Tunneling Smart Energy Protocols over ZigBee

Rolf Kistler, Marcel Bieri, Rolf Wettstein and Alexander Klapproth
Lucerne University of Applied Sciences and Arts, CEESAR-iHomeLab
Technikumstrasse 21, 6048 Horw, Switzerland
{rolf.kistler, marcel.bieri, rolf.wettstein, alexander.klapproth}@hslu.ch

Abstract

In the course of increasing the energy efficiency, simplifying processes and providing new customer services, millions of smart meters will be rolled out during the next few years. Wireless personal area network (WPAN) technologies play a major role in the deployment of such systems. The ZigBee Alliance noted the big potential lying beneath smart energy solutions and developed the Smart Energy profile, an application protocol that could take the role of the missing interoperable standard. However, electronic meters have been around for quite some time now and the industry has spent much effort on creating powerful and popular metering protocol standards such as DLMS/COSEM or IEC62056-21. A possible solution combining state of the art wireless mesh network technology and existing metering standards is to provide a mechanism to transport metering PDU over ZigBee. This paper elaborates the requirements of such an approach, proposes a design and discusses the results of a first implementation tunneling DLMS over ZigBee that can also be adapted to other domains.

1. Introduction

The common denominator for the definition of a smart meter is an electronic box that measures the electricity, gas, heat or water consumption and is equipped with at least one two-way communication interface. Besides the precise measurement followed by the storage and real-time communication of energy data, smart meters may provide various additional services. These allow for external displays visualizing energy data or showing custom text messages, enable demand response and load control applications, handle tariff- and pricing information, detect tampering or outage and so forth. Driven by new regulations and the energy utilities themselves, millions of such smart meters and associated systems are going to be installed in the near future to form powerful advanced metering infrastructures (AMI). Such AMI shall save both energy and money through improved transparency and awareness of the energy usage, accurate real-time data, higher quality prognoses, reduced peak loads, simplified read out and billing procedures, process automation and new tariff structures as well as innovative energy services.

In a home, several smart (sub-)meters can be connected to a local home area network (HAN) together with other upcoming devices such as in-home displays, load controllers, smart appliances, smart thermostats and a gateway connecting them remotely to the overall AMI. Due to their well-known advantages, mainly derived from having no need for new cabling, wireless technologies play an important role in the deployment of such HAN. With the progress in hardware transceiver development and through application of state of the art software protocol design, several WPAN technologies have evolved recently, ready to penetrate the market. ZigBee is one of them. As for the ZigBee Alliance, Smart Energy (SE) has proven to be a strong force driving the further development of their technology and the standard in general. Besides the mesh routing capabilities in the network layer (NWK) of the IEEE 802.15.4 based protocol stack, a major value of ZigBee lies in its concept of describing complete applications in a standardized way. The cluster library (ZCL), providing common functionality to all nodes, and the notion of a standard application profile made it possible to develop a definition of a new application standard, the ZigBee Smart Energy profile [17]. Leading players of the energy industry took part in its development and for several countries, ZigBee seems now to be the choice when it comes down to smart metering. The profile supports all the features of modern smart meters and the associated devices in a HAN. As the profile operates on the application layer, it could also easily work on top of other communication technologies such as Ethernet/IP or Powerline. And in fact, the HomePlug Powerline Alliance has already chosen to adopt the profile enabling interoperable SE networks across different physical media types [5].

Despite the recent evolutions, electronic meters have existed for several years now and the metering industry has spent much time and money on creating powerful ways to model, describe and read out metering data. As a result, metering protocol standards have evolved such as DLMS/COSEM [4] or IEC62056-21 [7]. These standards are widely used today, provide extensive tool support and define tailored features that go far beyond of what has been specified within the scope of the SE profile. While there is a good chance that the new ZigBee profile will be widely adopted, continuing support for the other existing metering

protocols is a must. Concerning ZigBee as a data transport vehicle, the metering community sees in a wireless mesh communication protocol a highly appreciated alternative to existing (wired) transport media. It all boils down to the idea of tunneling existing metering protocols over ZigBee. The alliance allowed for it when it included the *SE Tunneling (Complex Metering) Cluster* into the profile. However, so far neither the commands and attributes of this new cluster nor the tunneling mechanisms themselves have been specified (t.b.d in r15 of the profile [17]). What's more, the requirement of supporting existing protocols also arises in other domains, namely in building automation where BACnet [1] is broadly accepted.

The remainder of this paper focuses on the design of a generic protocol tunneling mechanism and discusses a concrete implementation that encapsulates DLMS frames and sends them over a ZigBee mesh network. The results are part of the EnerBee applied research project that seeks to create a reliable, monitored network infrastructure for wireless advanced metering devices [6].

2. Requirements and Assumptions

The main requirement originates from the fact that legacy systems have to be supported out of the box: Plug and play. So in an ideal case, any existing device can be connected to a wireless module and is immediately accessible without any configuration effort or the need for new tools. That implies a kind of transparent data channel or wireless cable replacement. Still, the frames transferred over the air must comply to the ZigBee standard enabling both the SE profile features specified so far as well as the new tunneling mechanism.

