum flexibility. Simulation results show that this protocol stack allows data transfers at acceptable latencies, despite low data rates at the physical layer.**

***Keywords*—IPv6, protocols, power line communication, smart metering**

I. INTRODUCTION

WITH the increasing awareness regarding the scarcity of resources, the need arises for new technologies leading to a more efficient end-use of energy [1]. In this context, terms like Automatic Meter Reading (AMR), Smart Metering [2], and Advanced Metering Infrastructure (AMI) become increasingly important. Such terms commonly imply the need for cross-linking spatially distributed devices like electronic meters and an infrastructure for data collection and processing. For such automation technological applications, highly reliable and available communication systems are indispensable.

Power line communication seems desirable as an access technology for communication [3]. However, it is difficult to guarantee communication system reliability and availability under extremely difficult channel conditions as offered by the PLC transmission channel [4],[5]. The channel properties provide numerous challenges when designing a protocol stack for low-speed PLC. Additionally, application-specific design guidelines need to be taken into account, which can be challenging, if neither standardized data formats are available nor the amount of data to be transferred is known precisely.

Our approach to protocol design focuses on economic efficiency, flexibility, and scalability. For the upper layers of the OSI model [6], we use IPv6 at the Internet Layer and TCP for the Transport Layer. At the Data Link Layer, we use polling as a contention-free MAC strategy, Automatic Repeat-Request (ARQ) and CRC for error control. The Physical Layer is based on OFDM, aiming at maximum reliability. The overall communication protocol stack is simulated and its performance evaluated by simulation in OPNET® Modeler.

II. PHYSICAL LAYER

Due to the unfavorable properties of the PLC transmission channel, the physical (PHY) layer does not allow for data rates comfortable for protocol design. Modems can feature a maximum of tens of kbit/s, if the system is designed for robustness and reliability. This especially holds true, if the frequency range between 9 kHz and 95 kHz [6] is used. The fact that this is only a narrow frequency band compared to broadband PLC is even worsened by the fact that it is only useful to utilize the upper frequencies of this spectrum due to the PLC channel properties. OFDM is an appropriate method for modulation, as explained in [7]. The problem of robust OFDM frame synchronization is solved by using the zero-crossings of the mains voltage, as described in [8]. Focusing on robustness, a low-speed PLC PHY can operate with the parameters in Table I, taken from [7].

TABLE I
LOW-SPEED PLC PHY PARAMETERS

Parameter	Type / Value
Modulation Scheme	OFDM (DBPSK)
Carrier Spacing	325.5 Hz
Number of Carriers	48
Net Bit Rate	Approx. 11 kbit/s

III. LAYER-2 PROTOCOL

MAC protocols specify a resource sharing strategy by controlling the access of multiple users to a shared transmission medium. Our design of a MAC protocol for PLC aims at avoiding collisions, therefore we select polling as the preferred MAC strategy. Due to its deterministic behavior, polling avoids untraceable complications caused by the so-called hidden-node problem [10].

A. Topology

The protocol is designed for a system where time allocation to customers within one low voltage "cell" is managed by a central unit. We denote this data concentrator as the master, whereas the meters within the cell are slaves assigned to the master. The protocol supports bidirectional communication between master and slaves. We distinguish so-called downlink (master to slave) and uplink (slave to master) modes. Within one cell, *N* slaves are connected to one master in a logical star configuration. Slaves may not exchange data directly between each other. However, every slave may communicate with other slaves under the administration of the master.

B. Polling strategy

The master polls the slaves in a cyclic order (i.e. 1, 2,…, N, 1, 2…). After completing a visit to slave *i*, the master waits for a specified time. The period during which the master continuously serves slave *i* is called a service period of slave *i*, the subsequent period is called a switch-over period for slave *i*. The sum of service and switch-over period is defined as the “master-slave-period” (M-S-period). For more detailed information and analysis methods, we refer to [11]. Fig. 1 displays the separate uplink and downlink polling cycles.