In general, metering applications do not pose strict performance requirements on throughput, round-trip times or real-time behaviour of the data communication channel. And although data volumes of certain actions can be quite considerable, there is no time constraint if such a task takes several minutes to be accomplished. A typical load profile in a residential environment holds, in a conservative scenario, 2 energy values of 2 tariffs stored every hour and is read once a day. Together with status information and time stamps, this results in approximately 550 bytes of payload data. Setups with values stored in quarter hour periods or even less are under discussion. And as soon as configuration actions or firmware uploads are involved, the payload increases to several kB.

As a must for EnerBee, the DLMS protocol and its applications have to be supported. This allows the designer to derive concrete assumptions as a basis for further considerations: First, DLMS is a request-response protocol. Therefore the assumption of half-duplex traffic is made in which each complete request protocol data unit (PDU) to a destination will be followed by a response PDU to the original source. To cope with retransmissions, fragmentation and flow control on the wireless link, there is a requirement stating that the communication timeouts within the application protocol need to be configurable in a broad range [1]. DLMS timeouts can be tuned to an extent that makes it even suitable for data transports across slow mobile (GPRS) networks. What is not needed is the handling of parallel data streams or multiple sessions. Each device is only required to cope with one active tunnel at a time [2].

The question arises whether ZigBee itself is suited for tunneling? While the EnerBee project is also concerned with challenges coming up regarding the very nature of any wireless system (signal absorption, interference, power consumption, configuration and diagnostics...), the tunneling mechanism itself accounts for a ready-to-use and reliable network infrastructure and assumes that ZigBee is able to provide it (also connecting meters in basements). And in case frames are lost, the application protocol (DLMS) is responsible to repeat incomplete requests.

A topic popping up in every tunneling system is the packet size and the overhead involved in the protocol. The PDU of the application protocol can be way bigger than what is supported by IEEE 802.15.4/ZigBee in which the payload per frame may be reduced down to as few as 70 bytes depending on the features used. PDU sizes also influence the memory requirements of the restricted embedded ZigBee nodes. Both topics are handled more closely in the fragmentation section bellow.

Not to be forgotten are security and privacy aspects that are critical for the acceptance of any SE system. Security must be applied in terms of data encryption and, even more important, authentication. The ZigBee core and the SE Profile support both features natively, which distinguishes this standard from many other wireless protocols.

Finally, the lifetime of metering solutions needs to be taken into account. Up until now, metering infrastructures were in the field for 15-20 years without replacement or serious maintenance efforts. Investors intend to calculate on the same basis for upcoming systems.

3. Design

Several design aspects have to be considered in order to fulfil the given requirements. Fig. 1 shows the network that serves as example within the scope of this paper. As alluded above, it is assumed that the network has been set up properly and all nodes joined correctly, including SE security. On one end of the HAN there is the ESP (Energy Service Portal), a device specified in the SE profile, acting as a gateway to a remote read-out station. On the other end there are the metering devices accessible over redundant routing paths. A DLMS read-out request will reach the ESP from remote and will be tunneled transparently to the metering devices, where it is processed, the response prepared and returned.

[1] In a broad range means up to several seconds, e.g. in the case of a response timeout

[2] Except from standard ZigBee routers (or Range Extenders as they are called in SE). These devices do not know anything about the SE application protocol as they just route ZigBee frames on the NWK layer

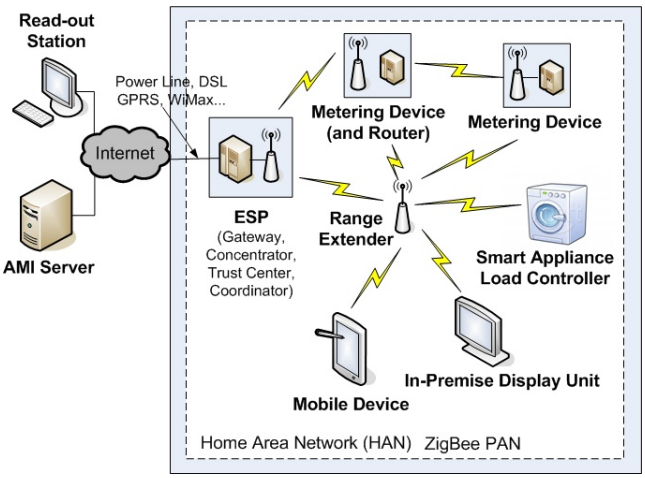

Figure 1. Smart Energy HAN

3.1. Addressing

An important issue to be solved is the mapping of different address schemes. Meters have their own addresses which are not compliant with ZigBee node addresses [3]. The meter address is embedded in the metering application PDU. As the initial sender may not even be part of the HAN, it is neither aware of the exact path the frame takes, nor does it know about ZigBee and its addressing scheme. There are two principle design approaches which influence the implementation of the nodes receiving a request (e.g. the ESP or a mobile configuration device):

- **Metering Address Known:** The node getting the metering request is able to read the destination address out of the PDU and map it to the according ZigBee node address.
- **Metering Address Unknown:** The node getting the metering request does not look into the PDU and directly forwards it to the network using a mechanism to assure the intended destination receives it properly.