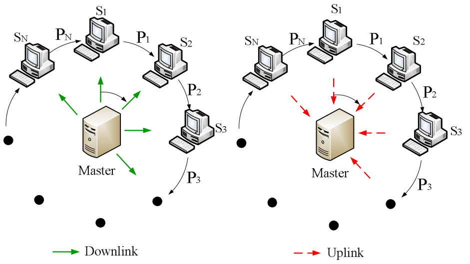

Fig. 1 Downlink and uplink cycle

With increasing number of slaves, the time between two consecutive grants of access to the communication medium becomes longer for each of the slaves. To meet real-time requirements, a gated polling policy is preferred over exhaustive or limited-1[11]. Fig. 2 depicts the *N*-th master-slave period schematically. The service period is limited by a timer at both master and slave.

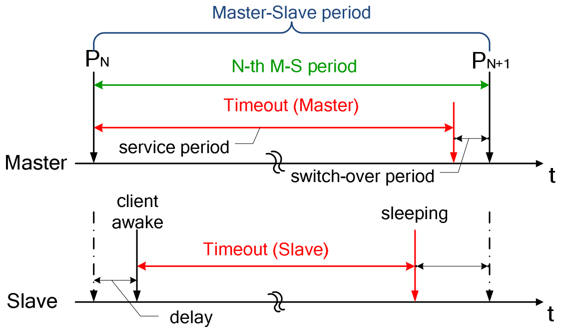

Fig. 2 *N*-th Master-Slave period

C. ARQ – Error handling

ARQ is preferred over FEC for error control. There are three basic types of ARQ protocols including Stop-and-wait-, Go-Back-N- and Selective Repeat ARQ. Because the latter two types of ARQ protocols require an independent reverse channel, these ARQ types are not suitable for PLC. Fig. 3 shows the pattern of Stop-and-Wait ARQ. A positive acknowledgement (ACK) from the receiver indicates that the transmitted packet has been received successfully, the transmitter may send the next packet in its queue. A negative acknowledgement (NAK) from the receiver indicates that the transmitted packet has been detected as faulty, retransmission is required as a result.

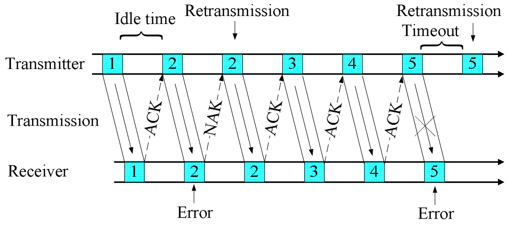

Fig. 3 Stop-and-Wait ARQ

D. Frame structure

Fig. 4 depicts the structure of a MAC frame used in our model. Table I shows the corresponding frame types.

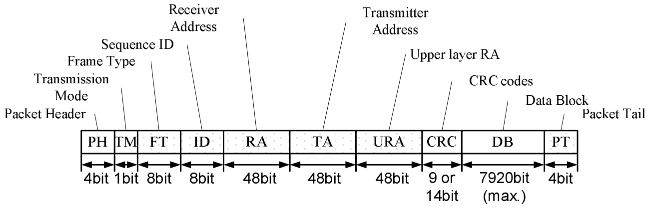

Fig. 4 MAC frame structure

The flag “ID” should only be set if the frame type is DATA, ACK or NAK. RA, TA and URA always have to be set. DB is only set if the frame type is DATA. The length of the CRC field depends on the frame type and varies between 9 and 14 bits. The total length of the MAC frame without the Data Block field is 178 bits, the maximum length is restricted to 8103 bits including data. In this case, the payload contributes 97.74% to the total frame length.

TABLE I
POSSIBLE "FRAME TYPES"

Frame Type	**Function**
POLL	Token, Polling
RTS	Request to send
CTS	Clear to send
DATA	Data payload
ACK	Positive acknowledgement
NAK	Negative acknowledgement
MASTER END	Timeout at master
SLAVE END	Timeout at slave

E. Timing during a Master-Slave-Period

The timing diagram of one M-S-period in downlink and uplink mode is depicted in Fig. 5 and Fig. 6, respectively. Between the transmission of a POLL frame and the exchange of user data, an RTS/CTS handshake is used to ensure that the channel is free, thereby avoiding deadlocks. In downlink mode, the POLL-frame performs the same task as an RTS.

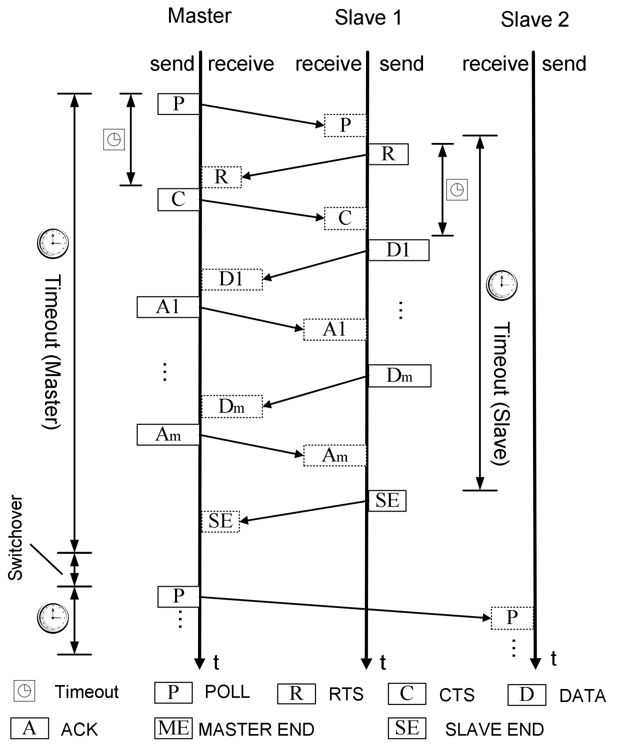

Fig. 5 Timing Diagram for uplink mode

A timer clock is started at the instant the master sends a POLL frame. Another timer is started at the slave node at the instant it receives the POLL frame. The slave timeout is defined for a shorter period than the master timeout in order to avoid collisions. If the timer at the master (slave) expires, it sends a MASTER END (SLAVE END) frame. The MASTER END (SLAVE END) may also be sent before the timer expires. This happens in the case when the master (slave) has no packets queued for transmission in downlink (uplink) mode.

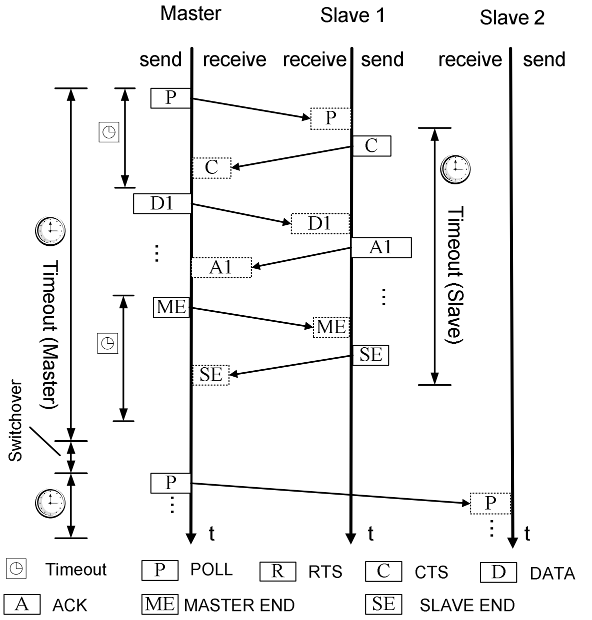

Fig. 6 Timing Diagram for downlink mode

F. *Collision avoidance*

As shown in Fig. 7, collisions may happen at the instant the slave's timer expires. If the slave immediately sends a SLAVE END frame, the incoming ACK or DATA frame will clash. Fig. 7 shows how these collisions are avoided by the slave delaying the transmission of the SLAVE END frame.

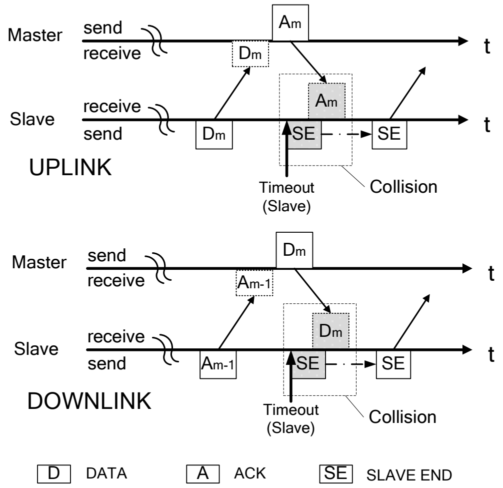

Fig. 7 Possible collisions and avoidance strategy

IV. HIGHER LAYERS

A. *Network Layer*

On the network layer, the Internet Protocol in version 6 (IPv6) [13] is employed due to its larger address space compared to IPv4 and its convenient hierarchical addressing scheme. In [12], the minimum host requirements of IPv6 for Low Cost Network Appliances (LCNA) are discussed. As the intended application relies on embedded systems with limited resources, i.e. electronic meters, the rules for such end nodes apply.