In the first approach, the gateway node needs knowledge about each protocol that shall be tunneled. The incoming data stream must be parsed in order to identify the protocol and extract the meter address. In a next step, the node must become aware of the ZigBee address of the node to which the destination meter is attached to forward the data to it. It should also be kept in mind that it's possible that several meters are connected to the same ZigBee node. The ZigBee node initiating the tunnel has different ways to get to the meter/ZigBee destination address tuples. In a centralized variant, one node (and maybe a fallback node) holds a complete address lookup table with all meter/ZigBee tuples of the HAN. This node must be accessible by all other nodes as they will ask it for the matching ZigBee node address given a meter address. It's similar to the way the Internet's DNS system works. In a decentralized variant, each node builds up its own small address lookup table. It can be filled on demand using a kind of address resolution request similar to the ARP mechanism used in the Ethernet/IP world. Both variants have their well-known advantages and drawbacks concerning network traffic load, code complexity, resource needs, single-point-of-failure robustness, address caching, configuration and additional infrastructure requirements.

Manual configuration of lookup tables shall be avoided as far as possible. Either the node initiating the tunnel asks all potential communication partners for the matching address (pull) or these provide it e.g. in the course of a joining procedure (push). This also involves the nodes on the other end of the tunnel which must be aware of both their ZigBee address and the addresses of the connected meters to be able to share this information with others when responding to their address match requests. Ideally, there is a standard way to ask the meters connected to the ZigBee "modem" for their meter address and store it locally.

The second approach is not aware of the protocol that is being tunneled. The data stream is not analyzed but directly forwarded into the ZigBee network. To make sure all potential destinations get it, broadcast or multicast is utilized both of which ZigBee supports. This is done with the presumption that the destination node recognizes the PDU including its own address and will respond as soon as it has received a complete request. It just needs to be assured that it gets the request as a whole before any communication timer fires. An option to reduce network traffic is to send only the first packet to all potential receivers. When the associated response is received, the ZigBee destination address of the meter is known and all subsequent incoming data is directly forwarded in unicast frames to this destination. A few trade-offs accompany this approach: To start with, as the protocol is unknown, the frame borders of the metering PDU cannot be detected. Either, the frame structure is completely ignored in the tunneling or a kind of frame end timeout is introduced. What's more, it can neither be determined natively when a command to a meter is completed and a new one starts nor is it possible to handle parallel commands as the data streams cannot be correctly multiplexed without address information. A response timeout may serve as the means to find command borders. If no response is received to a unicast request, the mechanism is set back and retries it using multicast/broadcast frames. Another issue to keep in mind is ZigBee's lack for the fragmentation and link security of unacknowledged frames (multicast/broadcast). So the first request should not exceed 80 bytes and will not use a link key (just a network key). One of the trickiest tasks in this approach involves finding of a set of communication parameters that works well for all potential metering protocols and communication link types. Still, if the assumptions made in section *Requirements and Assumptions* are applied (half-duplex traffic, no parallel streams, configurable timing parameters, reliable end-to-end transport), which is possible for DLMS and in many other cases, the option is feasible.

[3] We focus on 16-Bit IEEE 802.15.4/ZigBee short addresses here, which are unique within a ZigBee personal area network (PAN). PAN and HAN can be considered as equivalents in this context.

The two approaches alluded above mainly influence the implementation. If the interfaces are designed in a way that supports both variants, it is up to the implementer to chose the solution that fits best. From the outer perspective of the read-out station on one end and the meter on the other, both systems should look the same. To sum up, knowing the protocol leads to a solution that handles parallel streams, does not rely on multicast/broadcast frames and may closely follow the specific characteristics of each protocol. The biggest drawbacks here result out of the fact, that the incoming data stream needs to be parsed. This increases the complexity of the code and multiplies the effort involved with each new protocol that shall be tunneled. If the protocol is unknown, the solution only operates under certain conditions. But if those are satisfied, the system becomes very flexible and easy to maintain.

3.2. Tunneling

As already stated, the tunneling shall act as an extension to the Smart Energy profile. This empowers the designer to use all the features of the ZigBee stack including the powerful cluster library (ZCL) and the application support framework (APS) implementing service discovery, binding, fragmentation etc. The metering data will be inserted as payload into ZigBee Smart Energy PDU. The questions come up on what features are missing and what's the information to be passed along the link to put tunneling into operation?

In the ZigBee standard, clusters were introduced to implement (and cluster) application specific functionality within so called commands and attributes. There is no reason to abandon this paradigm, tunneling in SE will be based on ZigBee clusters. In fact, work has already been done on a design level. The commercial building automation (CBA) profile task group, whose profile has not yet been officially released, has defined two clusters to tunnel the BACnet protocol [16].