Address resolution from IPv6- to MAC addresses is provided by the Neighbor Discovery Protocol that strongly relies on the broadcast ability of the MAC layer [14]. To reduce the communication effort from N^2 messages necessary to resolve MAC- and IPv6 addresses of all N slaves, a start-up phase has been implemented in which every slave sends an Unsolicited Neighbor Advertisement to the master's MAC address. Additionally, the master acts as a public Neighbor Cache that answers to Neighbor Solicitations with authoritative, valid Neighbor Advertisements. This is possible as any slave-to-slave communication has to be handled by the master, and helps reducing Neighbor Discovery-related broadcasts to a minimum.

B. *Transport Layer*

On the transport layer, the Transmission Control Protocol is employed, providing a communication service with congestion control, fragmentation, checksum calculation and connection tracking. The correct order of packets is ensured with sequence numbers and deadlocks are avoided by means of timers. Due to selective retransmission mechanisms, Fragmentation on TCP layer is more efficient than on the network layer: if a fragment on transport layer is lost, only the missing datagram needs to be retransmitted.

C. *Application Layer*

A very simple file transfer protocol is employed for the transmission of user data. It is considered TFTP-based [15], relying on TCP. A small request data packet is sent from the master to every slave. Every slave then answers with its metering data packets. Separate acknowledgement packets for every datagram are not necessary at application layer, as the correct reception is already confirmed by TCP ACKs.

TABLE II
ND PARAMETER SETTINGS

Constant	specified value	used value
Router Lifetime	0-9000 s	9000 s
Router Advertisement Retransm.	3	3
Minimum Delay Between Router Advertisements	> 3 s, < 1800 s	1800 s
Maximum Router Advertisement Delay Time	0-0.5 s	0-0.5 s
Router Reachable Time	< 3600	43200 s
Maximum Neighbor Solicitations	3	3
Maximum Neighbor Advertisements	3	3
Node Reachable Time	3000 ms	43200 s
Neighbor Solicitation Retransmission Timer	1000 ms	300-900 s

V. SIMULATION

A. *System Capacity Estimation*

The total number of slaves and the amount of user data that can be transmitted by each slave can be derived numerically from an empirical formula which approximates waiting and handshaking times by taking into account the average bus utilization *bus_util*. With *N* slaves and *n* requests per hour, the amount of data on MAC layer (*mac_amount* bits consisting of MAC-, IPv6-, TCP header and user data) and a data rate of *bus_speed* in bit/s of the bus, this yields equation (1).

$$dataratio = \frac{N \cdot mac\_amount \cdot n}{bus\_speed \cdot bus\_util \cdot 3600} \tag{1}$$

The value of *dataratio=1* represents a system operating under full load. In order to ensure system stability, equation (1) should return values less then 0.85, which has been verified by simulation.

B. *Settings*

All simulations were performed in OPNet Modeler (Educational Version 14.5, Build 7545 32-bit). The scenario was chosen to be on the edge of stability according to equation (1) with the following parameters: 100 slaves are connected via a bus link with a defined data rate of 11000 bit/s, according to the parameters of the physical layer described in Section II. To ensure representative stability analysis, all simulations are performed over a period of 24 hours that starts immediately with the start-up phase. To obtain comparable runtime delay values of IP packets, the master sends an ICMPv6-“Echo Request” packet containing 64 bytes of data to all slaves after 900 seconds. After one hour, the actual metering process starts, which gives the system enough time to stabilize after the Neighbor Discovery and delay determination phase. All slaves are polled every 15 minutes by the master with a 256 bytes request packet on the application layer that is answered by 5500 bytes of metering data by each slave.

The user data transmission process has to be completed in less than 900 seconds (15 minutes), after which the next request is sent. The maximum packet size on MAC layer is set to 990 bytes. This is a compromise between throughput (which is affected by MAC overhead) and the time left for one slave to occupy the medium without any possibility for other interaction. Another important point is the optimum packet size as a base for CRC algorithms on the MAC layer. The 5500 bytes of user data lead to 6 fragments per metering data set per slave which have to be transmitted and acknowledged on the MAC layer. The timing parameters on the MAC layer allow the transmission of 2 such packets per master-slave-period, the maximum number of unacknowledged packets is one (“stop and wait”-algorithm).

Timing settings on TCP layer have to be adapted to enable connection establishment despite the large delays between consecutive packet interchanges. Therefore, the initial retransmission timeout is set to 600 seconds.

Timing settings for the Neighbor Discovery Protocol (see Table II) are set to limit broadcast traffic and retransmissions to a minimum, as permanent topology changes in the observed network segment are considered to be rare events. Nodes being unavailable is assumed to be a temporary state. The expiration time of router entries in the router cache of slaves violates the defined values, but as the whole system breaks down if the master is unavailable, this can be accepted with the goal to reduce higher layer protocol traffic. For the Neighbor Solicitation Retransmission Timer, the same delays have to be taken into account as for TCP retransmission timeouts. Therefore, the value is dynamically chosen between 600 and 900 seconds.

VI. RESULTS

A. *Bus Utilization*

The bus utilization is the instantaneous utilization of the bandwidth provided by the PHY in percent. It indicates how efficiently the protocol uses the available bandwidth. It is, for example, decreased by waiting times on MAC layer. On the contrary, a value of 100% represents full utilization of the medium. At the beginning of the simulation in Fig. 8, a value of about 80% is shown.

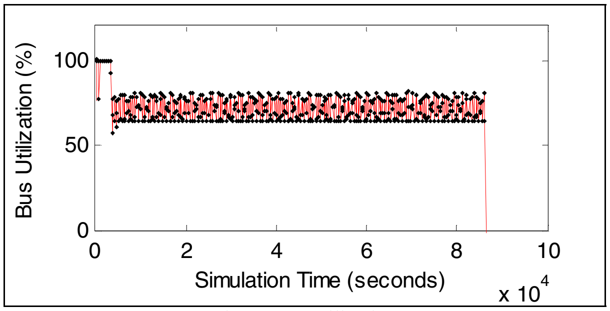

Fig. 8 Bus Utilization

This represents the start-up phase where all slaves and the master exchange their address information with Unsolicited Neighbor Advertisements. In the subsequent time span, only MAC protocol data units are exchanged – the available data rate of 11000 bit/s is fully utilized. After 900 seconds, the initial IPv6 delay measurement with ICMPv6-"Echo Request" packets is shown. After 3600 seconds, the master application transmits the first request packet which leads to a TCP handshake for connection establishment. The following constant exchange of datagrams leads to an average Bus Utilization of around 72%.

B. *Uplink Cycle Length*

The uplink cycle length specifies the amount of time the master needs to poll all 100 slaves once and to collect user data. This cycle length directly influences delay times between consecutive accesses to the communication medium by a slave. If no slave has any packets waiting in its queue, one cycle takes about 8 seconds. Depending on the amount of packets waiting in a slave's queue, this process can take up to 200 seconds (see Fig. 9). The exchange of address information as well as the delivery of ICMPv6 packets leads to a cycle length of 40 to 45 seconds. After the start of the application, there still do exist cycles with the minimum length of 8 seconds. This suggests that the system is still stable: not every polling cycle has to be used for exchanging data units.

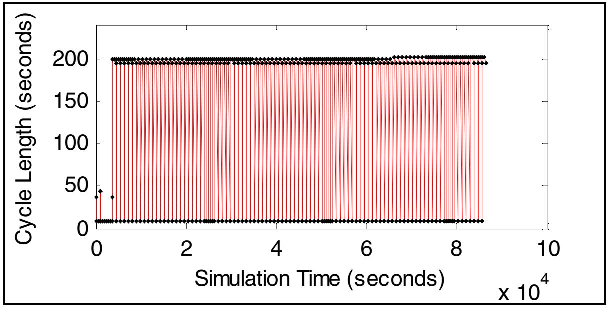

Fig. 9 Uplink Cycle Length

C. *Downlink Cycle Length*

The Downlink Cycle measures the amount of time the master needs for delivering data once to each of the 100 slaves. If there is no data waiting in a particular slave's queue in the master, the next slave is contacted. In the beginning, in Fig. 10 one can see the cycle in which the Neighbor Advertisements and, after 900 seconds, the ICMPv6 "Echo Requests" are delivered. After 1 hour, the peak cycle length of nearly 90 seconds is caused by the establishment of TCP connections to all 100 slaves and the subsequent delivery of the request packet on the application layer. As the amount of data delivered to the slave mainly consists of the request packets and acknowledgements on TCP layer, the mean duration is lower than the one of the uplink cycle length.