The first cluster, called *Generic Tunnel*, is a candidate for ZigBee cluster library. It certainly makes sense to reuse it in SE and other profiles. The cluster defines two 16-bit attributes for the maximum transfer sizes (*MaximumIncomingTransferSize, MaximumOutgoingTransferSize*) and one that takes a meter address of any format fitting in a string with up to 255 characters (*ProtocolAddress*). The specified commands may serve to build up a distributed address matching mechanism based on push or pull as explained in the previous secion *(MatchProtocolAddress, MatchProtocolAddressResponse, AdvertiseProtocolAddress)*. Extending the generic cluster, the CBA profile draft introduces a second cluster, the BACnet cluster. It was specifically designed for this type of protocol. The cluster does not contain any attributes and only one command to transfer a BACnet network protocol data unit *(TransferNPDU)*. If, in the future, other protocols shall be tunneled in CBA, new clusters will be specified, one for each new protocol, implementing its specific requirements. An advantage of this approach is that service discovery permits discovering the protocol clusters and finding out which device in the network support which types of protocols.

As for the tunneling proposed here for SE, using specific clusters for each protocol shall be avoided in favor of a more generic solution. Principally, the *generic tunnel* cluster shall be used as a base. However, it lacks the crucial transfer command to actually send and receive data. Moreover, to provide the type of protocol discovery akin to CBA, an additional attribute containing information about which metering protocols a SE device supports would be convenient. On top of that, additional features have been identified easing tasks if mobile read-out or commissioning nodes are present in the HAN. There are two possibilities to include the missing functionality and remain ZigBee compliant:

1. Extend the generic cluster with manufacturer specific extensions (commands and attributes described above).

2. Define a second cluster for the SE profile intended to handle all potential metering protocols.

The decision was taken to leave the generic cluster as it is and include the missing functionality into one additional separate SE tunneling cluster.

3.3. Fragmentation

Looking at the need to split the data into fragments at the tunnel entry and reassemble them at the end comes down to two questions: What is the maximum transfer unit (MTU) of the protocol being tunneled and is it required to send this unit in one coherent data chunk over the ZigBee link (transaction)? [4] Further, a difference needs to be made whether the application protocol itself sends its data in packets (DLMS/COSEM) or just streams it generating a continuous flow of characters (IEC62056-21).

If the MTU of the metering protocol is bigger than the MTU of the SE protocol and a metering PDU is to be transferred in one transaction, fragmentation is needed. The fragmentation algorithm must handle retransmissions, duplicate rejection, flow control and congestion control automatically. To achieve this task, the transaction buffers in the sender and the receiver must be dimensioned to store one complete data chunk that is bound to be transferred in fragments over the air (most likely a PDU). So if an IP packet coming from an Ethernet link is to be transferred in one transaction, such a buffer must be big enough to hold 1500 bytes in the worst case. In many cases, allocating a buffer of this size on an embedded ZigBee node fails due to memory constraints. One solution would be to adjust the MTU of the metering protocol to assure that every transaction fits into the allocated buffer. This may go down to the size of the payload of a standard (non-tunneling) SE protocol command which always fits into a single IEEE 802.15.4 frame (127 bytes). On one hand, fragmentation is no longer needed here. On the other hand, the protocol

[4] In this paper, a transaction stands for a chunk of data that is passed to the ZigBee stack as a whole (in a transaction buffer), is sent fragmented over the link and leaves the stack at the other end reassembled again (in another transaction buffer).

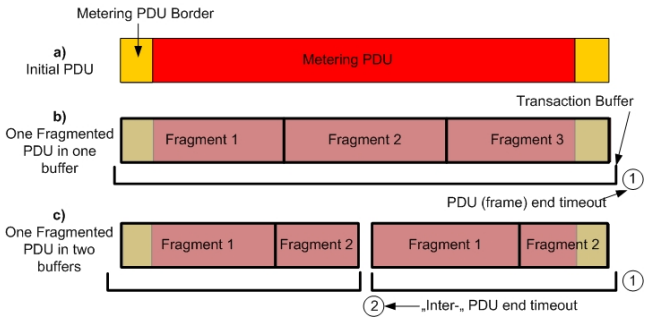

Figure 2. Fragmentation options

overhead increases drastically and the MTU must be configured beforehand.

The situation changes if the metering protocol can cope with transfer units being sent in separate transactions or if a protocol just streams data (Fig. 2). A simple brute force variant allocates a ring buffer that is big enough to avoid an overflow due to different link transfer speeds and processing power. It then just stores request data coming from remote into this ring buffer on one end and takes it out on the other completely filling and finally sending away transaction buffers (or simple IEEE 802.15.4 frames). In any case the receiver must be able to cope with the delays occurring between two transactions. These can be tuned to an extend in choosing transfer speeds and buffer sizes accordingly[5]. Even a combination is possible: Making the transaction buffer big enough to hold any transfer unit of the metering protocol with the biggest MTU and in case the buffer gets filled with streaming data, just send it away and hope the receiver is prepared for it.