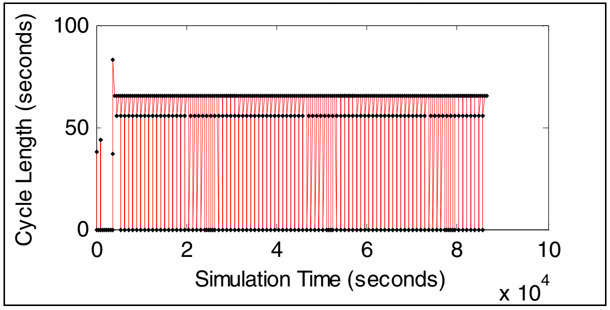

Fig. 10 Downlink Cycle Length

D. *IP End-to-End Delay*

On IP layer, the end-to-end delay between two nodes is measured between the time of creation of an IPv6 packet and the time of reception within the IP module of the receiver. It is influenced by the waiting time in the MAC queue of the transmitting node and the transmission duration. The initial delay determination only depends on the MAC algorithm and the number of slaves. It results in about 50 seconds, which is directly related to the transmission time of the packets on the bus and the waiting time of the ICMPv6 "Echo Reply" in the slaves' queues, influenced by the Uplink Cycle Length. The time delay varies between 50 and 350 seconds (see Fig. 11) with an average of about 150 seconds. The time span is directly related to the number of packets already waiting in the queue of slave 1 and to the length of both uplink and downlink polling cycles during the waiting time in the queue.

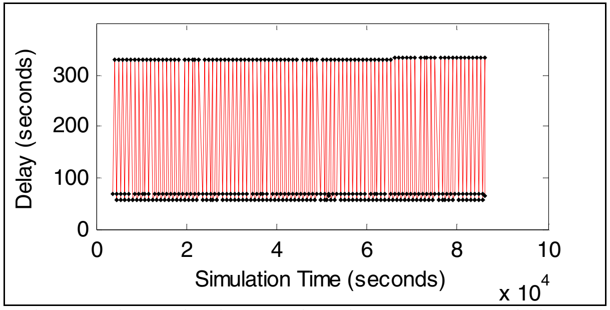

Fig. 11 End-to-End Delay on IP layer between Master and Slave 1

E. *Application Delay*

The application delay is measured between the transmission of the request packet and the complete reception of all fragments belonging to a metering data record. After having sent the request packet to slave 1, the master constantly has to wait about 600 seconds until all 6 fragments of the 5500 byte metering data set are received and reassembled (see Fig. 12).

This is less than the period of 900 seconds between two consecutive meter data requests. The system is stable.

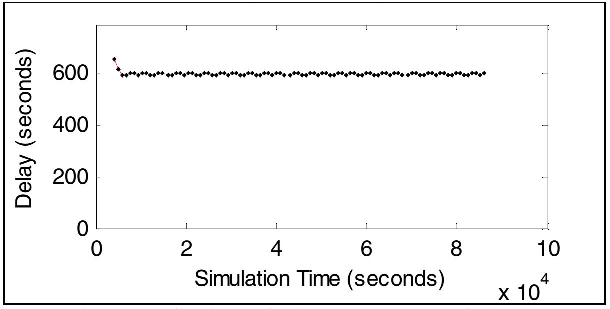

Fig. 12 Application Delay between Master and Slave 1

F. Parameter Studies

To simulate different preconditions, several parameters have been varied. On the one hand, the number of slaves participating in communication has been altered. As this directly influences the length of a polling cycle, both delay and throughput are affected. With more slaves, the delay between two consecutive accesses to the medium increases, as well as the total amount of data that has to be delivered in time. The amount of metering data must be reduced with increasing number of slaves to keep the system stable. Table III shows the results.