Fragmentation of large acknowledged unicast transmissions was introduced with ZigBee 2006 as an optional feature and it became mandatory with the ZigBee PRO stack profile released in 2007. The SE tunneling proposal utilizes this fragmentation feature with the prerequisite that the MTU of the metering protocol is set to a size that fits the internal transaction buffers. In the tests involving DLMS, the highest MTU of the DLMS payload embedded in HDLC is 248 bytes [6]. Although the MTU can be reduced down to 62 bytes, the mechanism was laid out for the highest MTU, which results in a transaction of 4 (IEEE 802.15.4) fragments. The tunneling is intended to work for transactions up to 1500 bytes. Allocating this transaction size also means that, in case of simple water meters supporting IEC, a complete (streamed) read-out fits into one transaction, as it only produces a few hundred bytes.

3.4. Flow Control

If the assumptions made for DLMS are valid and all transaction buffers are big enough to hold data with the size of the highest MTU, additional flow control mechanisms are unnecessary. Each request PDU will fit into the buffers and the whole communication chain will wait for the response PDU which also fits. DLMS handles end-to-end flow control on the application layer itself. Anyway, it was decided to extend the tunneling proposal with an SE flow control mechanism to relax constraints for other protocols. Five general scenarios have been identified that the mechanism shall help preventing (Fig. 3):

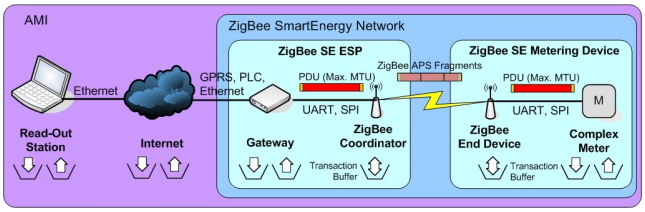

Figure 3. Smart Energy Flow Control Setup

1. **Slow Internet connection :** Data transfer from the meter to the gateway is faster than the gateway is able to send data over the Internet. The gateway input buffer gets filled.

2. **Slow meter :** Data transfer from the read-out station to the meter is faster than the meter is able to process the data. The meter's input buffer gets filled.

3. **Slow ZigBee connection :** The ZigBee Coordinator or the ZigBee End Device get their data over wire faster than they are able to pass it to the ZigBee network. The buffer in the Coordinator/End Device gets filled.

4. **Slow internal ESP/metering device connection:** The ZigBee Coordinator or the ZigBee End Device get their data over ZigBee faster than they are able to pass it to the Gateway or the Meter respectively. The buffer in the Coordinator/End Device gets filled.

5. **Asymmetric buffer space in ZigBee nodes :** If the buffers in the ZigBee Coordinator and the (maybe more restricted) ZigBee End Device are not equal in size, the smaller buffer overflows during a transfer.

A bunch of methods exist on the data link and the transport layer to control the data flow [13]. The decision on which one to choose depends on the situation at hand. Several parameters of the communication link need to be taken into account such as different data rates, buffer sizes, link qualities and round trip times. But the mandatory requirement for all of them is a form of feedback from the receiver to the sender. First, the receiver must have the means to tell the sender to stop or throttle the data flow if receive buffer space gets scarce. Or at least the sender must have a hint on what's the maximum amount of data the receiver may take and store at once. Second, as soon as the receiver has had enough time to process the buffered data, it should tell the sender that it is now ready to receive further data.

[5] It takes around 1.5s to fill a buffer of 1500 bytes if data is received at 9600 baud. This is a common transfer speed in the metering business. So the receiver must wait for more than 1.5s with a possibly incomplete PDU until it receives more data.

[6] The highest MTU of DLMS is 64kB but DLMS data units are always embedded in HDLC or TCP/IP frames. DLMS itself is able to handle fragmentation on the application layer.

The proposal is concerned with flow control within the Smart Energy network (Fig. 3). Hence, the flow needs to be controlled between the gateway and the complex meter solely. So it is assumed that flow control to remote, namely between the gateway and the read-out station, works and does not lie in our responsibility. Additionally, it is supposed that an application protocol based (end-to-end) flow control was in place before ZigBee entered the scene.

Thus, the flow control information needs to be propagated over three segments: Gateway/Coordinator, ZigBee Coordinator/End Device, End Device/Meter. Usually, the ZigBee modules are connected to the meter (or the gateway) over a serial line. In case of a synchronous interface (e.g. SPI) stopping the clock immediately stops the data flow. The asynchronous interface should support some kind of hardware (e.g. RTS/CTS wires) or software in-band flow control (e.g. ASCII XON/XOFF characters).

This leaves open the question on how flow control is handled over the air? What is specified in the ZigBee standard itself? To start with, ZigBee was initially thought for sensor network applications, in which a 127 byte long IEEE 802.15.4 frame provides enough space for the payload. Flow control is not supported natively and therefore also not explicitly mentioned in the core standard [18]. As mentioned above, on the application profile layer, the basic tunnel cluster specifies attributes which give information on transfer sizes within the ZigBee nodes. These attributes may be read over the air and give a hint on certain buffer sizes of the communication partners. But the most natural way to do flow control lies in the fragmentation algorithm. ZigBee fragmentation supports a window mechanism where the window size defines the maximum number of unacknowledged frames that can be handled at once (up to 8 as specified in [18]). This helps optimizing throughput and saves acknowledges. But as the acknowledge frames do not contain any information on flow control or buffer space left, the sender won't know about a buffer overflow in the receiver until some kind of error condition occurs (out of memory, message limit reached, delivery failed). Extending the existing acknowledge frames with additional fields is not an option as it breaks the ZigBee standard.

The only way left to provide flow control for SE without intervening with the ZigBee core standard is on the SE application level. This is why the decision was taken to extend the complex metering cluster with two new commands:

- **AckData command :** Is generated as a as a response to each *TransferData* command. It contains a field with a sequence number relating to the associated *TransferData* command (*DataNumber*) and a field providing information on how many octets may still be received by the receiver (*DataSpaceLeft*). The sender must wait for the *AckData* command before it sends any further data. An *AckData* command with *DataSpaceLeft* set to 0 completely stops the data flow.
- **ReadyData command :** Is generated after an *AckData* command with *DataSpaceLeft* set to 0. As soon as the receiver is ready to receive more data, it will send this command to the initial sender. The command contains information on how many octets may be received by the receiver *DataSpaceLeft*.

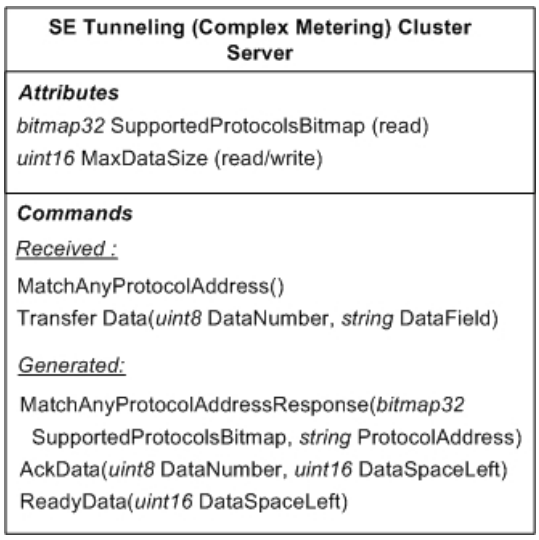

Figure 4. SE Tunneling Cluster (Meter side)

The complete complex tunneling cluster is depicted in Fig. 4 and should now include all commands and attributes to allow tunneling of various smart energy protocols over ZigBee. Concerning flow control, it still lies in the decision of the implementer, how sophisticated the flow control algorithm behaves using the information provided in the new commands (simple on/off, sliding window...).

4. Implementation and Results

In the current EnerBee implementation of the tunneling proposal, the ESP does not know about complex meter addresses. The first incoming remote data request packet is sent as a multicast to all meters that support Smart Energy tunneling. For DLMS that is completely reasonable, as the requests are quite small anyway. The ZigBee source address of the first response frame is stored and a session is opened to that node. All subsequent data frames travel between these two peers until a missing response resets the state machine and switches it back to multicast (the response timeout is 2000ms, the PDU end timeout 40ms). The state machine is depicted in Fig. 5 a).

As for the tunneling, the implementation closely follows the specifications of the two clusters *Generic Tunnel* and *Complex Metering*. This means, all commands and attributes are accessible and deliver a result although the address matching related functionality is unused at the moment. As proposed, the fragmentation provided by the ZigBee stack was put in place but had to be modified as the existing implementation was only built for a maximum of 255 bytes per transaction. If the transaction buffer gets filled, it is sent away. Even a brute force variant just filling IEEE 802.15.4 frames was successfully tested with DLMS without actually being further developed in the EnerBee project.

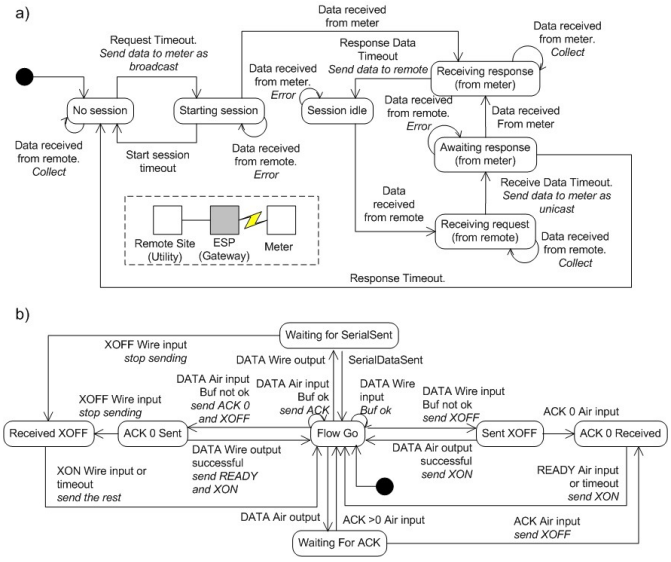

Figure 5. a) Smart Energy Tunneling and (b) Flow Control state machines

The two new cluster commands for the flow control on application level enable the firmware to support flow control over the air. The first tests have been made using in-band software flow control on the wired connections to the gateway and to the meter. The Generic Cluster attributes pointing to the tunneling transfer sizes remained unused, as the new acknowledge command (*AckData*) contains the number of bytes left to receive on the receiver side. Taking half-duplex data communication as a precondition, only one buffer was statically allocated on each ZigBee node. It acts as buffer for incoming as well as outgoing data transactions and is currently tuned to HDLC/DLMS frames of 248 bytes payload. Having only one buffer saves memory but also affects the flow control implementation in terms of reducing its complexity. If the fill level of the only buffer in a ZigBee reaches a certain water mark, data flow commands to stop further data are sent in both directions, to the wired as well as to the wireless interfaces. The flow is stopped until the complete buffer has been sent away successfully and the node is ready to receive new data. Fig. 5 b) shows the state machine which is responsible for the flow control.