TABLE III
IP END-TO-END DELAY

Number of slaves	**Configuration**	**IP End-to-End Delay**
50	initial delay measurement	< 40 seconds
50	data transfer (12000 bytes per slave per metering data set)	< 600 seconds
100	initial delay measurement	< 70 seconds
100	data transfer (5500 bytes per slave per metering data set)	< 400 seconds
200	initial delay measurement	< 150 seconds
200	data transfer (2700 bytes per slave per metering data set)	< 700 seconds

The higher IP end-to-end delay regarding data transfer with 50 slaves is caused by the larger amount of data that can be transmitted without yielding an unstable system (compare equation (1)). As 2 packets are transmitted per master-slave-cycle, the larger amount of data leads to more packets remaining in a slave's queue at a cycle's end, which then leads to longer delays between creation of an IPv6 packet an its delivery. The effective throughput of user data on application layer, calculated on a 900 seconds basis, is shown in Table IV.

TABLE IV
DATA THROUGHPUT

Number of slaves	**Transferred Data Amount per Request**	**Average Data Rate**
50	12000 byte + 256 byte	5447 bit/s
100	5500 byte + 256 byte	5116 bit/s
200	2700 byte + 256 byte	5255 bit/s

VII. CONCLUSION

Despite the considerable overhead caused by the large IPv6 header and also TCP, the protocol suite is applicable even at low PHY Layer data rates. In combination with the robust MAC algorithm, acceptable delay times and throughput can be achieved. The effective throughput on the application layer shows that the system behaves linearly. The overhead at MAC layer stays constant and only slightly decreases the usable data rate. Less than 50% of the available data rate of 11000 bit/s are used for bus arbitration and packet acknowledgements.

ACKNOWLEDGMENT

The authors kindly thank OPNET Technologies, Inc.® for their support in terms of the OPNET® University Program [16].

REFERENCES

[1] European Parliament, European Council, "Directive 2006/32/EC on energy end-use efficiency and energy services," April 2006.

[2] A. Moreno-Munoz, J.J.G. De La Rosa, "Integrating power quality to automated meter reading", *IEEE Industrial Electronics Magazine,* vol. 2, issue 2, pp. 10-18, June 2008

[3] G. Deconinck, "An evaluation of two-way communication means for advanced metering in Flanders (Belgium)", *IEEE Instrumentation and Measurement Technology Conference Proceedings, 2008*

[4] M. Goetz, K. Dostert, A Universal High Speed Powerline Channel Emulation System, *International Zurich Seminar on Broadband Communications, 2002*

[5] J. Bausch, T. Kistner, M. Babic, K. Dostert, Characteristics of Indoor Power Line Channels in the Frequency Range 50-500 klz, *IEEE International Symposium on Power Line Communications and its Applications, 2006*

[6] ISO/IEC 7498-1, Open Systems Interconnection – Basic Reference Model: The Basic Model, 1994

[7] CENELEC EN 50065-1, Signaling on low-voltage electrical installations in the frequency range 3 kHz to 148.5 kHz, 1991

[8] T.Kistner, Ein neuartiges mehrträgerbasiertes PLC-System mit störresistenter Synchronisation, Universitätsverlag Karlsruhe, 2008

[9] T. Kistner, M. Bauer, A. Hetzer, and K. Dostert, Analysis of Zero Crossing Synchronization for OFDM-Based AMR Systems, *IEEE International Symposium on Power Line Communications and its Applications*, *2008.*

[10] S. Mangold et al., IEEE 802.11e Wireless LAN for Quality of Service, Proceedings European Wireless, vol. 18, 2002

[11] H. Levy and M. Sidi, "Polling Systems: Applications, Modeling, and Optimization," IEEE transaction on communications, vol. 38, pp. 1750-1760, October 1990.

[12] Nobuo Okabe et al., "Host Requirements of IPv6 for Low Cost Network Appliances", http://tools.ietf.org/html/draft-okabe-ipv6-lcna-minreq-02, IETF, December 2002.

[13] Deering, S. R. Hinden: Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), 1997. Updated by RFC 5095.

[14] NARTEN, T., E. NORDMARK, W. SIMPSON H. SOLIMAN: *Neighbor Discovery for IP version 6 (IPv6)*. RFC 4861 (Draft Standard), 2007.

[15] SOLLINS, K.: *The TFTP Protocol (Revision 2)*. RFC 1350 (Standard), 1992. Updated by RFCs 1782, 1783, 1784, 1785, 2347, 2348, 2349.

[16] OPNET Technologies, Inc., http://www.opnet.com/