The test network as well as the test setup are shown in Fig. 6. An existing PC based DLMS tool (MAP120) was used to access two meters, a Landis+Gyr ZMD310 and a ZMF100 meter over DLMS. The meters were placed in a distance of around 40m with at least one hop and several light walls in between. In this setup, it has been shown that reading out DLMS data, also involving complex commands such as reading out a complete DLMS tree or a large load profile, is feasible using a ZigBee tunnel. Furthermore, as an additional use case not based on DLMS, a simple file transfer of several kB simulating a firmware upload from remote proved the proposed flow control capabilities.

5. Releated Work

As already stated, ZigBee's CBA task group specified the clusters to tunnel BACnet over ZigBee [16] which were partly taken over for the Smart Energy proposal. The profile has not yet been officially released and no known implementations of the BACnet over ZigBee tunneling exist. One source reports a test installation using a ZigBee/BACnet gateway [10]. However, most of the considerations made for Smart Energy and DLMS also apply to Building Automation (BA) and BACnet. And although BACnet supports unconfirmed requests (no response) and some BA actions require near real-time behaviour (lighting, shading), future work is planned to adapt the Smart Energy tunneling to test it together with BACnet.

An interesting paper, which is also focused on building automation, proposes a systems approach introducing a tunnel over IEEE 802.15.4 connecting different KNX/EIB based control network segments over the air without going into the details of its implementation [11].

Few works have been carried out studying streaming of audio and multimedia content over IEEE 802.15.4 or ZigBee [2] [14] [3]. The Telecom Applications Study Group of the ZigBee Alliance intends to standardize a "voice over ZigBee" profile. But the quite different requirements of such a system concerning timing and data flow led to different design approaches.

Most of the activities to transport larger data chunks over WPAN evolve around TCP/IP solutions over IEEE 802.15.4. Several approaches have been proposed which differ in their level of abstraction, the communication layers involved and the system structure in general [15] [8] [12]. Recently, even the ZigBee Alliance has committed itself to go into that direction and support IP protocols natively in their specification. A future solution having IP combined with the IETF 6loWPAN proposal including routing schemes such as ROLL [9] would neither make tunneling obsolete nor would it replace the need for new standards on the application level. In the end, the ZigBee Smart Energy profile specification is not bound to any specific physical channel. But having TCP/IP as a vehicle to transport data would solve some of the major issues discussed in this document arising with ZigBee today.

6. Conclusion

The growing together of information and communication technologies (ICT) and energy leads to new solutions aggregated under the term Smart Energy. New standards such as the ZigBee Smart Energy profile emerge that drive the development of SE applications and have the potential to become broadly accepted in the industry and hopefully by the consumer. Still, support for existing metering protocols such as DLMS is required and should be incorporated in to the system without sacrificing the benefits of state of the art wireless technologies.

This paper studied the requirements and possible designs to tunnel such protocols over ZigBee. The result-

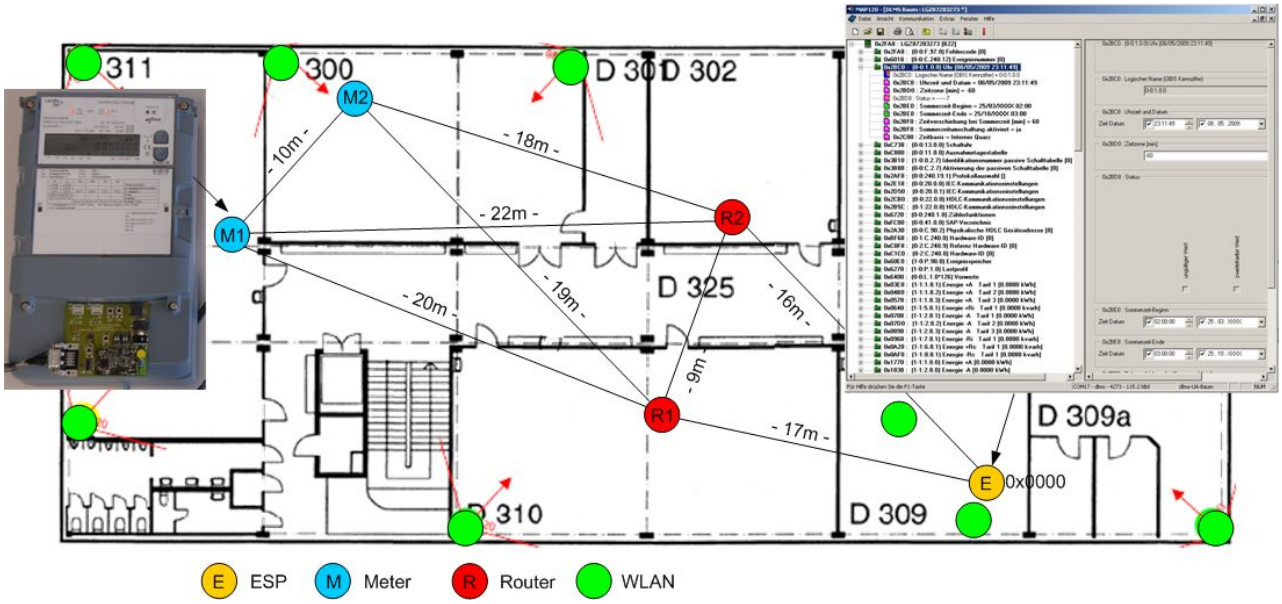

Figure 6. Smart Energy Tunneling Test Setup with Meter and MAP120 DLMS tool

ing proposal for a Smart Energy tunnel has been implemented and tested. It proved its capability in tackling common metering applications such as reading DLMS data out of a standard meter with existing commissioning tools out of the box. First steps have been made to bring the cluster specifications into the official Smart Energy profile. As for the data transport, an ongoing strong movement to support IP down to the sensor node has been identified that could affect future solutions. However, it will not render the need for new protocols on application level and continuing support for existing metering protocols unnecessary. The Smart Energy tunneling proposal satisfies both needs today.

7. Acknowledgements

The work has been supported by the CTI of the Swiss Federal Office for Professional Education and Technology (OPET) and Landis+Gyr.

References

[1] ASHRAE SSPC 135. (January 2009), BACnet-2008 - A Data Communication Protocol for Building Automatin and Control Networks [Online]. Available: http://www.bacnet.org/

[2] D. Brunelli, M. Maggiorotti, L. Benini, and F. L. Bellifemine. Analysis of Audio Streaming Capability of ZigBee Networks. In *5th European Workshop on Wireless Sensor Networks (EWSN2008)*, January 2008.

[3] S. Deshpande. Adaptive low-bitrate streaming over IEEE 802.15.4 low rate wireless personal area networks (LR-WPAN) based on link quality indication. In *International Conference on Wireless Communications and Mobile Computing*, pages 863–868, 2006.

[4] DLMS User Association. (2007), The DLMS/COSEM Specification [Online]. Available: http://www.dlms.com/en/conformance/specification.htm

[5] B. Heile. (February 2009), ZigBee + Homeplug Smart Energy Overview [Online]. Available: http://www.zigbee.org

[6] H. Hohl, J. Adame, A. Klapproth, and R. Kistler. *EnerBee - Reliable, monitored network infrastructure for wireless advanced metering devices*. CTI, 2007.

[7] IEC. (May 2005), Electricity metering - Data exchange for meter reading, tariff and load control - Part 21: Direct local data exchange [Online]. Available: http://webstore.iec.ch/webstore/webstore.nsf/artnum/031526

[8] IETF. (September 2007), IPv6 over Low power WPAN (6lowpan), RFC 4919, RFC 4944 [Online]. Available: http://www.ietf.org/html.charters/6lowpan-charter.html

[9] IETF. (April 2009), Routing Over Low power and Lossy networks (roll) [Online]. Available: http://www.ietf.org/html.charters/roll-charter.html

[10] J. P. Martocci. BACnet Unplugged - ZigBee and BACnet Connect. *ASHRAE Journal*, pages 42–46, June 2008.

[11] C. Reinisch, W. Kastner, G. Neugschwandtner, and W. Granzer. Wireless Technologies in Home and Building Automation. In *5th IEEE International Conference on Industrial Informatics (INDIN'07)*, pages 93–98, June 2007.

[12] RIPlink.org - Wavcom. (2008), RIPlink extend IP networks' reach to all kinds of resource limited devices [Online]. Available: http://www.riplink.org

[13] A. S. Tanenbaum. *Computer Networks (4th Edition)*. Prentice Hall PTR, 2002.

[14] C. Wang, K. Sohraby, R. Jana, L. Ji, and M. Daneshmand. Voice Communicatin over ZigBee Networks. *Communication Magazine*, 46.

[15] R.-C. Wang, R.-S. Chang, and H.-C. Chao. Internetworking Between ZigBee/802.15.4 and IPv6/802.3 Network. In *SIGCOMM Data Communication Festival*, August 2007.

[16] ZigBee Alliance. (November 2007), ZigBee Commercial Building Automation Profile Specification (Rev. 10) [Online]. Available: http://www.zigbee.org

[17] ZigBee Alliance. (December 2008), ZigBee Smart Energy Profile Specification (Rev. 15) [Online]. Available: http://www.zigbee.org

[18] ZigBee Alliance. (January 2008), ZigBee ZigBee Specification (Rev. 17) [Online]. Available: http://www.zigbee.org